RICE UNIVERSITY

Trust–Region Interior–Point Algorithms for a Class of Nonlinear Programming Problems

by

Luís Nunes Vicente

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

Doctor of Philosophy

Approved, Thesis Committee:

John E. Dennis, Chairman
Noah Harding Professor of Computational and Applied Mathematics

Thomas A. Badgwell
Professor of Chemical Engineering

Mahmoud El–Alem
Professor of Mathematics, Alexandria University, Egypt

Danny C. Sorensen
Professor of Computational and Applied Mathematics

Richard A. Tapia
Noah Harding Professor of Computational and Applied Mathematics

Houston, Texas
March, 1996

Abstract

Trust–Region Interior–Point Algorithms for a Class of Nonlinear Programming Problems

by

Luís Nunes Vicente

This thesis introduces and analyzes a family of trust–region interior–point (TRIP) reduced sequential quadratic programming (SQP) algorithms for the solution of minimization problems with nonlinear equality constraints and simple bounds on some of the variables. These nonlinear programming problems appear in applications in control, design, parameter identification, and inversion. In particular they often arise in the discretization of optimal control problems.

The TRIP reduced SQP algorithms treat states and controls as independent variables. They are designed to take advantage of the structure of the problem. In particular they do not rely on matrix factorizations of the linearized constraints, but use solutions of the linearized state and adjoint equations. These algorithms result from a successful combination of a reduced SQP algorithm, a trust–region globalization, and a primal–dual affine scaling interior–point method. The TRIP reduced SQP algorithms have very strong theoretical properties. It is shown in this thesis that they converge globally to points satisfying first and second order necessary optimality conditions, and in a neighborhood of a local minimizer the rate of convergence is quadratic. Our algorithms and convergence results reduce to those of Coleman and Li for box–constrained optimization. An inexact analysis is presented to provide a practical way of controlling residuals of linear systems and directional derivatives. Complementing this theory, numerical experiments for two nonlinear optimal control problems are included showing the robustness and effectiveness of these algorithms.

Another topic of this dissertation is a specialized analysis of these algorithms for equality–constrained optimization problems. The important feature of the way this family of algorithms specializes for these problems is that they do not require the computation of normal components for the step and an orthogonal basis for the null space of the Jacobian of the equality constraints. An extension of Moré and Sorensen's result for unconstrained optimization is presented, showing global convergence for these algorithms to a point satisfying the second–order necessary optimality conditions.

Acknowledgments

First, I would like to dedicate this dissertation to my parents.

I could had never written this thesis without the love, care, and support of my wife, Inês. I would like to thank her for all the joy and happiness she brought into my life. My daughter, Laura, and my son, António, have also inspired me greatly. I thank them very much and hope they will understand one day why I spent so little time playing with them. I thank my family in Portugal, and in particular my dear and supportive brother Pedro, for all the sincere encouragement and generous assistance I have received during the years in Houston.

My profound gratitude goes to my adviser, Professor John Dennis. John has always shared his ideas with me about mathematics and science, and his thoughts had a great influence upon me. He also set an example of generosity, friendship, and integrity that I will always keep in mind.

During my stay at Rice I worked closely with Professor Matthias Heinkenschloss. He introduced me to optimal control problems and much of the work reported in this thesis has resulted from this collaboration. I thank Matthias for being such a close friend and co–worker.

I would like to thank all members of my committee for their attention and support. I owe a special debt to Professors Danny Sorensen and Richard Tapia. I will never forget how much I learned from the classes I took from them. I thank Professor Mahmoud El–Alem for the stimulating discussions on trust regions and for his careful proofreading. I thank Professor Thomas Badgwell for his helpful comments and his interesting suggestions.

I also thank David Andrews and Zeferino Parada for proofreading parts of my thesis and thesis proposal.

Many thanks to all my friends and colleagues at Rice. I enjoyed the many conversations I had with David Andrews, Martin Bergman, Hector Klíe, and Richard Lehoucq.

Finally, I would like to thank Professors José Alberto Fernandes de Carvalho and Joaquim João Júdice. In 1992 and 1993 they encouraged me to study abroad and without their help and support my studies in United States would not have been possible.

Financial support from the following institutions is gratefully acknowledged: Comissão Permanente Invotan, Fundação Luso Americana para o Desenvolvimento, Comissão Cultural Luso Americana (Fulbright Program), and Departamento de Matemática da Universidade de Coimbra. Financial support for this work was also provided by the National Science Foundation cooperative agreement CCR–9120008 and by the Department of Energy contract DOE–FG03–93ER25178.

Contents

Abstract ii
Acknowledgments iv
List of Illustrations ix
List of Tables xi

1 Introduction 1
1.1 The Class of Nonlinear Programming Problems 1
1.2 Algorithms and Convergence Theory 2
1.3 Inexact Analysis and Implementation 4
1.4 Other Contributions 6
1.5 Organization of the Thesis 6
1.6 Notation 7

2 Globalization Schemes for Nonlinear Optimization 9
2.1 Basics of Unconstrained Optimization 9
2.2 Line Searches 11
2.3 The Trust–Region Technique 13
2.3.1 How to Compute a Step 15
2.3.2 The Trust–Region Algorithm 21
2.3.3 Global Convergence Results 21
2.3.4 Tikhonov Regularization 23
2.4 More about Line Searches and Trust Regions 24

3 Trust–Region SQP Algorithms for Equality–Constrained Optimization 25
3.1 Basics of Equality–Constrained Optimization 26
3.2 SQP Algorithms 28
3.3 Trust–Region Globalizations 32
3.4 A General Trust–Region Globalization of the Reduced SQP Algorithm 35
3.4.1 The Quasi–Normal Component 35
3.4.2 The Tangential Component 37

3.4.3 Outline of the Algorithm 40
3.4.4 General Assumptions 42
3.5 Intermediate Results 43
3.6 Global Convergence Results 47
3.7 The Use of the Normal Decomposition with the Least–Squares Multipliers 54
3.8 Analysis of the Trust–Region Subproblem for the Linearized Constraints 56

4 A Class of Nonlinear Programming Problems 58
4.1 Structure of the Minimization Problem 59
4.2 All–At–Once rather than Black Box 60
4.3 The Oblique Projection 63
4.4 Optimality Conditions 66
4.5 Optimal Control Examples 70
4.5.1 Boundary Control of a Nonlinear Heat Equation 71
4.5.2 Distributed Control of a Semi–Linear Elliptic Equation 72
4.6 Problem Scaling 73

5 Trust–Region Interior–Point Reduced SQP Algorithms for a Class of Nonlinear Programming Problems 74
5.1 Application of Newton's Method 76
5.2 Trust–Region Interior–Point Reduced SQP Algorithms 80
5.2.1 The Quasi–Normal Component 80
5.2.2 The Tangential Component 81
5.2.3 Reduced and Full Hessians 88
5.2.4 Outline of the Algorithms 89
5.2.5 General Assumptions 91
5.3 Intermediate Results 93
5.4 Global Convergence to a First–Order Point 98
5.5 Global Convergence to a Second–Order Point 102
5.6 Local Rate of Convergence 108
5.7 Computation of Steps and Multiplier Estimates 113
5.7.1 Computation of the Tangential Component 114
5.7.2 Computation of Multiplier Estimates 117

5.8 Numerical Example 117

6 Analysis of Inexact Trust–Region Interior–Point Reduced SQP Algorithms 122

6.1 Sources and Representation of Inexactness 124

6.2 Inexact Analysis 127

6.2.1 Global Convergence to a First–Order Point 128

6.2.2 Inexact Directional Derivatives 130

6.3 Inexact Calculation of the Quasi–Normal Component 131

6.3.1 Methods that Use the Transpose 132

6.3.2 Methods that Are Transpose Free 133

6.3.3 Scaled Approximate Solutions 135

6.4 Inexact Calculation of the Tangential Component 137

6.4.1 Reduced Gradient 137

6.4.2 Use of Conjugate Gradients to Compute the Tangential Component 138

6.4.3 Distance to the Null Space of the Linearized Constraints . . . 140

6.4.4 Fraction of Cauchy Decrease Condition 141

6.4.5 Inexact Calculation of Lagrange Multipliers 143

6.5 Numerical Experiments 144

6.5.1 Boundary Control Problem 145

6.5.2 Distributed Control Problem 146

7 Conclusions and Open Questions 152

7.1 Conclusions 152

7.2 Open Questions 153

Bibliography 155

Illustrations

1.1 Global convergence to a point that satisfies the first–order necessary optimality conditions: our result for problem (1.1) generalizes those obtained by the indicated authors for simpler problem classes. 4

1.2 Global convergence to a point that satisfies the second–order necessary optimality conditions: our result for problem (1.1) generalizes those obtained by the indicated authors for simpler problem classes. 5

2.1 A dogleg (at the left) and a conjugate–gradient (at the right) steps inside a trust region. To illustrate better the conjugate–gradient algorithm, the number of iterations is set to three, which of course exceeds the number of iterations for finite termination. 16

3.1 The quasi–normal and tangential components of the step for the coupled approach. 39

4.1 The normal and the quasi–normal components and the action of the orthogonal and oblique projectors. 65

5.1 Plots of $D(x)^2$ and $W(x)^T\nabla f(x)$ for $W(x)^T\nabla f(x) = -x+1$ and $x \in [0,4]$. 78

5.2 Plot of $D(x)^2W(x)^T\nabla f(x)$ for $W(x)^T\nabla f(x) = -x+1$ and $x \in [0,4]$. 78

5.3 The quasi–normal and tangential components of the step for the decoupled approach. We assume for simplicity that $\bar{D}_k = (\,1\,)$. 84

5.4 The quasi–normal and tangential components of the step for the coupled approach. We assume for simplicity that $\bar{D}_k = (\,1\,)$. 85

5.5 Control plot using the Coleman–Li affine scaling. 121

5.6 Control plot using the Dikin–Karmarkar affine scaling. 121

6.1 Performance of the inexact TRIP reduced SQP algorithms applied to the boundary control problem. Here $\ln_{10} f_k$ (dotted line), $\ln_{10}\|C_k\|$ (dashed line), and $\ln_{10}\|D_k W_k^T \nabla f_k\|$ (solid line) are plotted as a function of k. 147

6.2 Illustration of the performance of the inexact TRIP reduced SQP algorithms applied to the boundary control problem. These plots show the residuals $\ln_{10}\|J_k s_k^{\mathsf{q}}\|$ in dashed line and $\ln_{10}\|J_k(s_k^{\mathsf{q}} + s_k^{\mathsf{t}})\|$ in solid line. 148

6.3 Performance of the inexact TRIP reduced SQP algorithms applied to the distributed control problem. Here $\ln_{10} f_k$ (dotted line), $\ln_{10}\|C_k\|$ (dashed line), and $\ln_{10}\|D_k W_k^T \nabla f_k\|$ (solid line) are plotted as a function of k. 150

6.4 Illustration of the performance of the inexact TRIP reduced SQP algorithms applied to the distributed control problem. These plots show the residuals $\ln_{10}\|J_k s_k^{\mathsf{q}}\|$ in dashed line and $\ln_{10}\|J_k(s_k^{\mathsf{q}} + s_k^{\mathsf{t}})\|$ in solid line. 151

Tables

5.1 Number of linearized state and adjoint solvers to compute the tangential component. ($I(k)$ denotes the number of conjugate–gradient iterations.) 116
5.2 Numerical results for the boundary control problem. Case $\gamma = 10^{-2}$. . 120
5.3 Numerical results for the boundary control problem. Case $\gamma = 10^{-3}$. . 121

6.1 Number of iterations to solve the optimal control problems. 145
6.2 Number of iterations to solve large distributed semi–linear control problems. 151

Chapter 1

Introduction

Optimization, or mathematical programming, has developed enormously in the last fifty years and has reached a point where researchers often concentrate on a specific class of problems. Existing algorithmic ideas can be tailored to the characteristics of the class. These problem classes usually come from an application in industry or science. This is the case of the class of problems addressed in this thesis. Moreover, the structure of the problems in the class considered here is fundamental in taking advantage of recent advances in computer technology. The resulting algorithms are more robust and efficient, and their implementations fit more conveniently the purposes of the application.

1.1 The Class of Nonlinear Programming Problems

In this dissertation, we focus on a particular class of nonlinear programming problems that have many applications in engineering and science. The formulation of these problems is the following:

$$\begin{array}{ll} \text{minimize} & f(y,u) \\ \text{subject to} & C(y,u) = 0, \\ & a \le u \le b, \end{array} \tag{1.1}$$

where $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ and $C : \mathbb{R}^n \longrightarrow \mathbb{R}^m$ are smooth functions, $y \in \mathbb{R}^m$, $u \in \mathbb{R}^{n-m}$, and m and n are positive integers satisfying $m < n$. In this class of problems the variables x are split into two groups: state variables y, and control variables u. These are coupled through a set of nonlinear equality constraints $C(y,u) = 0$, the so–called (discretized) state equation. We also consider lower and upper bounds on the control variables u. However, bounds on the state variables y are not considered in this dissertation. The presence of such bounds would add another layer of difficulty to problem (1.1) and would require possibly a different algorithmic approach.

These optimization problems often arise in the discretization of optimal control problems that are governed by partial differential equations. We address the optimal control problems in finite dimensions after the discretization has taken place, but we do not neglect the physics and the structure that such problems have when posed naturally in infinite dimensions. These nonlinear programming problems also appear in parameter identification, inversion, and optimal design. This class of problems is rich, and we continue to find new applications on a regular basis.

The linearization of the nonlinear state equation yields the (discretized) linearized state equation and the corresponding adjoint equation. Efficient solutions of the linear system corresponding to these equations exist for many applications [22], [72], [149], and the optimization algorithm ought to take advantage of it. This linearization also offers a tremendous amount of structure. In particular, we use it to obtain a matrix whose columns form a nonorthogonal basis for the null space of the Jacobian matrix of the nonlinear equality constraints. Matrix–vector products with this matrix involve solutions of the linearized state and adjoint equations. Furthermore, a solution of the linearized state equation is naturally decomposed into two components, a *quasi–normal* component and a *tangential* component.

The algorithms that we propose and analyze in this thesis are based on an all–at–once approach (see [31]), where states y and controls u are treated as independent variables.

1.2 Algorithms and Convergence Theory

Although there are algorithms available for the solution of nonlinear programming problems that are more general than (1.1), the family of algorithms presented in this thesis is unique in the consequent use of structure inherent in many optimal control problems, the use of optimization techniques successfully applied in other contexts of nonlinear programming, and the rigorous theoretical justification.

We call our algorithms *trust–region interior–point (TRIP) reduced sequential quadratic programming (SQP) algorithms* since they combine:

1. SQP techniques to approximate the nonlinear programming problem by a sequence of quadratic programming subproblems. (We chose a reduced SQP algorithm because the reduction given by the null–space representation mentioned above appears naturally from the linearization of the nonlinear equality constraints. Both the quasi–normal and the tangential components are associated with solutions of unconstrained quadratic programming subproblems.)

2. Trust regions to guarantee global convergence, i.e. that convergence is attained from any starting point. (A trust region is imposed appropriately on the quasi–normal and tangential components constraining the respective quadratic programming subproblems. The trust–region technique we use is similar to those that Byrd and Omojokon [115], Dennis, El–Alem, and Maciel [35], and Dennis and Vicente [42] proposed for equality–constrained optimization. Besides assuring global convergence, trust regions regularize ill–conditioned second–order derivatives of the quadratic subproblems. This is very important since many problems in this class are ill–conditioned.)

3. An interior–point strategy to handle the bounds on the control variables u. (We adapt to our contex a primal–dual affine scaling algorithm proposed by Coleman and Li [24] for optimization problems with simple bounds. We accomplish this by taking advantage of the structure of our class of problems. The interior–point scheme requires no more information than is needed for the solution of these problems with no bounds on the control variables u.)

The TRIP reduced SQP algorithms have very powerful convergence properties as we show in this thesis. We prove:

1. Global convergence to a point satisfying the first–order necessary optimality conditions if first–order derivatives are used.

2. Global convergence to a point satisfying the second–order necessary optimality conditions if second–order derivatives are used.

3. Boundedness of the sequence of penalty parameters and the boundedness away from zero of the sequence of trust radii if second–order derivatives are used. The q–quadratic rate of local convergence for these algorithms is a consequence of the combination of this nice global–to–local behavior with a Newton–type iteration.

The assumptions we use to prove these results reduce to the weakest assumptions used to establish similar results in the special cases of unconstrained, equality–constrained, and box–constrained optimization. Our theoretical results, also reported in Dennis, Heinkenschloss, and Vicente [36], generalize similar ones obtained for these simpler problem classes. This is schematized in Figures 1.1 and 1.2.

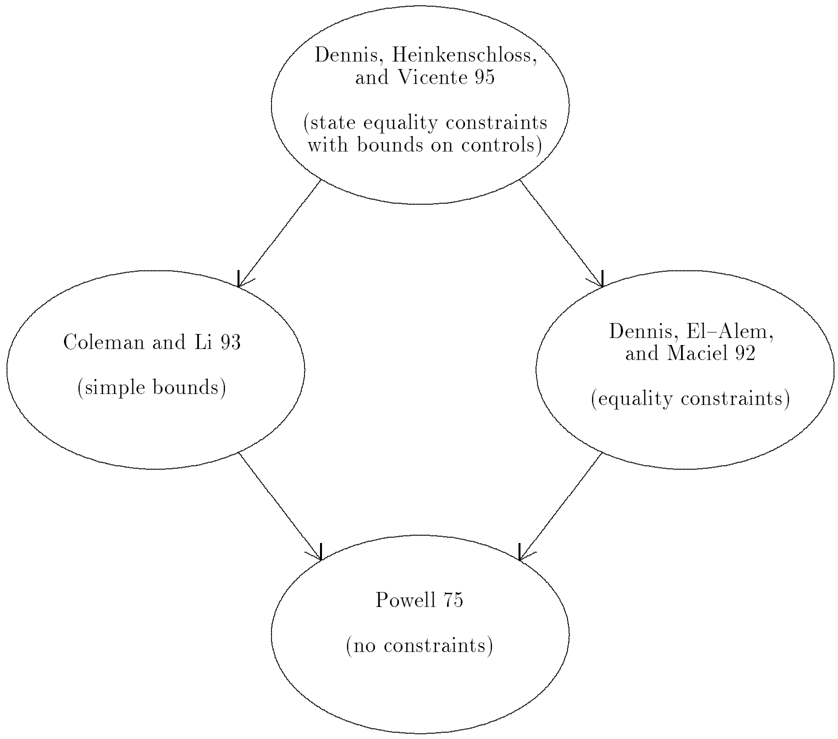

Figure 1.1 Global convergence to a point that satisfies the first–order necessary optimality conditions: our result for problem (1.1) generalizes those obtained by the indicated authors for simpler problem classes.

1.3 Inexact Analysis and Implementation

Neither the analysis of the TRIP reduced SQP algorithms nor their implementation would be complete without studying their behavior under the presence of inexactness. In practice, a very large linear system is solved inexactly yielding a certain residual. Depending on the iterative method chosen for its solution, there is the possibility of measuring and controlling the size of the residual vector. If the solution of the linear system is required at a given iteration of an optimization algorithm, the size of this residual should tighten with a measure of how feasible and optimal the current point is. An inexact analysis should provide a practical algorithmic way of accomplishing this tightening.

We present an inexact analysis for the TRIP reduced SQP algorithms that relates the size of the residual vectors of the linearized state and adjoint equations with the

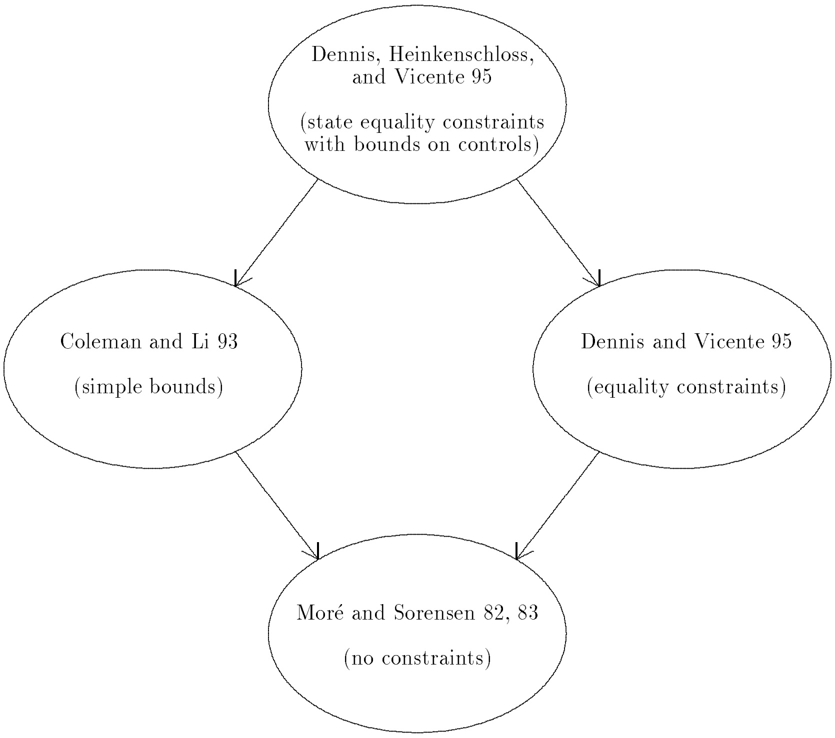

Figure 1.2 Global convergence to a point that satisfies the second–order necessary optimality conditions: our result for problem (1.1) generalizes those obtained by the indicated authors for simpler problem classes.

trust radius and the size of the constraint residual, the latter being quantities at hand at the beginning of each iteration. We provide practical rules of implementing this relationship that assure global convergence. To our knowledge, inexactness for SQP algorithms with trust–region globalizations has not been studied in the literature.

In practice the TRIP reduced SQP algorithms are robust and efficient techniques for a variety of problems. The implementation of these algorithms is currently being beta–tested with the intent of electronic distribution [76]. The current implementation provides the user with a number of alternatives to compute the steps and to approximate second–order derivatives. There are two versions, one in `Fortran 77` and one in `Matlab`. The implementation addresses the problem scaling, the computation of mass and stiffness matrices, and the setting of tolerances for inexact solvers. These issues arise frequently in optimal control problems governed by partial differential equations.

In this thesis, we present numerical results for two medium to large discretized optimal control problems: a boundary nonlinear parabolic control problem and a distributed nonlinear elliptic control problem. These numerical results are very satisfactory and indicate the effectiveness of our algorithms. Our implementation has been used successfully to solve control problems in fluid flow [22], [72].

1.4 Other Contributions

We present a brief survey of trust regions for unconstrained optimization that covers only the most important trust–region ideas used in our algorithms. In this framework, we compare line searches and trust regions from the point of view of regularization of ill–conditioned second–order approximations.

The ability to converge globally to points satisfying the second–order necessary optimality conditions is natural for trust–regions, and it has been shown in the literature for different classes of problems and different trust–region algorithms. We prove this property also for a family of general trust–region algorithms [35], [42] for equality–constrained optimization that use nonorthogonal null–space basis and quasi–normal components. This analysis, of value by itself, motivates all the convergence theory for the TRIP reduced SQP algorithms.

1.5 Organization of the Thesis

Chapters 2 and 3 review basic material on unconstrained and equality–constrained optimization that is used in the other chapters. The reader familiar with these basic concepts might want to skip many of the sections in these two chapters. In Chapter 2, we discuss and compare the regularization of ill–conditioned second–order approximations for line searches and trust regions. In Chapter 3, we derive global convergence to a point satisfying second–order necessary optimality conditions for a family of trust–region reduced SQP algorithms for equality–constrained optimization, and present an analysis of the trust–region subproblem for the linearized constraints.

The class of problems (1.1) is described in great detail in Chapter 4, where we establish optimality conditions and comment on the use of structure.

Chapters 5 and 6 are the two main chapters of this thesis. They describe the TRIP reduced SQP algorithms for our class of problems and prove their convergence properties. Chapter 5 focuses on the exact version of these algorithms and includes both global and local convergence results. In Chapter 6, we study the global behavior of the TRIP reduced SQP algorithms under the presence of inexactness. Sections 5.8 and 6.5 contain numerical experiments.

The most important conclusions and open questions are summarized in Chapter 7.

A short introduction and a summary of contents are given at the beginning of every chapter. There we cite related work and justify our algorithmic choices.

1.6 Notation

We list below some of the notation and abbreviations used in this thesis.

- $\ell(x, \lambda) = f(x) + \lambda^T C(x)$ is the Lagrangian function associated with the problem *minimize* $f(x)$ *subject to* $C(x) = 0$, where λ is the Lagrange multiplier vector.

- $\nabla f(x)$ is the gradient of the real–valued function $f(x)$ and $J(x)$ is the Jacobian of the vector–valued function $C(x) = \big(c_1(x), \ldots, c_m(x)\big)^T$.

- $\nabla^2 f(x)$, $\nabla^2 c_i(x)$, and $\nabla^2_{xx}\ell(x, \lambda) = \nabla^2 f(x) + \sum_{i=1}^m \lambda_i \nabla^2 c_i(x)$ are the Hessians matrices with respect to x of $f(x)$, $c_i(x)$, and $\ell(x, \lambda)$ respectively.

- $\mathcal{N}(A)$ represents the null space of the matrix A.

- $W(x)$ (resp. $Z(x)$) is a matrix whose columns form a basis (resp. an orthogonal basis) for the null space of $J(x)$.

- Subscripted indices are used to represent the evaluation of a function at a particular point of the sequences $\{x_k\}$ and $\{\lambda_k\}$. For instance, f_k represents $f(x_k)$ and ℓ_k is the same as $\ell(x_k, \lambda_k)$.

- The vector and matrix norms $\|\cdot\|$ are the ℓ_2 norms.

- The sequence $\{x_k\}$ is bounded if there exists $\alpha > 0$ independent of k such that $\|x_k\| \leq \alpha$ for all k. In this case we say that the element x_k of the sequence $\{x_k\}$ is uniformly bounded.

- I_p represents the identity matrix of order p with columns $e_1, \ldots, e_p$.

- $\lambda_1(A)$ denotes the smallest eigenvalue of the symmetric matrix A.
- $\kappa(A)$ represents the ℓ_2 condition number of the matrix A with respect to inversion. For nonsingular square matrices $\kappa(A) = \|A\| \, \|A^{-1}\|$. In general, we have $\kappa(A) = \frac{\sigma_1(A)}{\sigma_r(A)}$, where r is the rank of A, and $\sigma_1(A)$ and $\sigma_r(A)$ are the largest and smallest singular values of A, respectively.
- The element x_k of the sequence $\{x_k\}$ is $\mathcal{O}(y_k)$ if there exists a positive constant $\kappa > 0$ independent of k such that $\|x_k\| \leq \kappa \|y_k\|$ for all k.
- SQP algorithms: sequential quadratic programming algorithms.
- TRIP reduced SQP algorithms: trust–region interior–point reduced SQP algorithms.

Chapter 2

Globalization Schemes for Nonlinear Optimization

Consider the unconstrained optimization problem

$$\text{minimize} \qquad f(x)\,, \tag{2.1}$$

where $x \in \mathbb{R}^n$ and $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ is at least twice continuously differentiable. One purpose of this chapter is to use this problem to provide necessary background for this thesis of fundamental concepts of nonlinear optimization like line–search and trust–region globalization schemes. We support the claim that the trust–region technique has built–in a regularization of ill–conditioned second–order approximations. The organization of this chapter is the following. The optimality conditions and other basic concepts of unconstrained optimization are reviewed in Section 2.1. In Section 2.2, we give a very brief introduction to line searches. Trust regions are presented with more detail in Section 2.3. In Section 2.4, we compare these two globalization strategies focusing on their regularization properties.

2.1 Basics of Unconstrained Optimization

The optimality conditions for the unconstrained optimization problem (2.1) are given in the following proposition.

Proposition 2.1.1 Let f be continuously differentiable. If the point x_* is a local minimizer for problem (2.1) then

$$\nabla f(x_*) = 0.$$

In this case x_* is called a stationary point or a point that satisfies the first–order necessary optimality conditions.

Now let us assume that f is twice continuously differentiable. The second–order necessary (resp. sufficient) optimality conditions for x_* to be a local minimizer for (2.1) are

$$\nabla f(x_*) = 0 \quad \text{and}$$

$$\nabla^2 f(x_*) \quad \text{is positive semi–definite (resp. definite).}$$

The proofs of these basic results can be found in many textbooks like [39], [116].

A quasi–Newton method for the solution of (2.1) generates a sequence of iterates $\{x_k\}$ and steps $\{s_k\}$ such that $x_{k+1} = x_k + s_k$. At x_k, a quadratic model of $f(x_k + s)$,

$$q_k(s) = f(x_k) + g_k^T s + \frac{1}{2} s^T H_k s,$$

is formed, where $g_k = \nabla f(x_k)$ and H_k is a symmetric matrix of order n that approximates the Hessian $\nabla^2 f(x_k)$ and introduces curvature into the model. The quasi–Newton step s_k is computed using the quadratic model $q_k(s)$.

Algorithm 2.1.1 *(Basic Quasi–Newton Algorithm)*

1. Choose x_0.
2. For $k = 0, 1, 2, \ldots$ do

   2.1 Stop if x_k satisfies the stopping criterion.

   2.2 Compute s_k as an approximate solution of

   $$\text{minimize} \qquad f(x_k) + g_k^T s + \frac{1}{2} s^T H_k s$$

   2.3 Set $x_{k+1} = x_k + s_k$ and compute H_{k+1} possibly by updating H_k.

A possible stopping criterion is $\|g_k\| \le \epsilon_{tol}$ for some $\epsilon_{tol} > 0$.

If H_k is nonsingular, a typical quasi–Newton step s_k is given by $s_k = -H_k^{-1} g_k$. If in addition H_k is positive definite, then this quasi–Newton step $s_k = -H_k^{-1} g_k$ is the unconstrained minimizer of $q_k(s)$. In Newton's method, we have $H_k = \nabla^2 f(x_k)$. Newton's method is credited to Newton (see [143]) in the 1660's for finding a root of a nonlinear equation with one variable using a technique similar to Newton's method, but where the calculations are organized differently. Raphson [124] plays an important role in this discovery by rederiving Newton's technique in a way that is very close to what is used nowadays. The multidimensional version of Newton's method is due to Simpson [131] in 1740. See the survey paper by Ypma [150].

It is well–known that the basic quasi–Newton algorithm is not globally convergent to a stationary point [39][Figure 6.3.2]. If we want to start with any choice of x_0 and still guarantee convergence, then we need a globalization strategy. The most often used globalization strategies for quasi–Newton algorithms are line searches and trust regions.

A line–search strategy requires a direction d_k from which a step is obtained. The step s_k is of the form $\mu_k d_k$, where the step length μ_k is chosen in an appropriate way and d_k is a descent direction, i.e. $d_k^T g_k < 0$. If H_k is nonsingular, $d_k = -H_k^{-1} g_k$ might be a reasonable choice.

The trust–region technique does not necessarily choose a specific pattern of directions. Here a step s_k is a sufficiently good approximate solution of the trust–region subproblem

$$\begin{aligned} \text{minimize} \quad & q_k(s) \\ \text{subject to} \quad & \|s\| \le \delta_k, \end{aligned} \tag{2.2}$$

where δ_k is the trust radius. We will be more precise later. More general forms of this simple trust–region subproblem are considered in the papers [74], [100], [103], [105], [136], [140], [153].

2.2 Line Searches

If a line search is used, one might ask the step $s_k = \mu_k d_k$ to satisfy the Armijo–Goldstein–Wolfe conditions:

$$f(x_k + s_k) \le f(x_k) + \eta_1 g_k^T s_k, \tag{2.3}$$

$$\nabla f(x_k + s_k)^T s_k \ge \eta_2 g_k^T s_k, \tag{2.4}$$

where η_1 and η_2 are constants fixed for all k and satisfying $0 < \eta_1 < \eta_2 < 1$. Let θ_k denote the angle between d_k and $-g_k$ defined through

$$\cos(\theta_k) = -\frac{d_k^T g_k}{\|d_k\| \|g_k\|}, \quad \theta_k \in \left[0, \frac{\pi}{2}\right].$$

We present now the basic line–search algorithm and its classical convergence result.

Algorithm 2.2.1 *(Basic Line–Search Algorithm)*

1. Choose x_0, η_1, and η_2 such that $0 < \eta_1 < \eta_2 < 1$.

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

   2.1 Stop if x_k satisfies the stopping criterion.

   2.2 Compute a direction d_k based on $q_k(s)$.

   2.3 Compute $s_k = \mu_k d_k$ to satisfy (2.3) and (2.4), and set $x_{k+1} = x_k + s_k$.

A possible stopping criterion is $\|g_k\| \leq \epsilon_{tol}$ for some $\epsilon_{tol} > 0$.

Theorem 2.2.1 Let f be bounded below and ∇f be uniformly continuous. If for all k, $s_k = \mu_k d_k$ satisfies (2.3)–(2.4) and the direction d_k is descent, then

$$\lim_{k \to +\infty} \cos(\theta_k)\|g_k\| = 0.$$

Some of the ground work that led to this result was provided by Armijo [2] and Goldstein [65]. It was established by Wolfe [144], [145] and Zoutendijk [158], under the assumption that the gradient is Lipschitz continuous. However this condition can be relaxed and one can see that uniform continuity is enough (see Fletcher [53][Theorem 2.5.1]). Some practical line–search algorithms are described by Moré and Thuente [107]. For more references see also the books [39], [112], [116] and the review papers [40], [113].

From Theorem 2.2.1, a key ingredient to obtain global convergence to a stationary point is to keep the angle θ_k between $-g_k$ and d_k uniformly bounded away from $\frac{\pi}{2}$.

Now let us consider the case where H_k is nonsingular and $d_k = -H_k^{-1} g_k$. If the condition number $\kappa(H_k)$ of the matrix H_k is uniformly bounded, i.e. if there exists a $\nu > 0$ such that

$$\kappa(H_k) \leq \nu$$

for every k, then we have

$$\cos(\theta_k) = \frac{g_k^T H_k^{-1} g_k}{\|g_k\| \|H_k^{-1} g_k\|} \geq \frac{1}{\nu}. \tag{2.5}$$

One way of assuring that the direction $-H_k^{-1} g_k$ is descent is to force H_k to be positive definite. The following corollary of Theorem 2.2.1 is a result of these considerations.

Corollary 2.2.1 Let f be bounded below and ∇f be uniformly continuous. If for all k, H_k is positive definite, $s_k = -\mu_k H_k^{-1} g_k$ satisfies

(2.3)–(2.4), and the condition number $\kappa(H_k)$ of H_k is uniformly bounded, then $\{x_k\}$ satisfies

$$\lim_{k\to+\infty} \|g_k\| = 0.$$

2.3 The Trust–Region Technique

The development of trust regions started with the work of Levenberg [93] (1944), Marquardt [97] (1963), and Goldfeld, Quandt, and Trotter [64] (1966). A few years later Powell [120], [121] (1970, 1975), Hebden [71] (1973), and Moré [102] (1978) opened the field of research in this area. Trust–region algorithms are efficient and robust techniques to solve unconstrained optimization problems. An excellent survey in this area was written by Moré [103] in 1983.

Let us describe how the trust–region technique works. A step s_k has to decrease the quadratic model $q_k(s)$ from $s = 0$ to $s = s_k$. The way s_k is computed determines the magnitude of the predicted decrease $q_k(0) - q_k(s_k)$ and influences the type of global convergence of the trust–region algorithm. One can ask s_k to satisfy two classical conditions, either fraction of Cauchy decrease (simple decrease) or fraction of optimal decrease.

The first condition forces the predicted decrease to be at least as large as a fraction of the decrease given for $q_k(s)$ by the Cauchy step c_k. This step is defined as the solution of the one–dimensional problem

$$\begin{array}{rl} \text{minimize} & q_k(c) \\ \text{subject to} & \|c\| \le \delta_k,\ c \in span\{-g_k\}, \end{array}$$

and it is given by

$$c_k = \begin{cases} -\frac{\|g_k\|^2}{g_k^T H_k g_k} g_k & \text{if } \frac{\|g_k\|^3}{g_k^T H_k g_k} \le \delta_k, \\ -\frac{\delta_k}{\|g_k\|} g_k & \text{otherwise.} \end{cases} \tag{2.6}$$

The primitive form of a steepest–descent algorithm was discovered by Cauchy [20] in 1847. The step c_k is called the Cauchy step because the direction $-g_k$ is the steepest–descent direction for $q_k(s)$ at $s = 0$ in the ℓ_2 norm, i.e. $-\frac{g_k}{\|g_k\|}$ is the solution of

$$\begin{array}{rl} \text{minimize} & g_k^T d \\ \text{subject to} & \|d\| = 1. \end{array}$$

The step s_k is said to satisfy a fraction of Cauchy decrease for the trust–region subproblem (2.2) if

$$\begin{aligned} q_k(0) - q_k(s_k) &\geq \beta_1\Big(q_k(0) - q_k(c_k)\Big), \\ \|s_k\| &\leq \delta_k, \end{aligned} \tag{2.7}$$

where β_1 is positive and fixed across all iterations. The following lemma expresses this decrease condition in a way that is very convenient to prove global convergence to a stationary point.

Lemma 2.3.1 *(Powell [121])* If s_k satisfies the fraction of Cauchy decrease (2.7), then

$$q_k(0) - q_k(s_k) \geq \frac{\beta_1}{2}\|g_k\| \min\left\{\frac{\|g_k\|}{\|H_k\|},\, \delta_k\right\}.$$

Proof Define $\psi : \mathbb{R}^+ \longrightarrow \mathbb{R}$ as $\psi(t) = q_k\left(-t\frac{g_k}{\|g_k\|}\right) - q_k(0)$. Then $\psi(t) = -\|g_k\|t + \frac{r_k}{2}t^2$, where $r_k = \frac{g_k^T H_k g_k}{\|g_k\|^2}$. Let t_k^* be the minimizer of ψ in $[0, \delta_k]$. If $t_k^* \in (0, \delta_k)$ then

$$\psi(t_k^*) = \psi\left(\frac{\|g_k\|}{r_k}\right) = -\frac{1}{2}\frac{\|g_k\|^2}{r_k} \leq -\frac{1}{2}\frac{\|g_k\|^2}{\|H_k\|}. \tag{2.8}$$

If $t_k^* = \delta_k$ then either $r_k > 0$ in which case $\frac{\|g_k\|}{r_k} \geq \delta_k$ or $r_k \leq 0$ in which case $r_k\delta_k \leq \|g_k\|$. In either event,

$$\psi(t_k^*) = \psi(\delta_k) = -\delta_k\|g_k\| + \frac{r_k}{2}\delta_k^2 \leq -\frac{\delta_k}{2}\|g_k\|. \tag{2.9}$$

We can combine (2.8) and (2.9) with

$$q_k(0) - q_k(s_k) \geq \beta_1\Big(q_k(0) - q_k(c_k)\Big) = -\beta_1\psi(t_k^*)$$

to get the desired result. □

The second condition is more stringent and relates the predicted decrease to the decrease given on $q_k(s)$ by the optimal solution o_k of the trust–region subproblem (2.2). The step s_k is said to satisfy a fraction of optimal decrease for the trust–region subproblem (2.2) if

$$\begin{aligned} q_k(0) - q_k(s_k) &\geq \beta_2\Big(q_k(0) - q_k(o_k)\Big), \\ \|s_k\| &\leq \beta_3\delta_k, \end{aligned} \tag{2.10}$$

where β_2 and β_3 are positive and fixed across all iterations. The condition $\|s_k\| \leq \beta_3\delta_k$ replaces the condition $\|s_k\| \leq \delta_k$ in (2.7). There is no need to have a parameter like β_3 in (2.7) since the algorithms that compute steps satisfying only a fraction of Cauchy decrease do not cross the boundary of the trust region. An important point here is that if one sets out in practice to exactly solve (2.2), one will satisfy (2.10).

2.3.1 How to Compute a Step

Several algorithms were proposed to compute a step s_k that satisfies the fraction of Cauchy decrease (2.7). The first is due to Powell [120], and it is called the dogleg algorithm. The idea behind this algorithm is very simple and is described below.

Algorithm 2.3.1 *(Dogleg Algorithm (H_k Positive Definite))*

Compute the Cauchy step c_k. If $\|c_k\| = \delta_k$ then set $s_k = c_k$. Otherwise compute the quasi–Newton step $-H_k^{-1}g_k$, and if it is inside the trust region, set $s_k = -H_k^{-1}g_k$. If not, consider the convex combination $s(\alpha) = (1-\alpha)c_k - \alpha H_k^{-1}g_k$, $\alpha \in [0,1]$, and pick α_* such that $\|s(\alpha_*)\| = \delta_k$. Set $s_k = s(\alpha_*)$.

A dogleg step is depicted in Figure 2.1 for a value of α_* strictly between one and zero.

The dogleg algorithm is well defined for H_k positive definite (see for instance [39]) and can be extended to the case where H_k is indefinite. A possible way to accomplish this is to generalize the use of the classical conjugate–gradient algorithm of Hestenes and Stiefel [78] for the solution of the linear system $H_k s = -g_k$ with H_k positive definite. Steihaug [134] and Toint [139] adapted this algorithm for the solution of the trust–region subproblem (2.2). Here two new situations have to be considered. First H_k might not be positive definite. This can be fixed by stopping the conjugate–gradient loop when the first direction of nonpositive curvature is found and using this direction to move to the boundary of the trust–region. The other situation happens when an iterate of the conjugate–gradient algorithm passes the boundary of the trust region. Here the dogleg idea can be used to stop at the boundary of the trust region. This latter situation is illustrated in Figure 2.1. The conjugate–gradient algorithm is given below.

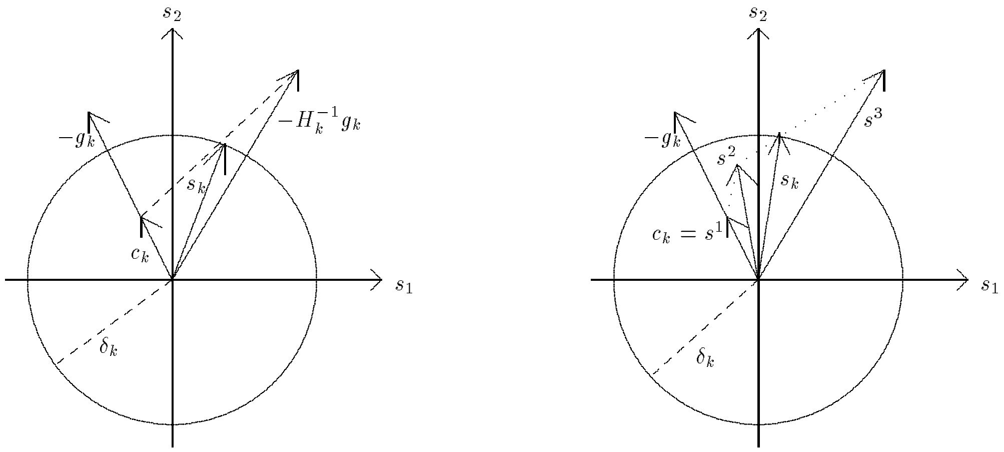

Figure 2.1 A dogleg (at the left) and a conjugate–gradient (at the right) steps inside a trust region. To illustrate better the conjugate–gradient algorithm, the number of iterations is set to three, which of course exceeds the number of iterations for finite termination.

Algorithm 2.3.2 *(Conjugate–Gradient Algorithm for Trust Regions)*

1. Set $s^0 = 0$, $r^0 = -g_k$, and $d^0 = r^0$; pick $\epsilon > 0$.
2. For $i = 0, 1, 2, \ldots$ do

   2.1 Compute $\gamma^i = \frac{(r^i)^T(r^i)}{(d^i)^T H_k (d^i)}$.

   2.2 Compute τ^i such that $\|s^i + \tau d^i\| = \delta_k$.

   2.3 If $\gamma^i \leq 0$, or if $\gamma^i > \tau^i$, then set $s_k = s^i + \tau^i d^i$ and stop; otherwise set $s^{i+1} = s^i + \gamma^i d^i$.

   2.4 Update the residual: $r^{i+1} = r^i - \gamma^i H_k d^i$.

   2.5 Check truncation criterion: if $\frac{\|r^{i+1}\|}{\|r^0\|} \leq \epsilon$, set $s_k = s^{i+1}$ and stop.

   2.6 Compute $\alpha^i = \frac{(r^{i+1})^T(r^{i+1})}{(r^i)^T(r^i)}$ and the new direction $d^{i+1} = r^{i+1} + \alpha^i d^i$.

The following proposition characterizes the type of step computed by these two algorithms.

Proposition 2.3.1 The Dogleg Algorithm 2.3.1 and the Conjugate–Gradient Algorithm 2.3.2 compute steps s_k that satisfy the Cauchy decrease condition (2.7) with $\beta_1 = 1$.

For both algorithms the proof relies on the fact that they start by minimizing the quadratic model $q_k(s)$ along the steepest–descent direction $-g_k$. The proof for the dogleg algorithm depends strongly on the positive definiteness of H_k and can be found in [39]. The proof for conjugate gradients is given in [134] and uses the fact that s^{i+1} is the optimal solution of the quadratic $q_k(s)$ in the Krylov subspace

$$\mathcal{K}_i(H_k, -g_k) = span \left\{-g_k, -H_k g_k, \ldots, -(H_k)^{i-1} g_k\right\}.$$

Other generalizations of the dogleg idea were suggested in the literature. Dennis and Mei [37] proposed the so–called double dogleg algorithm. Byrd, Schnabel, and Shultz [18], [130] introduced indefinite dogleg algorithms using two dimensional subspaces.

Now we turn our attention to algorithms for computing steps s_k that satisfy the fraction of optimal decrease (2.10). Typically these algorithms are based on Newton type iterations and rely on the following propositions.

Proposition 2.3.2 The trust–region subproblem (2.2) has no solutions at the boundary $\{s : \|s\| = \delta_k\}$ if and only if H_k is positive definite and $\|H_k^{-1} g_k\| \le \delta_k$.

A proof of this simple fact can be found in [106].

Proposition 2.3.3 *(Gay [56] and Sorensen [132])* The step o_k is an optimal solution of the trust–region subproblem (2.2) if and only if $\|o_k\| \le \delta_k$ and there exists $\gamma_k \ge 0$ such that

$$H_k + \gamma_k I_n \quad \text{is positive semi–definite,} \tag{2.11}$$

$$(H_k + \gamma_k I_n)\, o_k = -g_k, \text{ and} \tag{2.12}$$

$$\gamma_k\, (\delta_k - \|o_k\|) = 0. \tag{2.13}$$

The optimal solution o_k is unique if $H_k + \gamma_k I_n$ is positive definite.

The necessary part of these conditions can be seen as an application of a powerful tool of Lagrange multiplier theory, the so–called Karush–Kuhn–Tucker optimality conditions, to the trust–region subproblem (2.2). These conditions are stated in

Propositions 4.4.1 and 4.4.2. The parameter γ_k is the Lagrange multiplier associated with the trust–region constraint $\|s\|^2 \le \delta_k^2$. The gradient with respect to s of the Lagrangian function $\ell(s,\gamma) = q_k(s) - \gamma(\delta_k^2 - \|s\|^2)$ is zero if and only if (2.12) holds. Condition (2.13) is the complementarity condition. Conditions (2.12), (2.13), $\gamma_k \ge 0$, and $\|o_k\| \le \delta_k$ are the first–order necessary optimality conditions. If we add (2.11) we get the second–order necessary optimality conditions. Of course Lemma 2.3.3 says that these conditions are also sufficient but this part does not follow from the Karush–Kuhn–Tucker theory.

As a consequence of Proposition 2.3.3 we can write

$$q_k(0) - q_k(o_k) = \frac{1}{2}\left(\|R_k o_k\|^2 + \gamma_k \delta_k^2\right),$$

where $H_k + \gamma_k I_n = R_k^T R_k$. From this we have the following lemma.

Lemma 2.3.2 If s_k satisfies the fraction of optimal decrease (2.10), then

$$q_k(0) - q_k(s_k) \ge \frac{\beta_2}{2}\gamma_k \delta_k^2.$$

One can compare Lemmas 2.3.1 and 2.3.2 and see how the two decrease conditions (2.7) and (2.10) influence the accuracy of the predicted decrease $q_k(0) - q_k(s_k)$. Both lemmas are critical for proving global convergence results.

It follows from Propositions 2.3.2 and 2.3.3 that finding the optimal solution of the trust–region subproblem (2.2) is equivalent in all cases but one to finding γ such that $\gamma \ge 0$, $H_k + \gamma I_n$ is positive semi–definite and

$$\phi_1(\gamma) \equiv \delta_k - \|s(\gamma)\| = 0, \tag{2.14}$$

where $s(\gamma)$ satisfies

$$(H_k + \gamma I_n)\, s(\gamma) = -g_k.$$

The root finding problem (2.14) is usually solved by applying Newton's method to the equation:

$$\phi_2(\gamma) \equiv \frac{1}{\delta_k} - \frac{1}{\|s(\gamma)\|} = 0. \tag{2.15}$$

It can be shown that both functions ϕ_1 and ϕ_2 are convex and strictly decreasing in $(-\lambda_1(H_k), +\infty)$, where $\lambda_1(H_k)$ denotes the smallest eigenvalue of H_k. Reinsch [125] and Hebden [71] were the first to observe that Newton's method performs better when applied to (2.15). The reason is that ϕ_1 has a pole at $-\lambda_1(H_k)$ whereas ϕ_2 is nearly linear in $(-\lambda_1(H_k), +\infty)$.

A Newton's iteration for these root finding equations faces numerical problems if γ_k is very close to $-\lambda_1(H_k)$ or if the so called *hard case* occurs. The hard case is characterized by the following two conditions:

(a) g_k is orthogonal to the eigenspace of $-\lambda_1(H_k)$ and

(b) $\left\|(H_k + \gamma I_n)^{-1} g_k\right\| < \delta_k$, for all $\gamma > 0$.

If the hard case occurs, the rightmost root γ of (2.15) is such that $H_k + \gamma I_n$ is indefinite. Hence Newton's iteration has to be modified if one wants to compute a γ_k such that conditions (2.11)–(2.13) hold. In the hard case, a solution o_k for the trust–region subproblem (2.2) is given by

$$o_k = p + \tau q \tag{2.16}$$

where p solves $(H_k - \lambda_1(H_k) I_n) p = -g_k$, the vector q is a eigenvector corresponding to $\lambda_1(H_k)$, and τ is such that

$$\|p + \tau q\| = \delta_k.$$

Moré and Sorensen [106] proposed an algorithm that combines the application of Newton's method to (2.15) for the easy case with (2.16) for the hard case. They showed that the algorithm computes a step s_k satisfying the optimal decrease conditions (2.10). Their algorithm and corresponding Fortran implementation `GQTPAR` are based on previous work done by Gay [56] and Sorensen [132].

To compute $\phi_2(\gamma)$ and $\phi_2'(\gamma)$, algorithms of the Moré and Sorensen type require a Cholesky factorization $R_\gamma^T R_\gamma$ of $H_k + \gamma I_n$ whenever this matrix is positive definite. In fact if we solve $R_\gamma^T R_\gamma s_\gamma = -g_k$ and $R_\gamma^T q_\gamma = s_\gamma$ we have

$$\phi_2(\gamma) = \frac{1}{\delta_k} - \frac{1}{\|s_\gamma\|} \quad \text{and} \quad \phi_2'(\gamma) = -\frac{\|q_\gamma\|^2}{\|s_\gamma\|^3}.$$

In large problems the computation of the Cholesky factorization might not be practical.

Recent new algorithms to compute a step that satisfies a fraction of optimal decrease that are very promising for large problems have been proposed by Rendl and Wolkowicz [126], Sorensen [133], and Santos and Sorensen [129]. They rely on different parametrizations of the trust–region subproblem (2.2). Instead of a Cholesky factorization, these algorithms require only matrix–vector products. The material in the following paragraph follows the exposition in [129], [133].

The motivation for the new parametrization is that

$$\frac{1}{2}\alpha + q_k(s) = \frac{1}{2}\begin{pmatrix} 1 \\ s \end{pmatrix}^T \begin{pmatrix} \alpha & g_k^T \\ g_k & H_k \end{pmatrix} \begin{pmatrix} 1 \\ s \end{pmatrix}. \tag{2.17}$$

The new one–dimensional function depends on the parameter α and is defined as

$$\phi_3(\gamma;\alpha) \equiv \alpha - \gamma(\alpha).$$

Let $\gamma(\alpha)$ be the smallest eigenvalue of the bordered matrix given in (2.17). The hard case occurs when the eigenvectors of the bordered matrix associated with $\gamma(\alpha)$ have zero in its first component. If this is not the case, i.e. if there exists an s such that

$$\begin{pmatrix} \alpha & g_k^T \\ g_k & H_k \end{pmatrix} \begin{pmatrix} 1 \\ s \end{pmatrix} = \begin{pmatrix} 1 \\ s \end{pmatrix} \gamma(\alpha),$$

then we have

$$\begin{aligned} & H_k + \gamma(\alpha) I_n \quad \text{is positive semi–definite,} \\ & (H_k + \gamma(\alpha) I_n) s = -g_k, \\ & \phi_3(\gamma;\alpha) = -g_k^T s, \quad \text{and} \\ & \frac{d}{d\gamma}\phi_3(\gamma;\alpha) = \|s\|^2. \end{aligned} \tag{2.18}$$

From (2.18) we can see that solving the trust–region subproblem (2.2) is equivalent to finding α such that

$$\frac{d}{d\gamma}\phi_3(\gamma;\alpha) = \|s\|^2 = \delta_k.$$

If such a $\gamma(\alpha)$ is nonnegative, then the corresponding s is the optimal solution of the trust–region subproblem (2.2). The parameter α can be found by using interpolating schemes. If the trust–region subproblem (2.2) has an unconstrained minimizer, then during the process of choosing α a negative $\gamma(\alpha)$ is found such that $\|s\| < \delta_k$. In this case H_k is positive definite, $-H_k^{-1} g_k$ is inside the trust region, and the conjugate–gradient algorithm can be used to solve $H_k s = -g_k$.

2.3.2 The Trust–Region Algorithm

The predicted decrease $pred(s_k)$ given by s_k is defined as $q_k(0) - q_k(s_k)$. The actual decrease $ared(s_k)$ is given by $f(x_k) - f(x_k + s_k)$. The trust–region strategy relates the acceptance of s_k with the ratio

$$ratio(s_k) = \frac{ared(s_k)}{pred(s_k)}.$$

We have the following basic trust–region algorithm.

Algorithm 2.3.3 *(Basic Trust–Region Algorithm)*

1. Choose x_0, α, and η such that $0 < \alpha, \eta < 1$.
2. For $k = 0, 1, 2, \ldots$ do
   - 2.1 Stop if x_k satisfies the stopping criterion.
   - 2.2 Compute a step s_k based on the subproblem (2.2).
   - 2.3 If $ratio(s_k) < \eta$ reject s_k, set $\delta_{k+1} = \alpha \|s_k\|$ and $x_{k+1} = x_k$. If $ratio(s_k) \geq \eta$ accept s_k, choose $\delta_{k+1} \geq \delta_k$ and set $x_{k+1} = x_k + s_k$.

Of course the rules to update the trust radius can be much more involved to enhance efficiency but the above suffices to prove convergence results and to understand the trust–region mechanism.

Two reasonable stopping criteria are $\|g_k\| \leq \epsilon_{tol}$ and $\|g_k\| + \gamma_k \leq \epsilon_{tol}$ for a given $\epsilon_{tol} > 0$, where γ_k is the Lagrange multiplier associated with the trust–region constraint $\|s_k\| \leq \delta_k$ as described in Proposition 2.3.3. The former criterion forces global convergence to a stationary point (see Theorem 2.3.1), and the latter forces global convergence to a point satisfying the second–order necessary optimality conditions (see Theorem 2.3.3).

2.3.3 Global Convergence Results

Global convergence of trust–region algorithms to stationary points for unconstrained optimization is summarized in Theorems 2.3.1 and 2.3.2.

Theorem 2.3.1 *(Powell [121])* Let $\{x_k\}$ be a sequence generated by the Trust–Region Algorithm 2.3.3, where s_k satisfies the fraction of Cauchy decrease (2.7). Let f be continuously differentiable and bounded below in $\mathcal{L}(x_0) = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}$. If $\{H_k\}$ is bounded, then

$$\liminf_{k \to +\infty} \|g_k\| = 0. \tag{2.19}$$

Theorem 2.3.2 *(Thomas [137])* If in addition to the assumptions of Theorem 2.1, f is uniformly continuous in $\mathcal{L}(x_0)$ then

$$\lim_{k \to +\infty} \|g_k\| = 0.$$

The proofs of these theorems can be found in [103]. We remark that Powell in [121] proved (2.19) for a slightly different update of the trust radius.

The assumption on the Hessian approximation H_k can be weakened. Powell [122] proved a convergence result in the case where there is a bound on the second–order approximation H_k that depends linearly on the iteration counter k. Carter [19] established analogous results for the case where the gradients $g_k = \nabla f(x_k)$ are approximated rather than computed exactly.

If $H_k = \nabla^2 f(x_k)$ and s_k satisfies the fraction of optimal decrease (2.10) for every k, then it is also possible to analyze the global convergence of the Trust–Region Algorithm 2.3.3 to a point satisfying the second–order necessary optimality conditions.

Theorem 2.3.3 *(Moré and Sorensen [106], [132])* Let $\{x_k\}$ be a sequence generated by the Trust–Region Algorithm 2.3.3 with $H_k = \nabla^2 f(x_k)$ where s_k satisfies the fraction of optimal decrease (2.10). Let f be twice continuously differentiable and bounded below in the level set $\mathcal{L}(x_0)$. If the sequences $\{x_k\}$ and $\{H_k\}$ are bounded, then

$$\liminf_{k \to +\infty} \left(\|g_k\| + \gamma_k \right) = 0$$

and $\{x_k\}$ has a limit point x_* such that $\nabla^2 f(x_*)$ is positive semi–definite.

Moré [103] showed how to generalize these theorems for trust–region constraints of the form $\|S_k s\| \leq \delta_k$, where $\{S_k\}$ is a sequence of nonsingular scaling matrices. Related results can be found in references [56], [106], [130], [132].

2.3.4 Tikhonov Regularization

In this section we show how the Tikhonov regularization [138] for ill–conditioned linear least–squares is related to a particular trust–region subproblem. This is one of many arguments that justify the use of trust regions as a regularization technique. A different argument is given in the next section.

In many applications like reconstruction and parameter identification problems the objective function in (2.1) comes from the discretization of infinite dimensional problems of the form

$$\text{minimize} \qquad \|Ax - b\|_Y^2 \,, \tag{2.20}$$

where $x \in X$, $b \in Y$, and $A \in L(X,Y)$ is a linear bounded operator mapping the real Hilbert space X into the real Hilbert space Y. There are situations where, due to the lack of an inverse or a continuous inverse for A, the solution to (2.20) does not depend continuously on b (see for instance [69]). When a discretization is introduced this type of problem leads to finite dimensional problems of the form (2.1) where $f(x) = \|\bar{A}x - \bar{b}\|^2$ and $\bar{A}$ is ill–conditioned. (Here $\bar{A} \in \mathbb{R}^{m\times n}$ and $\bar{b} \in \mathbb{R}^m$, with $m > n$.)

A common technique to overcome this ill–posedness is the Tikhonov regularization. This regularization consists of solving a perturbed problem of the form

$$\text{minimize} \qquad \|Ax - b\|_Y^2 + \gamma\|Lx\|_X^2 \,, \tag{2.21}$$

where γ is a positive regularization parameter and L is in $L(X,X)$. To ensure the existence and uniqueness of the solution for (2.21), it is assumed that L is such that for every $\gamma > 0$ there exists a $c_\gamma > 0$ that satisfies $\|Ax\|_Y^2 + \gamma\|Lx\|_X^2 \geq c_\gamma\|x\|_X^2$ for all x in X. See [73].

One can see by looking at the gradient of $\|Ax-b\|_Y^2 + \gamma\|Lx\|_X^2$ that the Tikhonov regularization is strongly related to the trust–region subproblem in infinite dimensions:

$$\begin{aligned} \text{minimize} \quad & \|Ax - b\|_Y^2 \\ \text{subject to} \quad & \|Lx\|_X \leq \delta, \end{aligned} \tag{2.22}$$

where $\delta > 0$. In fact, if x_* is the solution for (2.22) with $\|Lx_*\|_X = \delta$, then x_* is the solution for (2.21) with $\gamma = \gamma_*$, where γ_* is the positive Lagrange multiplier for (2.22) associated with x_*. On the other hand, if x_* is the solution for (2.21) with $\gamma = \gamma_* > 0$, then x_* is the solution for (2.22) with $\delta = \|Lx_*\|_X$ and Lagrange multiplier γ_*.

2.4 More about Line Searches and Trust Regions

We now point out interesting relationships between line searches and trust regions.

A major difference between the global convergence results given in Corollary 2.2.1 and Theorem 2.3.2 is that a uniform bound on H_k^{-1} is required for line searches but not for trust regions. The study by Vicente [142] shows that this is related with the flexibility that trust–region algorithms have to choose the type of direction.

The criteria to accept a step in line searches and in trust regions are very similar. Suppose that a line search only requires the Armijo–Goldstein–Wolfe condition (2.3) to accept a step s_k. This condition can be rewritten as

$$\frac{f(x_k) - f(x_k + s_k)}{-g_k^T s_k} \geq \eta_1, \tag{2.23}$$

and it becomes evident how similar this is to the condition

$$\frac{f(x_k) - f(x_k + s_k)}{-g_k^T s_k - s_k^T H_k s_k} \geq \eta,$$

used in the trust–region technique. One can see that trust regions use curvature to accept or reject a step but line searches do not. However many practical implementations of line searches include second–order information in the sufficient decrease condition (2.23), i.e. the Armijo–Goldstein–Wolfe condition (2.3).

One final comment about the regularization issue is in order. It is also possible to regularize in a line search by adding to H_k a positive multiple γI_n of the identity matrix. Of course one must choose γ, and this becomes a performance issue that does not arise in trust–region algorithms. The solution o_k of the trust–region subproblem (2.2) satisfies the conditions given in Property 2.3.3 and the parameter γ is implicitly defined by the size of the trust–region radius δ_k.

Chapter 3

Trust–Region SQP Algorithms for Equality–Constrained Optimization

In this chapter, we address trust–region sequential quadratic programming (SQP) algorithms for the equality–constrained optimization problem

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \quad C(x) = 0, \end{aligned} \tag{3.1}$$

where $f : \mathbb{R}^n \longrightarrow \mathbb{R}$, $c_i : \mathbb{R}^n \longrightarrow \mathbb{R}$, $i = 1, \ldots, m$, $C(x) = \Big(c_1(x) \cdots c_m(x)\Big)^T$, and $m < n$. The functions $f(x)$ and $c_i(x)$, $i = 1, \ldots, m$, are assumed to be at least twice continuously differentiable in the domain of interest.

The material given in this chapter is useful to introduce the new trust–region interior–point reduced SQP algorithms in Chapters 5 and 6 and to understand the analysis given in Chapter 4 for a specific class of nonlinear programming problems. The organization of this chapter is the following. We start in Sections 3.1 and 3.2 by reviewing basic material for equality–constrained optimization, like the optimality conditions, the application of Newton's method, and SQP algorithms. The various trust–region globalizations suggested in the literature for these algorithms are surveyed in Section 3.3.

The algorithm that we focus on this chapter is very similar to the trust–region globalizations of the reduced SQP algorithm suggested and analyzed by Byrd and Omojokon [115] and Dennis, El–Alem, and Maciel [35]. It is described in great detail in Section 3.4. Then Sections 3.5 and 3.6 present the global convergence for this algorithm. The global convergence to a point satisfying the first–order necessary optimality conditions has been proved in [35]. Our contribution is to prove global convergence to a point satisfying the second–order necessary optimality conditions. See also Dennis and Vicente [42].

The conditions imposed to obtain this result are shown to be satisfied in Section 3.7 for the normal component and the least–squares multipliers. We point out that El–Alem [48] has proved the same global convergence result for a trust–region algorithm that uses the normal component, the least–squares multipliers, and a nonmonotone scheme to update the penalty parameter.

We finish this chapter in Section 3.8 with an analysis of the trust–region subproblem for the linearized constraints.

3.1 Basics of Equality–Constrained Optimization

To state optimality conditions for problem (3.1) a constraint qualification typically is required. We use a strong form of constraint qualification called regularity.

> **Definition 3.1.1** A point x_* is regular for problem (3.1) if the rows of the Jacobian matrix $J(x_*)$ are linearly independent.

In this chapter, we assume regularity. The Lagrangian function associated with problem (3.1) is given by

$$\ell(x, \lambda) = f(x) + \lambda^T C(x).$$

The matrix $W(x) \in \mathbb{R}^{n\times(n-m)}$ denotes a matrix whose columns form a basis for the null space $\mathcal{N}(J(x))$ of the Jacobian $J(x)$ of $C(x)$. The next two propositions review the optimality conditions for problem (3.1). For proofs and related material see the books [53], [60], [96], [112].

> **Proposition 3.1.1** *(First–Order Necessary Optimality Conditions)* If the regular point x_* is a local minimizer of (3.1), then there exists a $\lambda_* \in \mathbb{R}^m$ such that
>
> $$C(x_*) = 0 \ \text{ and}$$
> $$\nabla_x \ell(x_*, \lambda_*) = \nabla f(x_*) + J(x_*)^T \lambda_* = 0.$$

The vector λ_* is the vector of Lagrange multipliers. Although it is the name of Lagrange [90] that is associated with the optimality conditions for optimization problems with equality constraints, credit should be given also to Euler (see the discussion in [112][Chapter 14, Section 9]). In the eighteen century the two mathematicians solved problems in calculus of variations using optimality conditions for equality constraints.

A point x_* that satisfies the first–order necessary optimality conditions is called a stationary point.

Proposition 3.1.2 *(Second–Order Optimality Conditions)* If x_* is a regular point for (3.1), then second–order necessary (resp. sufficient) optimality conditions for x_* to be a local minimizer are the existence of a $\lambda_* \in \mathbb{R}^m$ such that

$$C(x_*) = 0,$$

$$\nabla_x \ell(x_*, \lambda_*) = \nabla f(x_*) + J(x_*)^T \lambda_* = 0, \text{ and}$$

$$\nabla^2_{xx} \ell(x_*, \lambda_*) \text{ is positive semi–definite (resp. definite) on } \mathcal{N}(J(x_*)).$$

From a basic result of linear algebra we can restate these conditions as follows.

Proposition 3.1.3 *(First–Order Necessary Optimality Conditions)* If the regular point x_* is a local minimizer of (3.1), then

$$C(x_*) = 0 \text{ and}$$

$$W(x_*)^T \nabla f(x_*) = 0.$$

Proposition 3.1.4 *(Second–Order Optimality Conditions)* If x_* is a regular point for (3.1), then second–order necessary (resp. sufficient) optimality conditions for x_* to be a local minimizer are the existence of a $\lambda_* \in \mathbb{R}^m$ such that

$$C(x_*) = 0,$$

$$W(x_*)^T \nabla f(x_*) = 0, \text{ and}$$

$$W(x_*)^T \nabla^2_{xx} \ell(x_*, \lambda_*) W(x_*) \text{ is positive semi–definite (resp. definite),}$$

where λ_* satisfies $\nabla_x \ell(x_*, \lambda_*) = \nabla f(x_*) + J(x_*)^T \lambda_* = 0$.

The optimality conditions given in Propositions 3.1.3 and 3.1.4 use the matrix $W(x_*)$ to reduce the gradient of f and the Hessian of the Lagrangian to the null space of $J(x_*)$.

3.2 SQP Algorithms

We describe now SQP and reduced* SQP algorithms for problem (3.1). SQP algorithms are very successful for the solution of constrained optimization problems. See e.g. [5], [59], [91], [108]. They are often quasi–Newton type algorithms in the sense that they rely on a Newton iteration and approximate second–order derivatives.

The primary goal of these algorithms is to find a point that satisfies the first–order necessary optimality conditions. So we proceed as in Chapter 2 and define at (x_k, λ_k) a quadratic model of $\ell(x_k + s, \lambda_k)$,

$$q_k(s) = \ell_k + \nabla_x \ell_k^T s + \frac{1}{2} s^T H_k s,$$

where H_k is a symmetric approximation to $\nabla^2_{xx}\ell_k$, and from our notation $\ell_k = \ell(x_k, \lambda_k)$. This quadratic model is then minimized subject to the linearized constraints:

$$J_k s + C_k = 0, \tag{3.2}$$

with $J_k = J(x_k)$ and $C_k = C(x_k)$. The basic SQP algorithm is described next.

Algorithm 3.2.1 *(Basic SQP Algorithm)*

1. Choose x_0 and λ_0.
2. For $k = 0, 1, 2, \ldots$ do
   2.1 Stop if (x_k, λ_k) satisfies the stopping criterion.
   2.2 Compute the step s_k as an approximate solution of
   $$\begin{array}{rl} \text{minimize} & \ell_k + \nabla_x \ell_k^T s + \frac{1}{2} s^T H_k s \\ \text{subject to} & J_k s + C_k = 0. \end{array} \tag{3.3}$$
   2.3 Set $x_{k+1} = x_k + s_k$ and $\lambda_{k+1} = \lambda_k + \Delta\lambda_k$, where $\Delta\lambda_k$ are the multipliers associated with the quadratic programming subproblem (3.3).

*We prefer to call these algorithms reduced SQP instead of reduced Hessian SQP. For us, reduced SQP means that the step is decomposed into two components, and one of them is reduced to the null space of the Jacobian matrix of the equality constraints.

The stopping criterion might be $\|\nabla_x \ell_k\| + \|C_k\| \le \epsilon_{tol}$ for a given $\epsilon_{tol} > 0$.

Suppose that the point x_k is regular, $H_k = \nabla^2_{xx}\ell_k$, and $\nabla^2_{xx}\ell_k$ is positive definite on $\mathcal{N}(J_k)$. Then the solution s_k and the corresponding multipliers $\Delta\lambda_k$ of the quadratic programming subproblem (3.3) are equal to the Newton step on the system of first–order necessary optimality conditions

$$\nabla f(x) + J(x)^T \lambda = 0,$$
$$C(x) = 0,$$

given by the solution of the linear system

$$\begin{pmatrix} \nabla^2_{xx}\ell_k & J_k^T \\ J_k & 0 \end{pmatrix} \begin{pmatrix} s \\ \Delta\lambda \end{pmatrix} = \begin{pmatrix} -\nabla_x \ell_k \\ -C_k \end{pmatrix}. \tag{3.4}$$

See Boggs [5] for an extensive survey on SQP algorithms.

In order to present the basic reduced SQP algorithm used here, we consider a *quasi–normal* decomposition of the step s_k of the form

$$s_k = s_k^{\mathsf{q}} + s_k^{\mathsf{t}}. \tag{3.5}$$

The component s_k^{q} is called the quasi–normal (or quasi–vertical) component, and it is a solution for the linearized constraints (3.2). The component s_k^{t} is the tangential (or horizontal) component, and it must satisfy $J_k s_k^{\mathsf{t}} = 0$, i.e. it must lie in the null space of J_k. Hence this component is of the form $s_k^{\mathsf{t}} = W_k \bar{s}_k^{\mathsf{t}}$ for some $\bar{s}_k^{\mathsf{t}} \in \mathbb{R}^{n-m}$. Here $W_k = W(x_k)$ represents a basis for the null space $\mathcal{N}(J_k)$. Given the component s_k^{q}, the quadratic $q_k(s)$ depends uniquely on $\bar{s}^{\mathsf{t}}$ in the following way:

$$\bar{q}_k(\bar{s}^{\mathsf{t}}) \equiv q_k(s_k^{\mathsf{q}} + W_k \bar{s}^{\mathsf{t}}) = q_k(s_k^{\mathsf{q}}) + \bar{g}_k^T \bar{s}^{\mathsf{t}} + \frac{1}{2}(\bar{s}^{\mathsf{t}})^T \bar{H}_k (\bar{s}^{\mathsf{t}})$$

with

$$\begin{aligned} \bar{H}_k &= W_k^T H_k W_k, \\ \bar{g}_k &= W_k^T \nabla q_k(s_k^{\mathsf{q}}) \\ &= W_k^T \left(H_k s_k^{\mathsf{q}} + \nabla f_k \right), \text{ and} \\ q_k(s_k^{\mathsf{q}}) &= \ell_k + \nabla_x {\ell_k}^T s_k^{\mathsf{q}} + \tfrac{1}{2}(s_k^{\mathsf{q}})^T H_k (s_k^{\mathsf{q}}). \end{aligned}$$

The basic reduced SQP algorithm follows.

Algorithm 3.2.2 *(Basic Reduced SQP Algorithm)*

1. Choose x_0 and λ_0.
2. For $k = 0, 1, 2, \ldots$ do

   2.1 Stop if (x_k, λ_k) satisfies the stopping criterion.

   2.2 Compute s_k^{q} as an approximate solution of $J_k s^{\mathsf{q}} + C_k = 0$.

   2.3 Compute $\bar{s}_k^{\mathsf{t}}$ as an approximate solution of

   $$\text{minimize} \qquad q_k(s_k^{\mathsf{q}}) + \bar{g}_k^T \bar{s}^{\mathsf{t}} + \frac{1}{2}(\bar{s}^{\mathsf{t}})^T \bar{H}_k(\bar{s}^{\mathsf{t}}) \; .$$

   2.4 Set $x_{k+1} = x_k + s_k = x_k + s_k^{\mathsf{q}} + W_k \bar{s}_k^{\mathsf{t}}$ and compute λ_{k+1}.

The algorithm is stopped if for instance $\|\bar{g}_k\| + \|C_k\| \leq \epsilon_{tol}$ for some $\epsilon_{tol} > 0$.

An advantage of reduced SQP algorithms over SQP algorithms is that they allow a secant update $\tilde{H}_k$ of the reduced Hessian $W_k^T \nabla_{xx}^2 \ell_k W_k$. The dimension of $W_k^T \nabla_{xx}^2 \ell_k W_k$ is usually much smaller than the dimension of $\nabla_{xx}^2 \ell_k$. Furthermore, $W(x_*)^T \nabla_{xx}^2 \ell(x_*, \lambda_*) W(x_*)$ is positive definite at a point (x_*, λ_*) satisfying the second–order sufficient optimality conditions. This suggests that we can update $\tilde{H}_{k+1}$ from $\tilde{H}_k$ by using positive definite secant updates like the very effective BFGS secant update†. However, if an approximation H_k of the full Hessian $\nabla_{xx}^2 \ell_k$ is not computed then the evaluation of the *cross term* $W_k^T \nabla_{xx}^2 \ell_k s_k^{\mathsf{q}}$ becomes a serious issue. This cross term can be approximated by finite differences, by secant updates, or by zero [4]. There has been significant activity in studying the local rate of convergence of secant updates for reduced SQP algorithms. See the papers [4], [114], [147] and the references therein.

---

†BFGS is an abbreviation for the names Broyden, Fletcher, Goldfarb, and Shanno who in 1970 independently discovered this secant update. In unconstrained optimization, for instance, BFGS updates H_{k+1} by a rank two modification of H_k of the form

$$H_{k+1} = H_k + \frac{y_k y_k^T}{y_k^T s_k} - \frac{H_k s_k s_k^T H_k}{s_k^T H_k s_k},$$

where $s_k = x_{k+1} - x_k$ and $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$. If H_k is positive definite and $y_k^T s_k > 0$, then H_{k+1} is also positive definite. The fundamental material about secant updates can be found in the classical references [38], [39].

The Normal Decomposition

A popular step decomposition, which amounts to special choices for s_k^{q} and W_k, is the *normal decomposition*:

$$s_k = s_k^{\mathsf{n}} + s_k^{\mathsf{t}} = s_k^{\mathsf{n}} + Z_k \bar{s}_k^{\mathsf{t}}, \tag{3.6}$$

s_k^{n} is the minimum norm solution of the linearized constraints, and

the columns of Z_k form an orthogonal basis for $\mathcal{N}(J_k)$.

The matrix Z_k can be computed from the QR factorization of J_k^T. This factorization is of the form:

$$J_k^T = \begin{pmatrix} Y_k & Z_k \end{pmatrix} \begin{pmatrix} R_k \\ 0 \end{pmatrix}, \tag{3.7}$$

where $\begin{pmatrix} Y_k & Z_k \end{pmatrix}$ is orthogonal and R_k upper triangular and nonsingular. The normal component s_k^{n} is then given by

$$s_k^{\mathsf{n}} = -J_k^T (J_k J_k^T)^{-1} C_k = -Y_k R_k^{-T} C_k. \tag{3.8}$$

Associated with the normal decomposition is the least–squares multiplier update. These multipliers are the solution of the linear least–squares problem

$$\text{minimize} \qquad \left\| \nabla f_k + J_k^T \lambda \right\|$$

and are given by

$$\lambda_k = -(J_k J_k^T)^{-1} J_k \nabla f_k = -R_k^{-1} Y_k^T \nabla f_k. \tag{3.9}$$

It is easy to show (see e.g. [114]) that the Newton step $(s_k, \Delta\lambda_k)$ obtained by solving (3.4) can be expressed as follows:

$$s_k = s_k^{\mathsf{n}} + Z_k \bar{s}_k^{\mathsf{t}}, \tag{3.10}$$

$$s_k^{\mathsf{n}} = -J_k^T (J_k J_k^T)^{-1} C_k, \tag{3.11}$$

$$\bar{s}_k^{\mathsf{t}} = -\left(Z_k^T \nabla_{xx}^2 \ell_k Z_k \right)^{-1} Z_k^T \left(\nabla_{xx}^2 \ell_k s_k^{\mathsf{n}} + \nabla f_k \right), \tag{3.12}$$

$$\lambda_{k+1} = \Delta\lambda_k + \lambda_k = -(J_k J_k^T)^{-1} J_k \left(\nabla_{xx}^2 \ell_k s_k + \nabla f_k \right).$$

The q–quadratic rate of convergence‡ here is for the pair (x_k, λ_k). However a q–quadratic rate convergence in x_k can be obtained also by using (3.10)–(3.12) with the least–squares multipliers (3.9). An elegant proof of this latter result was provided by Goodman [68]. He showed that the iterates generated by (3.10)–(3.12), (3.9) can be seen as the result of applying Newton's method to

$$Z(x)^T \nabla f(x) = 0,$$

$$C(x) = 0,$$

where $Z(x)$ is a smooth extension of the orthogonal matrix provided by the QR factorization of $J(x)^T$.

3.3 Trust–Region Globalizations

Since the mid eighties a significant effort has been made to globalize SQP algorithms with trust regions.

Globalizations of SQP algorithms were given by Celis, Dennis, and Tapia [21] (see also Yuan [152] and Zhang [157]), Conn, Gould, and Toint [30], El–Alem [47], Fletcher [52], Vardi [141] (see also El–Hallabi [51]), and Powell and Yuan [123].

The reduced SQP algorithm has been globalized with trust regions by Byrd and Omojokon [115], Byrd, Schnabel, and Shultz [17], Coleman and Yuan [27], Dennis, El–Alem, and Maciel [35], Dennis and Vicente [42], El–Alem [48], [49], Lalee, Nocedal, and Plantenga [91], Plantenga [118], and Zhang and Zhu [156]. See also Alexandrov [1].

We recommend the surveys given in [35] and [118] for an overview of these different trust–region globalizations. Trust–region algorithms have been applied also to optimization problems with equality and inequality constraints. See the work by Burke [13], Burke, Moré, and Toraldo [14], Conn, Gould, and Toint [29], [30], and Yuan [154].

In this thesis we deal with a trust–region globalization of reduced SQP algorithms. The fundamental questions associated with the application of trust regions to reduced

‡We say that the sequence of vectors $\{z_k\}$ converges q–quadratically to z_* if there exists a positive constant c, independent of k, such that $\|z_{k+1} - z_*\| \le c\,\|z_k - z_*\|^2$ for all k. The letter q stands for *quotient* and distinguishes the q–quadratic rate from the r–rate, where r stands for *root*. See [116][Chapter 9].

SQP algorithms are the form of trust–region subproblems, the type of decomposition of the step, the choice of Lagrange multipliers, and the choice of the merit function. We address these issues in the following points.

1. The choice of trust–region subproblems now seems a settled question. Most of the references cited for trust–region reduced SQP algorithms [35], [42], [48], [49], [91], [115], [118] consider essentially the same choice of trust–region subproblems that was introduced first by Byrd and Omojokon [115][§]. We focus on this issue in Section 3.4.

2. The decomposition of the step considered in references [17], [27], [48], [49], [91], [115], [118], [156] is the normal decomposition (3.6).

   In many application problems there are other reasonable decompositions of the step. This is clearly the case for the class of problems introduced in Chapter 4. One important feature of these decompositions is that s^{q} is not orthogonal to $\mathcal{N}(J(x))$ and that $W(x)$ does not have orthogonal columns. We called such decompositions quasi–normal. In the context of trust regions this was addressed first in Dennis, El–Alem, and Maciel [35] and later in Dennis and Vicente [42].

   The algorithms we introduce in this thesis use a quasi–normal decomposition.

3. The choice of Lagrange multipliers is associated intimately with the type of step decomposition. Most of the researchers [17], [27], [48], [49], [91], [115], [118], [156] considered the least–squares multipliers (3.9) or variations thereof.

   The work given in [35], [42] departs from the former references by assuming a more general form for the multipliers. For example, in the class of problems described in Chapter 4, the most reasonable choice of multipliers is not the least–squares update but the so–called adjoint update.

[§]The Thesis [115] was directed by Professor R. H. Byrd. The trust–region algorithm proposed here is usually referred as the Byrd and Omojokon algorithm.

4. The choice of merit function has been always an open question. The following merit functions have been used in this context:

   $\ell(x,\lambda) + \rho\|C(x)\|^2$ (Augmented Lagrangian),

   $f(x) + \sum_{i=1}^{m} \rho_i |c_i(x)|$ (ℓ_1 Penalty function),

   $f(x) + \rho\|C(x)\|^2$ (ℓ_2 Penalty function), and

   $f(x) + \rho\|C(x)\|$ (ℓ_2 Penalty function without constraint term squared),

   where the ρ's denote weights or penalty parameters. The augmented Lagrangian has been used in [35], [42], [48], [49], [156], the ℓ_1 penalty function in [17], the ℓ_2 penalty function in [27], and the ℓ_2 penalty function without constraint term squared in [91], [115], [118].

Let us describe briefly the trust–region globalization analyzed by Dennis, El–Alem, and Maciel [35]. The components of the step s_k^{q} and $\bar{s}_k^{\mathsf{t}}$ are only required to satisfy a fraction of Cauchy decrease (or simple decrease) on the corresponding trust–region subproblem. A key assumption that is imposed on the quasi–normal component s_k^{q} is that it has to be $\mathcal{O}(\|C_k\|)$. In this globalization the augmented Lagrangian is used as a merit function combined with the El–Alem's scheme [47] to update the penalty parameter. The main result proved in [35] is global convergence to a stationary point (see Theorem 3.6.1). It is important to remark that this result is obtained under very mild conditions on the components of the step, on the multipliers estimates, and on the Hessian approximations. Thus, the Dennis, El–Alem, and Maciel [35] result is similar to the result given by Powell [121] for unconstrained optimization and described in Theorem 2.3.1 (see Figure 1.1).

One of the purposes of this chapter is to analyze under what modifications and conditions this trust–region reduced SQP algorithm possesses global convergence to a point that satisfies the second–order necessary optimality conditions. Our goal is to generalize the result given by Moré and Sorensen [106], [132] for unconstrained optimization and described in Theorem 2.3.3 (see Figure 1.2). We accomplish this by imposing a fraction of optimal decrease on the tangential component $\bar{s}_k^{\mathsf{t}}$ of the step, by using exact second–order derivatives, and by imposing conditions on the quasi–normal component s_k^{q} and on the Lagrange multipliers. These conditions are the following:

$$\nabla_x \ell_k^T s_k^{\mathsf{q}} \ \text{is} \ \mathcal{O}(\delta_k \|C_k\|) \ \text{and} \ \|\Delta\lambda_k\| = \|\lambda_{k+1} - \lambda_k\| \ \text{is} \ \mathcal{O}(\delta_k). \tag{3.13}$$

In the case where $\|C_k\|$ is small compared with δ_k, the first condition implies that any increase of the quadratic model $q_k(s)$ of the Lagrangian from x_k to $x_k + s_k^{\mathsf{q}}$ is $\mathcal{O}(\delta_k^2)$. To see why this is relevant recall that a fraction of optimal decrease is being imposed on the tangential component $\bar{s}_k^{\mathsf{t}}$ and from Lemma 2.3.2 this yields a decrease of at least $\mathcal{O}(\delta_k^2)$ on the quadratic model. The second condition is needed for the same reasons because $\Delta\lambda_k$ also appears in the definition of the predicted decrease used in the trust–region reduced SQP algorithm. See also [42].

Gill, Murray, and Wright [61] and El–Alem [46] considered in their analyses that $\nabla_x \ell_k$ is $\mathcal{O}(\|s_k\|)$. In the latter work this assumption is used to prove local convergence results, and in the former to establish properties of an augmented Lagrangian merit function. We point out that this assumption implies that $\nabla_x \ell_k^T s_k^{\mathsf{q}}$ is $\mathcal{O}(\delta_k \|C_k\|)$ since s_k is $\mathcal{O}(\delta_k)$ and we assume that s_k^{q} is $\mathcal{O}(\|C_k\|)$.

We show that both conditions in (3.13) are satisfied when the normal component and the least–squares multipliers are used. This is in agreement with the result obtained by El–Alem [48]. We show in Chapter 5 that these conditions are satisfied also for all reasonable choices of quasi–normal components and multipliers for the class of nonlinear programming problems introduced in Chapter 4 (see Remark 5.2.1). This class of problems arises in many applications, in particular from the discretization of optimal control problems.

3.4 A General Trust–Region Globalization of the Reduced SQP Algorithm

The trust–region globalization of the Reduced SQP Algorithm 3.2.2 that we consider consists of computing the components s_k^{q} and $\bar{s}_k^{\mathsf{t}}$ as approximate solutions of particular trust–region subproblems.

3.4.1 The Quasi–Normal Component

The component s_k^{q} is computed as an approximate solution of the trust–region subproblem for the linearized constraints defined by

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|J_k s^{\mathsf{q}} + C_k\|^2 \\ \text{subject to} & \|s^{\mathsf{q}}\| \le \delta_k, \end{array} \tag{3.14}$$

where δ_k is the trust radius.

To guarantee global convergence we require s_k^{q} to satisfy

$$\|s_k^{\mathsf{q}}\| \le \kappa_1 \|C_k\|, \tag{3.15}$$

where κ_1 is a positive constant independent of the iterate k of the algorithm. This condition is saying that close to feasibility the quasi–normal component has to be small.

As we described in Section 2.3, s_k^{q} satisfies a fraction of Cauchy decrease (or simple decrease) for the trust–region subproblem (3.14) if

$$\begin{aligned} &\|C_k\|^2 - \|J_k s_k^{\mathsf{q}} + C_k\|^2 \ge \beta_1^{\mathsf{q}} \left(\|C_k\|^2 - \|J_k\, c_k^{\mathsf{q}} + C_k\|^2 \right), \\ &\|s_k^{\mathsf{q}}\| \le \delta_k, \end{aligned} \tag{3.16}$$

where $\beta_1^{\mathsf{q}} > 0$ does not depend on k and c_k^{q} is the so–called Cauchy step for this trust–region subproblem, i.e. c_k^{q} is the optimal solution of

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|J_k c^{\mathsf{q}} + C_k\|^2 \\ \text{subject to} & \|c^{\mathsf{q}}\| \le \delta_k, \ c^{\mathsf{q}} \in span\{-J_k^T C_k\}, \end{array}$$

and therefore

$$c_k^{\mathsf{q}} = \begin{cases} -\frac{\|J_k^T C_k\|^2}{\|J_k J_k^T C_k\|^2} J_k^T C_k & \text{if } \frac{\|J_k^T C_k\|^3}{\|J_k J_k^T C_k\|^2} \le \delta_k, \\ -\frac{\delta_k}{\|J_k^T C_k\|} J_k^T C_k & \text{otherwise.} \end{cases}$$

If s_k^{q} satisfies the Cauchy decrease condition (3.16), then we can apply Lemma 2.3.1 and conclude that the decrease given by s_k^{q} is such that

$$\|C_k\|^2 - \|J_k s_k^{\mathsf{q}} + C_k\|^2 \ge \frac{\beta_1^{\mathsf{q}}}{2} \|J_k^T C_k\| \min\left\{ \frac{\|J_k^T C_k\|}{\|J_k^T J_k\|}, \ \delta_k \right\}. \tag{3.17}$$

To prove global convergence of the general trust–region reduced SQP algorithm to a stationary point we require s_k^{q} to satisfy a simpler decrease condition. This condition relates the decrease given by s_k^{q} on $\|J_k s^{\mathsf{q}} + C_k\|^2$ with the vector C_k and not with the gradient $J_k^T C_k$ of this least–squares functional. It can be stated as follows

$$\begin{aligned} &\|C_k\|^2 - \|J_k s_k^{\mathsf{q}} + C_k\|^2 \ge \kappa_2 \|C_k\| \min\left\{\kappa_3 \|C_k\|, \delta_k\right\}, \\ &\|s_k^{\mathsf{q}}\| \le \delta_k, \end{aligned} \tag{3.18}$$

where κ_2 and κ_3 are positive constants independent of k. It is not difficult to show that if J_k, $J_k^T J_k$, and $(J_k J_k^T)^{-1}$ are uniformly bounded then the Cauchy decrease condition (3.17) implies the decrease condition (3.18).

If global convergence to a point that satisfies second–order necessary optimality conditions is the goal of the trust–region reduced SQP algorithm, then we need to impose also on the component s_k^{q} the condition

$$\nabla_x \ell_k^T s_k^{\mathsf{q}} \le \kappa_4 \|C_k\| \delta_k, \tag{3.19}$$

where κ_4 is a positive constant independent of the iterates. The important consequence of this condition is that if $\|C_k\|$ is small compared with δ_k, then any increase of the quadratic model $q_k(s)$ of the Lagrangian along the quasi–normal component s_k^{q} is of $\mathcal{O}(\delta_k^2)$. See inequality (3.36).

3.4.2 The Tangential Component

We suggest two approaches to compute the tangential component. They are called *decoupled* and *coupled* and differ in the type of trust–region constraint.

The Decoupled Trust–Region Approach

In this case the tangential component is computed from the trust–region subproblem

$$\begin{aligned} \text{minimize} \quad & \bar{q}_k(\bar{s}^{\mathsf{t}}) \\ \text{subject to} \quad & \|\bar{s}^{\mathsf{t}}\| \le \delta_k. \end{aligned} \tag{3.20}$$

To assure global convergence to a stationary point the component $\bar{s}_k^{\mathsf{t}}$ is required to satisfy a fraction of Cauchy decrease (or simple decrease) for the trust–region subproblem (3.20). The Cauchy step c_k^{d} for this trust–region subproblem is defined as the solution of

$$\begin{aligned} \text{minimize} \quad & \bar{q}_k(c^{\mathsf{d}}) \\ \text{subject to} \quad & \|c^{\mathsf{d}}\| \le \delta_k, \; c^{\mathsf{d}} \in span\{-\bar{g}_k\}. \end{aligned}$$

The fraction of Cauchy decrease condition that $\bar{s}_k^{\mathsf{t}}$ has to satisfy is

$$\begin{aligned} & \bar{q}_k(0) - \bar{q}_k(\bar{s}_k^{\mathsf{t}}) \ge \beta_1^{\mathsf{d}} \left(\bar{q}_k(0) - \bar{q}_k(c_k^{\mathsf{d}}) \right), \\ & \|\bar{s}_k^{\mathsf{t}}\| \le \delta_k, \end{aligned} \tag{3.21}$$

where β_1^{d} is some positive constant independent of k.

To guarantee global convergence to a point that satisfies the second–order necessary optimality conditions, the component $\bar{s}_k^{\mathsf{t}}$ has to satisfy a fraction of optimal decrease for the trust–region subproblem (3.20). This condition is as follows:

$$\begin{aligned} &\bar{q}_k(0) - \bar{q}_k(\bar{s}_k^{\mathsf{t}}) \geq \beta_2^{\mathsf{d}} \left(\bar{q}_k(0) - \bar{q}_k(o_k^{\mathsf{d}}) \right), \\ &\|\bar{s}_k^{\mathsf{t}}\| \leq \beta_3^{\mathsf{d}} \delta_k, \end{aligned} \tag{3.22}$$

where o_k^{d} is the optimal solution of (3.20) and β_2^{d}, $\beta_3^{\mathsf{d}} > 0$ are positive constants independent of k.

The Coupled Trust–Region Approach

In this approach the tangential component is computed from the trust–region subproblem

$$\begin{aligned} \text{minimize} \quad & \bar{q}_k(\bar{s}^{\mathsf{t}}) \\ \text{subject to} \quad & \|W_k \bar{s}^{\mathsf{t}}\| \leq \delta_k. \end{aligned} \tag{3.23}$$

This subproblem differs from (3.20) in the form of the trust–region constraint. In the trust–region subproblem (3.20) the constraint is $\|\bar{s}^{\mathsf{t}}\| \leq \delta_k$ and does not force the whole tangential component $W_k \bar{s}_k^{\mathsf{t}}$ to lie inside the trust–region. The coupled approach offers a better regularization of the tangential component in the cases where W_k is ill–conditioned. This point is better explained in Section 5.2.2 by using a particular form of W_k. The components s_k^{q} and s_k^{t} for this approach are depicted in Figure 3.1. It is quite clear from the picture that s_k might not lie inside the trust region $\{s : \|s\| \leq \delta_k\}$. Of course the same thing happens in the decoupled approach but here there is even no guarantee that the tangential component s_k^{t} is itself inside the trust region.

If global convergence to a stationary point is the goal of the trust–region reduced SQP algorithm, then $\bar{s}_k^{\mathsf{t}}$ is required to satisfy a fraction of Cauchy decrease (or simple decrease) for the trust–region subproblem (3.23). We discuss this point now.

The steepest–descent direction at $\bar{s}^{\mathsf{t}} = 0$ associated with $\bar{q}_k(\bar{s}^{\mathsf{t}})$ in the ℓ_2 norm is $-\bar{g}_k$. See Section 2.3. If we take into account the matrix W_k, then the steepest–descent direction in the $\|W_k \cdot \|$ norm is given by $-(W_k^T W_k)^{-1} \bar{g}_k$. We consider the steepest–descent direction $-\bar{g}_k$ and require $\bar{s}_k^{\mathsf{t}}$ to satisfy the Cauchy condition

$$\begin{aligned} &\bar{q}_k(0) - \bar{q}_k(\bar{s}_k^{\mathsf{t}}) \geq \beta_1^{\mathsf{c}} \Big(\bar{q}_k(0) - \bar{q}_k(c_k^{\mathsf{c}}) \Big), \\ &\|\bar{s}_k^{\mathsf{t}}\| \leq \delta_k, \end{aligned} \tag{3.24}$$

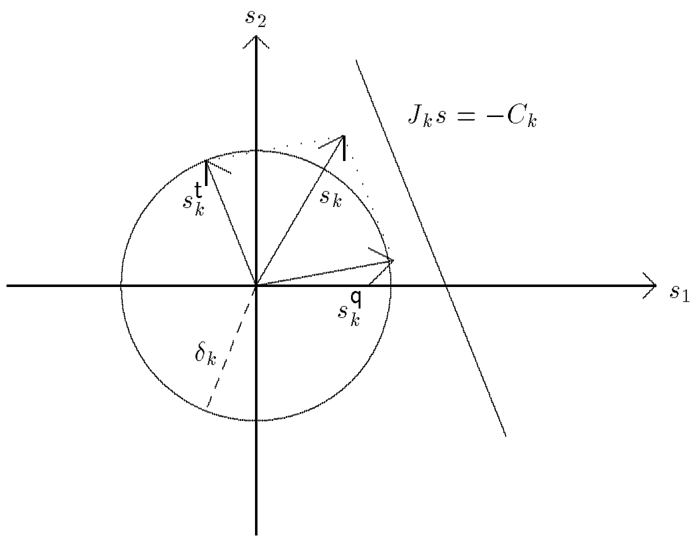

Figure 3.1 The quasi–normal and tangential components of the step for the coupled approach.

where β_1^{c} is a positive constant independent of k and c_k^{c} is the Cauchy step that solves

$$\begin{array}{ll} \text{minimize} & \bar{q}_k(c^{\mathsf{c}}) \\ \text{subject to} & \|W_k c^{\mathsf{c}}\| \leq \delta_k, \; c^{\mathsf{c}} \in span\{-\bar{g}_k\}. \end{array}$$

The results given in this chapter hold also if c_k^{c} is defined along $-(W_k^T W_k)^{-1}\bar{g}_k$ provided the sequence $\{\|(W_k^T W_k)^{-1}\|\}$ is bounded.

In order to establish global convergence to a point that satisfies the second–order necessary optimality conditions, we need $\bar{s}_k^{\mathsf{t}}$ to satisfy a fraction of optimal decrease for the trust–region subproblem (3.23). This condition is as follows:

$$\begin{aligned} &\bar{q}_k(0) - \bar{q}_k(\bar{s}_k^{\mathsf{t}}) \geq \beta_2^{\mathsf{c}}\Big(\bar{q}_k(0) - \bar{q}_k(o_k^{\mathsf{c}})\Big), \\ &\|W_k \bar{s}_k^{\mathsf{t}}\| \leq \beta_3^{\mathsf{c}} \delta_k, \end{aligned} \tag{3.25}$$

where o_k^{c} is the optimal solution of (3.23) and β_2^{c}, $\beta_3^{\mathsf{c}} > 0$ are positive constants independent of k.

3.4.3 Outline of the Algorithm

We introduce now the merit function and the corresponding actual and predicted decreases. The merit function used is the augmented Lagrangian

$$L(x, \lambda; \rho) = f(x) + \lambda^T C(x) + \rho C(x)^T C(x),$$

where ρ is the penalty parameter. The actual decrease $ared(s_k; \rho_k)$ at the iteration k is given by

$$ared(s_k; \rho_k) = L(x_k, \lambda_k; \rho_k) - L(x_{k+1}, \lambda_{k+1}; \rho_k).$$

The predicted decrease (see [35]) is the following:

$$pred(s_k; \rho_k) = L(x_k, \lambda_k; \rho_k) - \left(q_k(s_k) + \Delta\lambda_k^T(J_k s_k + C_k) + \rho_k \|J_k s_k + C_k\|^2\right).$$

Other forms of predicted decrease were proposed in the literature that use the augmented Lagrangian as a merit function. See El–Alem [49] and the references therein.

To update the penalty parameter ρ_k we use the scheme proposed by El–Alem [47]. This scheme is Step 2.4 of Algorithm 3.4.1 below. Other schemes to update the penalty parameter were suggested in the literature. El–Alem [48], [49] proposed a nonmonotone scheme to update the penalty parameter for which he proved many convergence results, including global convergence to points satisfying the second–order necessary optimality conditions. Lalee, Nocedal, and Plantenga [91] proposed and tested another nonmonotone scheme for the penalty parameter, but they did not provide any convergence analysis.

The general reduced trust–region SQP algorithm is given below.

Algorithm 3.4.1 *(Trust–Region Reduced SQP Algorithm)*

1 Choose x_0, δ_0, and λ_0. Set $\rho_{-1} \geq 1$. Choose α_1, η_1, δ_{min}, δ_{max}, and $\bar{\rho}$ such that $0 < \alpha_1$, $\eta_1 < 1$, $0 < \delta_{min} \leq \delta_{max}$, and $\bar{\rho} > 0$.

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

2.1 Stop if (x_k, λ_k) satisfies the stopping criterion.

2.2 Compute s_k^{q} based on the subproblem (3.14).
Compute $\bar{s}_k^{\mathsf{t}}$ based on the subproblem (3.20) (or subproblem (3.23) in the coupled case).
Set $s_k = s_k^{\mathsf{q}} + W_k \bar{s}_k^{\mathsf{t}}$.

2.3 Compute λ_{k+1} and set $\Delta\lambda_k = \lambda_{k+1} - \lambda_k$.

2.4 Compute $pred(s_k; \rho_{k-1})$:

$$q_k(0) - q_k(s_k) - \Delta\lambda_k^T(J_k s_k + C_k) + \rho_{k-1}\Big(\|C_k\|^2 - \|J_k s_k + C_k\|^2\Big).$$

If $pred(s_k; \rho_{k-1}) \geq \frac{\rho_{k-1}}{2}\Big(\|C_k\|^2 - \|J_k s_k + C_k\|^2\Big)$ then set $\rho_k = \rho_{k-1}$. Otherwise set

$$\rho_k = 2\left(\frac{q_k(s_k) - q_k(0) + \Delta\lambda_k^T(J_k s_k + C_k)}{\|C_k\|^2 - \|J_k s_k + C_k\|^2}\right) + \bar{\rho}.$$

2.5 If $\frac{ared(s_k;\rho_k)}{pred(s_k;\rho_k)} < \eta_1$, set

$$\delta_{k+1} = \alpha_1 \max\Big\{\|s_k^{\mathsf{q}}\|, \|(s_k)_u\|\Big\} \text{ in the decoupled case or}$$

$$\delta_{k+1} = \alpha_1 \max\Big\{\|s_k^{\mathsf{q}}\|, \|W_k(s_k)_u\|\Big\} \text{ in the coupled case,}$$

and reject s_k.
Otherwise accept s_k and choose δ_{k+1} such that

$$\max\{\delta_{min}, \delta_k\} \leq \delta_{k+1} \leq \delta_{max}.$$

2.6 If s_k was rejected set $x_{k+1} = x_k$ and $\lambda_{k+1} = \lambda_k$. Otherwise set $x_{k+1} = x_k + s_k$ and $\lambda_{k+1} = \lambda_k + \Delta\lambda_k$.

A reasonable stopping criterion for global convergence to a stationary point is $\|\bar{g}_k\| + \|C_k\| \leq \epsilon_{tol}$ for a given $\epsilon_{tol} > 0$. If global convergence to a point satisfying the second–order necessary optimality conditions is the goal of the algorithm, then the stopping criterion should look like $\|\bar{g}_k\| + \|C_k\| + \gamma_k \leq \epsilon_{tol}$, where γ_k is the Lagrange multiplier associated with the trust–region constraint in (3.20) and (3.23) (see equations (3.29) and (3.31)).

It is important to understand that the role of δ_{min} is just to reset δ_k after a step s_k has been accepted. During the course of finding such a step the trust radius can be decreased below δ_{min}. To our knowledge Zhang, Kim, and Lasdon [155] were the first to suggest this modification. We remark that the rules to update the trust radius in the previous algorithm can be much more complicated but these suffice to prove convergence results and to understand the trust–region mechanism.

3.4.4 General Assumptions

In order to establish global convergence results, we use the general assumptions given in [35]. Let Ω be an open subset of $\mathbb{R}^n$ such that for all iterations k, x_k and $x_k + s_k$ are in Ω.

Assumptions 3.1–3.5

3.1 The functions f, c_i, $i = 1, \ldots, m$ are twice continuously differentiable functions in Ω.

3.2 The Jacobian matrix $J(x)$ has full row rank in Ω.

3.3 The functions f, ∇f, $\nabla^2 f$, C, J, and $\nabla^2 c_i$, $i = 1, \ldots, m$, are bounded in Ω.

3.4 The sequences $\{W_k\}$, $\{H_k\}$, and $\{\lambda_k\}$ are bounded.

3.5 The matrix $(J(x)J(x)^T)^{-1}$ is uniformly bounded in Ω.

Assumptions 3.3 and 3.4 are equivalent to the existence of positive constants $\nu_0, \ldots, \nu_8$ such that

$$\begin{array}{llll} |f(x)| \le \nu_0, & \|\nabla f(x)\| \le \nu_1, & \|\nabla^2 f(x)\| \le \nu_2, & \|C(x)\| \le \nu_3, \\ \|J(x)\| \le \nu_4, & \text{and} & \|\nabla^2 c_i(x)\| \le \nu_5, & i = 1, \ldots, m, \end{array}$$

for all $x \in \Omega$, and

$$\|W_k\| \le \nu_6, \quad \|H_k\| \le \nu_7, \text{ and } \quad \|\lambda_k\| \le \nu_8,$$

for all k.

If Algorithm 3.4.1 is particularized to satisfy the following conditions on the steps, on the quadratic model, and on the Lagrange multipliers, then we can prove global convergence to a point satisfying the second–order necessary optimality conditions.

Conditions 3.1–3.2

3.1 The quasi–normal component s_k^{q} satisfies the feasibility condition (3.15) and the decrease condition (3.18).

The tangential component $\bar{s}_k^{\mathsf{t}}$ satisfies the fraction of Cauchy decrease condition (3.21) (or (3.24) in the coupled case).

3.2 The quasi–normal component s_k^{q} satisfies condition (3.19).

The tangential component satisfies the fraction of optimal decrease condition (3.22) (or (3.25) in the coupled case).

The Lagrange multipliers λ_k satisfy

$$\|\Delta\lambda_k\| = \|\lambda_{k+1} - \lambda_k\| \leq \kappa_5 \delta_k, \tag{3.26}$$

The Hessian approximation H_k is exact, i.e. $H_k = \nabla^2_{xx}\ell_k$ for all k.

The Hessians $\nabla^2 f$ and $\nabla^2 c_i$, $i = 1, \ldots, m$ are Lipschitz continuous.

Condition 3.1 is required for global convergence to a stationary point. Conditions 3.1–3.2 are imposed to achieve global convergence to a point that satisfies the second–order necessary optimality conditions.

3.5 Intermediate Results

In this section we list some technical results that are needed for the global convergence theory.

We start by pointing out that the decrease condition (3.18) and the fact that s_k^{t} is in $\mathcal{N}(J_k)$ imply

$$\|C_k\|^2 - \|J_k s_k + C_k\|^2 \geq \kappa_2 \|C_k\| \min\left\{\kappa_3\|C_k\|, \delta_k\right\}. \tag{3.27}$$

As a direct consequence of the way the penalty parameter is updated, we have the following result.

Lemma 3.5.1 The sequence $\{\rho_k\}$ satisfies

$$\begin{aligned} &\rho_k \geq \rho_{k-1} \geq 1 \quad \text{and} \\ &pred(s_k; \rho_k) \geq \frac{\rho_k}{2}\left(\|C_k\|^2 - \|J_k s_k + C_k\|^2\right). \end{aligned} \tag{3.28}$$

We now analyze the fraction of Cauchy and optimal decrease conditions for the tangential component. For the optimal decrease it is important to write down what necessary conditions the optimal solutions o_k^{d} and o_k^{c} of the trust–region subproblems (3.20) and (3.23) satisfy.

In the case of the decoupled approach these conditions are:

$$\bar{H}_k + \gamma_k I_{n-m} \quad \text{is positive semi–definite,} \tag{3.29}$$

$$\left(\bar{H}_k + \gamma_k I_{n-m}\right) o_k^{\mathsf{d}} = -\bar{g}_k, \text{ and} \tag{3.30}$$

$$\gamma_k \left(\delta_k - \|o_k^{\mathsf{d}}\|\right) = 0,$$

where $\gamma_k \geq 0$ is the Lagrange multiplier associated with the trust–region constraint $\|\bar{s}^{\mathsf{t}}\| \leq \delta_k$. (See Proposition 2.3.3.)

For the coupled approach such necessary conditions are as follows:

$$\bar{H}_k + \gamma_k W_k^T W_k \quad \text{is positive semi–definite,} \tag{3.31}$$

$$\left(\bar{H}_k + \gamma_k W_k^T W_k\right) o_k^{\mathsf{c}} = -\bar{g}_k, \text{ and} \tag{3.32}$$

$$\gamma_k\left(\delta_k - \|o_k^{\mathsf{c}}\|\right) = 0,$$

where $\gamma_k \geq 0$ is the Lagrange multiplier associated with the trust–region constraint $\|W_k \bar{s}^{\mathsf{t}}\| \leq \delta_k$. (This result also is derived from Proposition 2.3.3. In fact, the change of variables $\hat{s}^{\mathsf{t}} = (W_k^T W_k)^{\frac{1}{2}} \bar{s}^{\mathsf{t}}$, reduces the trust–region subproblem (3.23) to a trust–region subproblem of the form (2.2).)

Lemma 3.5.2 Let Assumptions 3.1–3.4 hold. If $\bar{s}_k^{\mathsf{t}}$ satisfies Condition 3.1, then

$$q_k(s_k^{\mathsf{q}}) - q_k(s_k) \geq \kappa_6 \|\bar{g}_k\| \min\left\{\kappa_7 \|\bar{g}_k\|, \kappa_8 \delta_k\right\} \tag{3.33}$$

and, moreover, if $\bar{s}_k^{\mathsf{t}}$ satisfies Condition 3.2, then

$$q_k(s_k^{\mathsf{q}}) - q_k(s_k) \geq \kappa_9 \gamma_k \delta_k^2, \tag{3.34}$$

where $\kappa_6, \ldots, \kappa_9$ are positive constants independent of the iterate k.

Proof The condition (3.33) is an application of Powell's Lemma 2.3.1. The condition (3.34) is a direct application of Lemma 2.3.2 for the necessary conditions given after Lemma 3.5.1. □

The following inequality is needed in many forthcoming lemmas.

Lemma 3.5.3 Under Assumptions 3.1–3.4 and Conditions 3.1–3.2,

$$q_k(0) - q_k(s_k^{\mathsf{q}}) - \Delta\lambda_k^T (J_k s_k + C_k) \geq -\kappa_{10} \|C_k\| \delta_k, \tag{3.35}$$

where κ_{10} is a positive constant independent of k.

Proof The proof follows the arguments in [35][Lemma 7.3]. The term $q_k(0)-q_k(s_k^{\mathsf{q}})$ can be bounded using (3.15), (3.19), and Assumption 3.4, in the following way:

$$\begin{aligned} q_k(0)-q_k(s_k^{\mathsf{q}}) &= -\nabla_x \ell_k^T s_k^{\mathsf{q}} - \frac{1}{2}(s_k^{\mathsf{q}})^T H_k (s_k^{\mathsf{q}}) \\ &\geq -\kappa_4 \|C_k\| \delta_k - \frac{1}{2}\|H_k\|\,\|s_k^{\mathsf{q}}\|^2 \\ &\geq -\kappa_4 \|C_k\| \delta_k - \frac{1}{2}\nu_7 \kappa_1 \|C_k\| \delta_k. \end{aligned} \tag{3.36}$$

On the other hand, it follows from (3.26) and $\|J_k s_k + C_k\| \leq \|C_k\|$ that

$$-\Delta\lambda_k^T (J_k s_k + C_k) \geq -\kappa_5 \|C_k\| \delta_k.$$

If we combine these two bounds we get (3.35) with $\kappa_{10} = \kappa_4 + \frac{1}{2}\nu_7\kappa_1 + \kappa_5$. □

The convergence theory is based on the following actual versus predicted estimates. These are minor modifications of the estimates given in [47].

Lemma 3.5.4 Let Assumptions 3.1–3.4 hold. There exists a positive constant κ_{11} independent of k, such that

$$\begin{aligned} |ared(s_k;\rho_k) - pred(s_k;\rho_k)| \leq \; & \kappa_{11}\Big(\|s_k\|^2 + \|\Delta\lambda_k\|\,\|s_k\|^2 \\ & + \rho_k\left(\|s_k\|^3 + \|C_k\|\,\|s_k\|^2\right)\Big). \end{aligned} \tag{3.37}$$

If H_k satisfies Condition 3.2, then

$$\begin{aligned} |ared(s_k;\rho_k) - pred(s_k;\rho_k)| \leq \; & \kappa_{12}\Big(\|\Delta\lambda_k\|\,\|s_k\|^2 \\ & + \rho_k\left(\|s_k\|^3 + \|C_k\|\,\|s_k\|^2\right)\Big), \end{aligned} \tag{3.38}$$

where κ_{12} is a positive constant independent of k.

Proof If we add and subtract $\ell(x_{k+1},\lambda_k)$ to $ared(s_k;\rho_k) - pred(s_k;\rho_k)$ and expand $\ell(\cdot,\lambda_k)$ around x_k we get

$$\begin{aligned} ared(s_k;\rho_k) - pred(s_k;\rho_k) = \; & \tfrac{1}{2} s_k^T \Big(H_k - \nabla_{xx}^2 \ell(x_k + t_k^1 s_k, \lambda_k)\Big) s_k \\ & + \Delta\lambda_k^T(-C_{k+1} + C_k + J_k s_k) \\ & - \rho_k\Big(\|C_{k+1}\|^2 - \|J_k s_k + C_k\|^2\Big) \end{aligned}$$

for some $t_k^1 \in (0,1)$. Again using the Taylor expansion we can write

$$\begin{aligned} ared(s_k;\rho_k) - pred(s_k;\rho_k) &= \tfrac{1}{2} s_k^T \left(H_k - \nabla_{xx}^2 \ell(x_k + t_k^1 s_k, \lambda_k) \right) s_k \\ & \quad -\tfrac{1}{2} \textstyle\sum_{i=1}^m (\Delta\lambda_k)_i s_k^T \nabla^2 c_i(x_k + t_k^2 s_k) s_k \\ & \quad -\rho_k \left(\textstyle\sum_{i=1}^m c_i(x_k + t_k^3 s_k)(s_k)^T \nabla^2 c_i(x_k + t_k^3 s_k)(s_k) \right. \\ & \quad +(s_k)^T J(x_k + t_k^3 s_k)^T J(x_k + t_k^3 s_k)(s_k) \\ & \quad \left. -(s_k)^T J(x_k)^T J(x_k)(s_k) \right), \end{aligned}$$

where $t_k^2,\ t_k^3 \in (0,1)$. Now we expand $c_i(x_k + t_k^3 s_k)$ around $c_i(x_k)$. This and Assumptions 3.1–3.4 give us the estimate (3.37) for some positive constant κ_{11}.

Inequality (3.38) is derived as inequality (3.37), using the Lipschitz continuity of the second derivatives and the fact that $\rho_k \geq 1$. $\square$

We terminate this section with the following lemma.

Lemma 3.5.5 Let Assumptions 3.1–3.4 hold. Every step s_k satisfies

$$\|s_k\| \leq \kappa_{13} \delta_k. \tag{3.39}$$

If s_k is rejected in Step 2.6 of Algorithm 3.4.1, then

$$\delta_{k+1} \geq \kappa_{14} \|s_k\|. \tag{3.40}$$

The constants κ_{13} and κ_{14} are positive and do not depend on k.

Proof In the coupled trust–region approach we have $\|s_k\| \leq 2\delta_k$ and $\delta_{k+1} \geq \frac{\alpha_1}{2}\|s_k\|$. See Step 2.5 of Algorithm 3.4.1. In the decoupled approach, $\|s_k\| = \|s_k^{\mathsf{q}} + W_k \bar{s}_k^{\mathsf{t}}\| \leq (1+\nu_6)\delta_k$ and similarly $\delta_{k+1} \geq \frac{\alpha_1}{2} \min\left\{1, \frac{1}{\nu_6}\right\} \|s_k\|$, where ν_6 is a uniform bound for $\|W_k\|$, see Assumption 3.4. We can combine these bounds to obtain

$$\begin{aligned} \|s_k\| &\leq \max\{2, 1+\nu_6\}\, \delta_k, \\ \delta_{k+1} &\geq \tfrac{\alpha_1}{2} \min\left\{1, \tfrac{1}{\nu_6}\right\} \|s_k\|. \end{aligned}$$

In the case where fraction of optimal decrease conditions (3.22) or (3.25) are imposed on $\bar{s}_k^{\mathsf{t}}$, the constants κ_{13} and κ_{14} depend also on β_3^{d} and β_3^{c}. $\square$

3.6 Global Convergence Results

The global convergence of the Trust–Region Reduced SQP Algorithm 3.4.1 to a stationary point is given in the following theorem.

Theorem 3.6.1 *(Dennis, El–Alem, and Maciel [35])* If Assumptions 3.1–3.4 hold and the components of the step satisfy Condition 3.1, then

$$\liminf_{k \to +\infty} \left(\|W_k^T \nabla f_k\| + \|C_k\| \right) = 0. \tag{3.41}$$

In this section we assume that the components of the step, the quadratic model, and the multiplier estimates are computed to satisfy Conditions 3.1–3.2, and we prove the following result from which we can establish the existence of a limit point of the sequence of iterates that satisfies the second–order necessary optimality conditions.

Theorem 3.6.2 If Assumptions 3.1–3.4 hold and the components of the step, the quadratic model, and the multiplier estimates satisfy Conditions 3.1–3.2, then

$$\liminf_{k \to +\infty} \left(\|W_k^T \nabla f_k\| + \|C_k\| + \gamma_k \right) = 0. \tag{3.42}$$

We defer the proof of this theorem to establish its major consequence.

Theorem 3.6.3 Let Assumptions 3.1–3.4 and Conditions 3.1–3.2 hold. Assume that $W(x)$ and $\lambda(x)$ are continuous functions and $\lambda_k = \lambda(x_k)$ for all k.

If $\{x_k\}$ is a bounded sequence generated by Algorithm 3.4.1, then there exists a limit point x_* such that

- $C(x_*) = 0$,
- $W(x_*)^T \nabla f(x_*) = 0$, and
- $W(x_*)^T \nabla^2_{xx} \ell(x_*, \lambda(x_*)) W(x_*)$ is positive semi–definite.

Moreover, if $\lambda(x_*)$ is such that $\nabla_x \ell(x_*, \lambda(x_*)) = 0$ then x_* satisfies the second–order necessary optimality conditions.

Proof Let $\{k_i\}$ be the index subsequence considered in (3.42). Since $\{x_{k_i}\}$ is bounded, it has a subsequence $\{x_{k_j}\}$ that converges to a point x_* and for which

$$\lim_{j \to +\infty} \left(\|W_{k_j}^T \nabla f_{k_j}\| + \|C_{k_j}\| + \gamma_{k_j} \right) = 0. \tag{3.43}$$

Now from this and the continuity of $C(x)$, we get $C(x_*) = 0$. Then we use the continuity of $W(x)$ and $\nabla f(x)$ to obtain

$$W(x_*)^T \nabla f(x_*) = 0.$$

Since $\lambda_1(\cdot)$ is a continuous function, we can use (3.29), (or (3.31) for the coupled approach), $\lim_{j \to +\infty} \gamma_{k_j} = 0$, the continuity of $W(x)$, $\lambda(x)$, and of the second derivatives of $f(x)$ and $c_i(x)$, $i = 1, \ldots, m$, to obtain

$$\lambda_1 \left(W(x_*)^T \nabla_{xx}^2 \ell(x_*, \lambda(x_*)) W(x_*) \right) \geq 0.$$

This shows that $W(x_*)^T \nabla_{xx}^2 \ell(x_*, \lambda(x_*)) W(x_*)$ is positive semi–definite. $\square$

The continuity of an orthogonal null–space basis $Z(x)$ for $\mathcal{N}(J(x))$ has been discussed in [16], [26], [58]. A straightforward implementation of the QR factorization of $J(x)^T$ might produce a discontinuous null–space orthogonal basis $Z(x)$. However, Coleman and Sorensen [26] showed how to modify the QR factorization in such a way that $Z(x)$ inherits the smoothness of $J(x)^T$. A class of nonorthogonal continuous null–space basis $W(x)$ is described in Chapter 4.

The equation $\nabla_x \ell(x_*, \lambda(x_*)) = 0$ is satisfied for consistent updates of the Lagrange multipliers like the least–squares update (3.9) or the adjoint update (4.14).

Now we prove Theorem 3.6.2. The proof of (3.42), although simpler, has the same structure as the proof of (3.41) given in [35].

Proof of Theorem 3.6.2

We prove by contradiction that

$$\liminf_{k \to +\infty} \left(\|\bar{g}_k\| + \|C_k\| + \gamma_k \right) = 0.$$

We show that the supposed existence of a $\epsilon_{tol} > 0$ such that

$$\|\bar{g}_k\| + \|C_k\| + \gamma_k > \epsilon_{tol}, \tag{3.44}$$

for all k, leads to a contradiction.

The proof requires the lower bounds for the predicted decrease given by the following three lemmas.

Lemma 3.6.1 Under Assumptions 3.1–3.4 and Conditions 3.1–3.2, the predicted decrease in the merit function satisfies

$$\begin{aligned} pred(s_k;\rho) \;\geq\; & \kappa_6\|\bar{g}_k\| \min\Big\{\kappa_7\|\bar{g}_k\|, \kappa_8\delta_k\Big\} - \kappa_{10}\|C_k\|\delta_k \\ & +\rho\Big(\|C_k\|^2 - \|J_k s_k + C_k\|\Big)^2, \end{aligned} \tag{3.45}$$

and also

$$pred(s_k;\rho) \;\geq\; \kappa_9\gamma_k\delta_k^2 - \kappa_{10}\|C_k\|\delta_k + \rho\Big(\|C_k\|^2 - \|J_k s_k + C_k\|\Big)^2, \tag{3.46}$$

for any $\rho > 0$.

Proof The two conditions (3.45) and (3.46) follow from a direct application of (3.35) and from the two different lower bounds (3.33) and (3.34) on $q_k(s_k^{\mathsf{q}}) - q_k(s_k)$.
$\square$

Lemma 3.6.2 Let Assumptions 3.1–3.4 and Conditions 3.1–3.2 hold and assume that $\|\bar{g}_k\| + \|C_k\| + \gamma_k > \epsilon_{tol}$. If $\|C_k\| \leq \theta\delta_k$, where θ is a constant satisfying

$$\theta \;\leq\; \min\left\{\frac{\epsilon_{tol}}{3\delta_{max}}, \frac{\kappa_6\epsilon_{tol}}{6\kappa_{10}\delta_{max}} \min\left\{\frac{\kappa_7\epsilon_{tol}}{3\delta_{max}}, \kappa_8\right\}, \frac{\kappa_9\epsilon_{tol}}{6\kappa_{10}}\right\}, \tag{3.47}$$

then the predicted decrease in the merit function satisfies either

$$\begin{aligned} pred(s_k;\rho) \;\geq\; & \tfrac{\kappa_6}{2}\|\bar{g}_k\| \min\Big\{\kappa_7\|\bar{g}_k\|, \kappa_8\delta_k\Big\} \\ & +\rho\Big(\|C_k\|^2 - \|J_k s_k + C_k\|^2\Big) \end{aligned} \tag{3.48}$$

or

$$pred(s_k;\rho) \;\geq\; \tfrac{\kappa_9}{2}\gamma_k\delta_k^2 + \rho\Big(\|C_k\|^2 - \|J_k s_k + C_k\|^2\Big), \tag{3.49}$$

for any $\rho > 0$.

Proof From $\|\bar{g}_k\| + \|C_k\| + \gamma_k > \epsilon_{tol}$ and the first bound on θ given by (3.47), we get

$$\|\bar{g}_k\| + \gamma_k > \frac{2}{3}\epsilon_{tol}.$$

Thus either $\|\bar{g}_k\| > \frac{1}{3}\epsilon_{tol}$ or $\gamma_k > \frac{1}{3}\epsilon_{tol}$. Let us first assume that $\|\bar{g}_k\| > \frac{1}{3}\epsilon_{tol}$. Using this, (3.45), $\delta_k \le \delta_{max}$, and the second bound on θ given by (3.47), we obtain

$$\begin{aligned} pred(s_k;\rho) &\ge \tfrac{\kappa_6}{2}\|\bar{g}_k\| \min\Big\{\kappa_7\|\bar{g}_k\|, \kappa_8\delta_k\Big\} \\ &\quad + \tfrac{\kappa_6\epsilon_{tol}}{6} \min\Big\{\tfrac{\kappa_7\epsilon_{tol}}{3}, \kappa_8\delta_k\Big\} - \kappa_{10}\delta_{max}\|C_k\| \\ &\quad + \rho\Big(\|C_k\|^2 - \|J_k s_k + C_k\|^2\Big) \\ &\ge \tfrac{\kappa_6}{2}\|\bar{g}_k\| \min\Big\{\kappa_7\|\bar{g}_k\|, \kappa_8\delta_k\Big\} \\ &\quad + \rho\Big(\|C_k\|^2 - \|J_k s_k + C_k\|^2\Big). \end{aligned}$$

Now suppose that $\gamma_k > \frac{1}{3}\epsilon_{tol}$. To establish (3.49), we combine (3.46) and the last bound on θ given by (3.47) and get

$$\begin{aligned} pred(s_k;\rho) &\ge \tfrac{\kappa_9}{2}\gamma_k\delta_k^2 + \Big(\tfrac{\kappa_9}{6}\epsilon_{tol}\delta_k - \kappa_{10}\|C_k\|\Big)\,\delta_k \\ &\quad + \rho\Big(\|C_k\|^2 - \|J_k s_k + C_k\|^2\Big) \\ &\ge \tfrac{\kappa_9}{2}\gamma_k\delta_k^2 + \rho\Big(\|C_k\|^2 - \|J_k s_k + C_k\|^2\Big). \end{aligned}$$

□

We can set ρ to ρ_{k-1} in Lemma 3.6.2 and conclude that, if $\|\bar{g}_k\| + \|C_k\| + \gamma_k > \epsilon_{tol}$ and $\|C_k\| \le \theta\delta_k$, then the penalty parameter at the current iterate does not need to be increased. See Step 2.4 of Algorithm 3.4.1.

The proof of the next lemma follows the argument given in the proof of Lemma 3.6.2 to show that either $\|\bar{g}_k\| > \frac{1}{3}\epsilon_{tol}$ or $\gamma_k > \frac{1}{3}\epsilon_{tol}$ holds.

Lemma 3.6.3 Let Assumptions 3.1–3.4 and Conditions 3.1–3.2 hold and assume that $\|\bar{g}_k\| + \|C_k\| + \gamma_k > \epsilon_{tol}$. If $\|C_k\| \le \theta\delta_k$, where θ satisfies (3.47), then there exists a constant $\kappa_{15} > 0$ such that

$$pred(s_k;\rho_k) \ge \kappa_{15}\delta_k^2. \qquad (3.50)$$

Proof By Lemma 3.6.2 we know that either (3.48) or (3.49) holds. Now we set $\rho = \rho_k$. In the first case we use $\|\bar{g}_k\| > \frac{1}{3}\epsilon_{tol}$ and get

$$\begin{aligned} pred(s_k;\rho_k) &\ge \tfrac{\kappa_6\epsilon_{tol}}{6} \min\{\tfrac{\kappa_7\epsilon_{tol}}{3}, \kappa_8\delta_k\} \\ &\ge \tfrac{\kappa_6\epsilon_{tol}}{6} \min\{\tfrac{\kappa_7\epsilon_{tol}}{3\delta_{max}}, \kappa_8\}\delta_k \\ &\ge \tfrac{\kappa_6\epsilon_{tol}}{6\delta_{max}} \min\{\tfrac{\kappa_7\epsilon_{tol}}{3\delta_{max}}, \kappa_8\}\delta_k^2. \end{aligned}$$

In the second case we use $\gamma_k > \frac{1}{3}\epsilon_{tol}$, to obtain

$$pred(s_k;\rho_k) \geq \frac{\kappa_9\epsilon_{tol}}{6}\delta_k^2.$$

Hence (3.50) holds with

$$\kappa_{15} = \min\left\{\frac{\kappa_6\epsilon_{tol}}{6\delta_{max}}\min\left\{\frac{\kappa_7\epsilon_{tol}}{3\delta_{max}},\kappa_8\right\},\frac{\kappa_9\epsilon_{tol}}{6}\right\}.$$

□

Next we prove that under the supposition $\|\bar{g}_k\| + \|C_k\| + \gamma_k > \epsilon_{tol}$, the penalty parameter ρ_k is uniformly bounded.

Lemma 3.6.4 Let Assumptions 3.1–3.4 and Conditions 3.1–3.2 hold and assume that $\|\bar{g}_k\| + \|C_k\| + \gamma_k > \epsilon_{tol}$ for all k. Then

$$\rho_k \leq \rho_*,$$

where ρ_* does not depend on k, and thus $\{\rho_k\}$ and $\{L_k\}$ are bounded sequences.

Proof If ρ_k is increased at iteration k, then it is updated according to the rule

$$\rho_k = 2\left(\frac{q_k(s_k) - q_k(0) + \Delta\lambda_k^T(J_ks_k + C_k)}{\|C_k\|^2 - \|J_ks_k + C_k\|^2}\right) + \bar{\rho}.$$

We can write

$$\begin{aligned}\tfrac{\rho_k}{2}\Big(\|C_k\|^2 - \|J_ks_k + C_k\|^2\Big) &= \nabla_x\ell_k^T s_k^{q} + \tfrac{1}{2}(s_k^{q})^T H_k(s_k^{q}) \\ &\quad - \Big(q_k(s_k^{q}) - q_k(s_k)\Big) + \Delta\lambda_k^T(J_ks_k + C_k) \\ &\quad + \tfrac{\bar{\rho}}{2}\Big(\|C_k\|^2 - \|J_ks_k + C_k\|^2\Big).\end{aligned}$$

By applying (3.27) to the left hand side and (3.33) and (3.35) to the right hand side, we obtain

$$\begin{aligned}\tfrac{\rho_k}{2}\kappa_2\|C_k\|\min\{\kappa_3\|C_k\|,\delta_k\} &\leq \kappa_{10}\delta_k\|C_k\| + \tfrac{\bar{\rho}}{2}\Big(-2(J_k^TC_k)^Ts_k - \|J_ks_k\|^2\Big) \\ &\leq (\kappa_{10} + \bar{\rho}\kappa_{13}\nu_4)\delta_k\|C_k\|.\end{aligned}$$

If ρ_k is increased at iteration k, then from Lemma 3.6.2 we certainly know that $\|C_k\| > \theta\delta_k$. Now we use this fact to establish that

$$\left(\frac{\kappa_2}{2}\min\{\kappa_3\theta,1\}\right)\,\rho_k \leq \kappa_{10} + \bar{\rho}\kappa_{13}\nu_4.$$

We proved that $\{\rho_k\}$ is bounded. From this and Assumptions 3.1–3.4 we conclude that $\{L_k\}$ is also bounded. $\square$

We can prove also under the supposition (3.44) that the trust radius is bounded away from zero.

Lemma 3.6.5 Let Assumptions 3.1–3.4 and Conditions 3.1–3.2 hold. If $\|\bar{g}_k\| + \|C_k\| + \gamma_k > \epsilon_{tol}$ for all k, then

$$\delta_k \geq \delta_* > 0,$$

where δ_* does not depend on k.

Proof If s_{k-1} was an acceptable step, then $\delta_k \geq \delta_{min}$. If not then $\delta_k \geq \kappa_{14}\|s_{k-1}\|$, and we consider the cases $\|C_k\| \leq \theta\delta_k$ and $\|C_k\| > \theta\delta_k$, where θ satisfies (3.47). In both cases we use the fact

$$1 - \eta_1 \leq \left| \frac{ared(s_{k-1}; \rho_{k-1})}{pred(s_{k-1}; \rho_{k-1})} - 1 \right|.$$

Case I. $\|C_{k-1}\| \leq \theta\delta_{k-1}$. From Lemma 3.6.3, inequality (3.50) holds for $k = k-1$. Thus we can use $\|s_{k-1}\| \leq \kappa_{13}\delta_{k-1}$, (3.26) and (3.38) with $k = k-1$, to obtain

$$\left| \frac{ared(s_{k-1}; \rho_{k-1})}{pred(s_{k-1}; \rho_{k-1})} - 1 \right| \leq \frac{\kappa_{12}(\kappa_5\kappa_{13}\delta_{k-1}^2 + \rho_*\kappa_{13}^2\delta_{k-1}^2 + \rho_*\theta\kappa_{13}\delta_{k-1}^2)\|s_{k-1}\|}{\kappa_{15}\delta_{k-1}^2}.$$

Thus $\delta_k \geq \kappa_{14}\|s_{k-1}\| \geq \frac{(1-\eta_1)\kappa_{14}\kappa_{15}}{\kappa_{12}(\kappa_5\kappa_{13}+\rho_*\kappa_{13}^2+\rho_*\theta\kappa_{13})} \equiv \kappa_{16}$.

Case II. $\|C_{k-1}\| > \theta\delta_{k-1}$. In this case from (3.27) and (3.28) with $k = k-1$ we get

$$\begin{aligned} pred(s_{k-1}; \rho_{k-1}) &\geq \tfrac{\rho_{k-1}}{2}\kappa_2\|C_{k-1}\| \min\{\kappa_3\|C_{k-1}\|, \delta_{k-1}\} \\ &\geq \rho_{k-1}\kappa_{17}\delta_{k-1}\|C_{k-1}\| \\ &\geq \rho_{k-1}\theta\kappa_{17}\delta_{k-1}^2, \end{aligned}$$

where $\kappa_{17} = \frac{\kappa_2}{2}\min\{\kappa_3\theta, 1\}$. Again we use $\rho_{k-1} \geq 1$, (3.26) and (3.38) with $k = k-1$, and this time the last two lower bounds on $pred(s_{k-1}; \rho_{k-1})$, and write

$$\begin{aligned} \left| \tfrac{ared(s_{k-1};\rho_{k-1})}{pred(s_{k-1};\rho_{k-1})} - 1 \right| &\leq \kappa_{12}\left(\tfrac{\rho_{k-1}(\kappa_5\kappa_{13}+\kappa_{13}^2)\delta_{k-1}^2\|s_{k-1}\|}{\rho_{k-1}\theta\kappa_{17}\delta_{k-1}^2} + \tfrac{\rho_{k-1}\kappa_{13}\delta_{k-1}\|C_{k-1}\|\|s_{k-1}\|}{\rho_{k-1}\kappa_{17}\delta_{k-1}\|C_{k-1}\|} \right) \\ &\leq \kappa_{12}\left(\tfrac{\kappa_5\kappa_{13}+\kappa_{13}^2+\theta\kappa_{13}}{\theta\kappa_{17}} \right)\|s_{k-1}\|. \end{aligned}$$

Hence $\delta_k \geq \kappa_{14}\|s_{k-1}\| \geq \frac{(1-\eta_1)\theta\kappa_{14}\kappa_{17}}{\kappa_{12}(\kappa_5\kappa_{13}+\kappa_{13}^2+\theta\kappa_{13})} \equiv \kappa_{18}$.

The result follows by setting $\delta_* = \min\{\delta_{min}, \kappa_{16}, \kappa_{18}\}$. $\square$

The next result is needed also for the proof of Theorem 3.6.2.

Lemma 3.6.6 Let Assumptions 3.1–3.4 and Conditions 3.1–3.2 hold. If $\|\bar{g}_k\| + \|C_k\| + \gamma_k > \epsilon_{tol}$ for all k, then an acceptable step always is found in finitely many trial steps.

Proof Let us prove the assertion by contradiction. Assume that for a given $\bar{k}$, $x_k = x_{\bar{k}}$ for all $k \geq \bar{k}$. This means that $\lim_{k\to+\infty} \delta_k = 0$ and all steps are rejected after iteration $\bar{k}$. See Steps 2.5 and 2.6 of Algorithm 3.4.1. We can consider the cases $\|C_k\| \leq \theta\delta_k$ and $\|C_k\| > \theta\delta_k$, where θ satisfies (3.47), and appeal to arguments similar to those used in Lemma 3.6.5 to conclude that in any case

$$\left|\frac{ared(s_k;\rho_k)}{pred(s_k;\rho_k)} - 1\right| \leq \kappa_{19}\delta_k, \quad k \geq \bar{k}\,,$$

where κ_{19} is a positive constant independent of the iterates. Since we are assuming that $\lim_{k\to+\infty} \delta_k = 0$, we have $\lim_{k\to+\infty} \frac{ared(s_k;\rho_k)}{pred(s_k;\rho_k)} = 1$ and this contradicts the rules that update the trust radius in Step 2.5 of Algorithm 3.4.1. $\square$

Now we can finally prove Theorem 3.6.2.

Theorem 3.6.2 If Assumptions 3.1–3.4 hold and the components of the step, the quadratic model, and the multiplier estimates satisfy Conditions 3.1–3.2, then

$$\liminf_{k\to+\infty} \left(\|W_k^T \nabla f_k\| + \|C_k\| + \gamma_k\right) = 0. \tag{3.51}$$

Proof Suppose that there exists an $\epsilon_{tol} > 0$ such that $\|\bar{g}_k\| + \|C_k\| + \gamma_k > \epsilon_{tol}$ for all k.

At each iteration k either $\|C_k\| \leq \theta\delta_k$ or $\|C_k\| > \theta\delta_k$, where θ satisfies (3.47). In the first case we appeal to Lemmas 3.6.3 and 3.6.5 and obtain

$$pred(s_k;\rho_k) \geq \kappa_{15}\delta_*^2.$$

If $\|C_k\| > \theta\delta_k$, we have from $\rho_k \geq 1$, (3.27), (3.28), and Lemma 3.6.5, that

$$pred(s_k;\rho_k) \geq \frac{\kappa_2}{2}\theta \min\{\kappa_3\theta, 1\}\delta_*^2.$$

Hence there exists a positive constant κ_{20} not depending on k such that $pred(s_k;\rho_k) \geq \kappa_{20}$. From Lemma 3.6.6, we can ignore the rejected steps and work only with successful iterates. So, without loss of generality, we have

$$L_k - L_{k+1} = ared(s_k;\rho_k) \geq \eta_1 pred(s_k;\rho_k) \geq \eta_1\kappa_{20}.$$

Now, if we let k go to infinity, this contradicts the boundedness of $\{L_k\}$. Thus we proved that there exists an index subsequence say $\{k_i\}$ such that

$$\lim_{i\to+\infty}\left(\|\bar{g}_{k_i}\|+\|C_{k_i}\|+\gamma_{k_i}\right)=0.$$

The proof is completed by showing that the limit above implies the limit (3.51). From $\lim_{i\to+\infty}\|C_{k_i}\|=0$ and $\|s^{\mathsf{q}}_{k_i}\|\le\kappa_1\|C_{k_i}\|$ for all i, we obtain $\lim_{i\to+\infty}\|s^{\mathsf{q}}_{k_i}\|=0$. But $\bar{g}_{k_i}=W_{k_i}^T\left(H_{k_i}s^{\mathsf{q}}_{k_i}+\nabla f_{k_i}\right)$ and $\{H_k\}$ and $\{W_k\}$ are bounded sequences; so we finally get (3.51). $\square$

The local convergence of these trust–region reduced SQP algorithms is studied in [42] under tighter conditions on the multiplier estimates and the quasi–normal components.

3.7 The Use of the Normal Decomposition with the Least–Squares Multipliers

The normal component has been defined in (3.8). We now redefine s^{n}_k as

$$s^{\mathsf{n}}_k=\begin{cases}-J_k^T(J_kJ_k^T)^{-1}C_k & \text{if } \|J_k^T(J_kJ_k^T)^{-1}C_k\|\le\delta_k,\\ -\xi_kJ_k^T(J_kJ_k^T)^{-1}C_k & \text{otherwise,}\end{cases}\tag{3.52}$$

where $\xi_k=\frac{\delta_k}{\|J_k^T(J_kJ_k^T)^{-1}C_k\|}$. This redefinition forces the normal component to stay inside the trust region (see condition (3.18)).

The results in the rest of this chapter use Assumption 3.5. This assumption implies the existence of a constant $\nu_9>0$ satisfying $\|(J(x)J(x)^T)^{-1}\|\le\nu_9$ for all x in Ω and $\|(J_kJ_k^T)^{-1}\|\le\nu_9$ for all nonnegative integers k.

We prove in Lemma 3.7.1 that the normal component (3.52) always gives a fraction of optimal decrease for the trust–region subproblem for the linearized constraints (3.14). This condition is as follows:

$$\begin{aligned}&\|C_k\|^2-\|J_ks^{\mathsf{q}}_k+C_k\|^2\ge\beta_2^{\mathsf{q}}\left(\|C_k\|^2-\|J_ko^{\ell}_k+C_k\|^2\right),\\&\|s^{\mathsf{q}}_k\|\le\beta_3^{\mathsf{q}}\delta_k,\end{aligned}\tag{3.53}$$

where β_2^{q}, β_3^{q} are positive constants independent of k, and o^{ℓ}_k is the optimal solution of the trust–region subproblem for the linearized constraints (3.14). As a result the normal component (3.52) satisfies the fraction of Cauchy decrease (3.16) for the trust–region subproblem for the linearized constraints (3.14). Since $\{(J_kJ_k^T)^{-1}\}$ is a bounded sequence this implies the decrease condition (3.18) used in our convergence theory.

Lemma 3.7.1 Let Assumptions 3.1–3.5 hold. The normal component (3.52) satisfies a fraction of optimal decrease (3.53) for the trust–region subproblem for the linearized constraints.

Proof From the definition in (3.52) it is obvious that $\|s_k^{\mathsf{n}}\| \le \beta_3^{\mathsf{q}}\delta_k$ holds with $\beta_3^{\mathsf{q}} = 1$.

If $\|J_k^T(J_kJ_k^T)^{-1}C_k\| \le \delta_k$, then s_k^{n} solves (3.14), and the result holds for any value of β_2^{q} in $(0,1]$. If this is not the case, then

$$\|C_k\|^2 - \|J_k s_k^{\mathsf{n}} + C_k\|^2 = \xi_k(2-\xi_k)\|C_k\|^2 \ge \xi_k\|C_k\|^2 \ge \frac{\delta_k}{\nu_4\nu_9}\|C_k\|, \tag{3.54}$$

since $\|J_k^T(J_kJ_k^T)^{-1}C_k\| \le \nu_4\nu_9\|C_k\|$ and $\xi_k \le 1$.

We also have

$$\begin{aligned}
\|C_k\|^2 - \|J_k o_k^\ell + C_k\|^2 &= -2(J_k^TC_k)^T o_k^\ell - (o_k^\ell)^T(J_k^TJ_k)(o_k^\ell) \\
&\le 2\nu_4\|C_k\|\ \|o_k^\ell\| + \nu_4^2\|o_k^\ell\|^2 \\
&\le 2\nu_4\delta_k\|C_k\| + \nu_4^2\delta_k\|o_k^\ell\| \\
&\le (2\nu_4 + \nu_4^3\nu_9)\delta_k\|C_k\|,
\end{aligned}$$

since $\|J_k^T(J_kJ_k^T)^{-1}\|\|C_k\| > \delta_k \ge \|o_k^\ell\|$. Combining this last inequality with (3.54) we get

$$\|C_k\|^2 - \|J_k s_k^{\mathsf{n}} + C_k\|^2 \ge \frac{1}{\nu_4^2\nu_9(2+\nu_4^2\nu_9)}\left(\|C_k\|^2 - \|J_k o_k^\ell + C_k\|^2\right),$$

and this completes the proof. $\square$

In the next lemma we show that the normal component (3.52) and the least–squares multipliers (3.9) satisfy the requirements in Condition 3.2 needed to prove global convergence to a point satisfying the second–order necessary optimality conditions.

Lemma 3.7.2 Let Assumptions 3.1–3.5 hold. The normal component (3.52) and the least–squares multipliers (3.9) satisfy conditions (3.19) and (3.26).

Proof It can be easily confirmed that $\nabla_x \ell_k^T s_k^{\mathsf{n}} = 0$. Thus, $\nabla_x \ell_k^T s_k^{\mathsf{n}} \leq \kappa_4 \|C_k\| \delta_k$. The condition (3.26) holds since from Assumptions 3.1–3.3 and 3.5, the function $\lambda(x) = -(J(x)J(x)^T)^{-1}J(x)\nabla f(x)$ has bounded derivatives in Ω and hence is Lipschitz continuous in the domain Ω. $\square$

3.8 Analysis of the Trust–Region Subproblem for the Linearized Constraints

In this section we investigate the trust–region subproblem for the linearized constraints (3.14). We saw in Section 3.7 that the normal component gives a fraction of optimal decrease for the trust–region subproblem for the linearized constraints. To compute a step s_k^{q} that satisfies this property, we can also use the techniques proposed in [106], [126], [133] and described in Section 2.3.1.

In the next theorem we show that the trust–region subproblem (3.14), due to its particular structure, tends to fall in the hard case in the latest stages of Algorithm 3.4.1. This result is relevant in our opinion since the algorithms proposed in [106], [126], [133] for the solution of trust–region subproblems deal with the hard case.

The trust–region subproblem (3.14) can be rewritten as

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}C_k^T C_k + (J_k^T C_k)^T s^{\mathsf{q}} + \frac{1}{2}(s^{\mathsf{q}})^T (J_k^T J_k)(s^{\mathsf{q}}) \\ \text{subject to} & \|s^{\mathsf{q}}\| \leq \delta_k. \end{array} \tag{3.55}$$

The matrix $J_k^T J_k$ is always positive semi–definite and, under Assumption 3.2, has rank m. Let $E_k(0)$ denote the eigenspace associated with the eigenvalue 0, i.e. $E_k(0) = \{v_k \in \mathbb{R}^n : \; J_k^T J_k v_k = 0\}$. The hard case, as we saw in Section 2.3.1, is defined by the two following conditions:

(a) $(v_k)^T(J_k^T C_k) = 0$ for all v_k in $E_k(0)$ and

(b) $\|(J_k^T J_k + \gamma I_n)^{-1} J_k^T C_k\| < \delta_k$ for all $\gamma > 0$.

Theorem 3.8.1 Under Assumptions 3.1–3.5, if $\lim_{k\to+\infty} \frac{\|C_k\|}{\delta_k} = 0$ then there exists a k_h such that, for all $k \geq k_h$, the trust–region subproblem (3.55) falls in the hard case as defined above by (a) and (b).

Proof First we show that (a) holds at every iteration of the algorithm. If $v_k \in E_k(0)$,

$$J_k^T J_k v_k = 0.$$

Multiplying both sides by $(J_k J_k^T)^{-1} J_k$ gives us

$$J_k v_k = 0.$$

Thus $(v_k)^T (J_k^T C_k) = 0$ for all v_k in $E_k(0)$.

Now we prove that there exists a k_h such that (b) holds for every $k \geq k_h$. Since $h_k(\gamma) = \|(J_k^T J_k + \gamma I_n)^{-1} J_k^T C_k\|$ is a monotone strictly decreasing function of γ for $\gamma > 0$,

$$\lim_{\gamma \to 0^+} h_k(\gamma) < \delta_k$$

is equivalent to $h_k(\gamma) < \delta_k$, for all $\gamma > 0$. But from the singular value decomposition of J_k^T [66][Page 71] we obtain

$$\lim_{\gamma \to 0^+} h_k(\gamma) = \| \lim_{\gamma \to 0^+} (J_k^T J_k + \gamma I_n)^{-1} J_k^T C_k\| = \|J_k^T (J_k J_k^T)^{-1} C_k\|.$$

Hence $h_k(\gamma) < \delta_k$ holds for all $\gamma > 0$ if and only if $\|J_k^T (J_k J_k^T)^{-1} C_k\| < \delta_k$.

Now since $\lim_{k \to +\infty} \frac{\|C_k\|}{\delta_k} = 0$, there exists a k_h such that $\|C_k\| < \frac{1}{\nu_4 \nu_9} \delta_k$ for all $k \geq k_h$. Thus $\|J_k^T (J_k J_k^T)^{-1} C_k\| \leq \nu_4 \nu_9 \|C_k\| < \delta_k$, for all $k \geq k_h$, and this completes the proof of the theorem. $\square$

We complete this section with the following corollary.

Corollary 3.8.1 Under Assumptions 3.1–3.5, if $\lim_{k \to +\infty} \|C_k\| = 0$ and the trust radius is uniformly bounded away from zero, then there exists a k_h such that, for all $k \geq k_h$, the trust–region subproblem (3.55) falls in the hard case as defined above by (a) and (b).

Proof If $\lim_{k \to +\infty} \|C_k\| = 0$ and the trust radius is uniformly bounded away from zero then $\lim_{k \to +\infty} \frac{\|C_k\|}{\delta_k} = 0$ and Theorem 3.8.1 can be applied. $\square$

Chapter 4

A Class of Nonlinear Programming Problems

In this chapter, we introduce and analyze the following important class of nonlinear programming problems:

$$\begin{array}{ll} \text{minimize} & f(y,u) \\ \text{subject to} & C(y,u) = 0, \\ & u \in \mathcal{B} = \{u: \ a \le u \le b\}, \end{array} \tag{4.1}$$

with

$$x \equiv \begin{pmatrix} y \\ u \end{pmatrix},$$

$y \in \mathbb{R}^m$, $u \in \mathbb{R}^{n-m}$, $a \in (\mathbb{R} \cup \{-\infty\})^{n-m}$, and $b \in (\mathbb{R} \cup \{+\infty\})^{n-m}$. The functions $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ and $C : \mathbb{R}^n \longrightarrow \mathbb{R}^m$, $m < n$, are assumed to be at least twice continuously differentiable. The Jacobian matrix of $C(x)$ is denoted by $J(x)$. The notation used for this class of problems is such that vectors $(s)_y$ and $(s)_u$ represent the subvectors of $s \in \mathbb{R}^n$ corresponding to the y and u components, respectively.

Minimization problems of the form (4.1) often arise from the discretization of optimal control problems. In this case y is the vector of state variables, u is the vector of control variables, and $C(y,u) = 0$ is the (discretized) state equation. The state equation can be nonlinear in the state variables y, in the control variables u, or in both. In Section 4.5, we provide two examples of optimal control problems for which the discretization is of the form (4.1): a boundary nonlinear parabolic control problem and a distributed nonlinear elliptic control problem. There are optimal control problems arising in fluid flow for which a discretization also is of the form (4.1) (see e.g. Cliff, Heinkenschloss, and Shenoy [22] and Heinkenschloss [72]). Other applications include optimal design and parameter identification problems.

In Chapters 5 and 6, we propose, analyze, and test a family of trust–region interior–point reduced SQP algorithms for the solution of problem (4.1).

One of the goals of this chapter is to present some properties of problem (4.1). In Section 4.1, we introduce the basic structure of the problem. The first and second order optimality conditions for (4.1) are stated in Section 4.4. We state them in a nonstandard form that leads in Chapter 5 to the diagonal matrix used in the affine scaling interior–point approach. In Section 4.2, we study the relationship between the all–at–once approach followed in this thesis and the black box approach based on a equivalent reduced formulation. These connections are known for problems with no bound constraints but they motivate the all–at–once approach based on (4.1) and reveal useful information for problems with bound constraints on u. In Section 4.3, we present properties of the projection associated with problem (4.1). Two nonlinear optimal control example problems for which the discretization is of the form (4.1) are described in Section 4.5. In the last section we comment on the important issue of the problem scaling inherent in optimal control problems.

4.1 Structure of the Minimization Problem

The Lagrange function $\ell : \mathbb{R}^{n+m} \longrightarrow \mathbb{R}^n$ associated with minimizing $f(x)$ subject to the equality constraint $C(x) = \big(c_1(x), \ldots, c_m(x)\big)^T = 0$ is given by

$$\ell(x, \lambda) = f(x) + \lambda^T C(x),$$

where $\lambda \in \mathbb{R}^m$ are the Lagrange multipliers.

The linearized constraints are given by $J(x)s = -C(x)$. If we take

$$s = \begin{pmatrix} s_y \\ s_u \end{pmatrix}, \quad s_y \in \mathbb{R}^m, \; s_u \in \mathbb{R}^{n-m},$$

and $J(x) = \begin{pmatrix} C_y(x) & C_u(x) \end{pmatrix}$, we can write this linearization as

$$\begin{pmatrix} C_y(x) & C_u(x) \end{pmatrix} \begin{pmatrix} s_y \\ s_u \end{pmatrix} = -C(x). \tag{4.2}$$

We say that s satisfies the (discretized) linearized state equation if it is a solution to (4.2). If $C_y(x)$ is invertible, the solutions of the linearized state equation are of the form

$$s = s^{\mathrm{q}} + W(x) s_u, \tag{4.3}$$

where

$$s^{\mathrm{q}} = \begin{pmatrix} -C_y(x)^{-1} C(x) \\ 0 \end{pmatrix} \tag{4.4}$$

is a particular solution and

$$W(x) = \begin{pmatrix} -C_y(x)^{-1}C_u(x) \\ I_{n-m} \end{pmatrix} \tag{4.5}$$

is a matrix whose columns form a basis for the null space $\mathcal{N}(J(x))$ of $J(x)$. This quasi–normal decomposition of s is of the type (3.5) defined in Section 3.2 since in general the columns of $W(x)$ are not orthogonal and s^{q} is not the minimum norm solution of the linearized constraints (see Figure 4.1). The role of the quasi–normal component s^{q} is to move towards feasibility. Furthermore, the y component of the quasi–normal component s^{q} is just the step that one would compute if one would apply Newton's method for the solution of the nonlinear equation $C(y,u) = 0$ for fixed u. The tangential component $W(x)s_u$ is in the null space of $J(x)$ and its role as we can see in Chapter 5 is to move towards optimality.

We assume that $C_y(x)$ is nonsingular. In many applications this is a reasonable assumption that can be shown for appropriate choices of the discretization parameters. However ill–conditioning can occur and we take this into account in the development of our algorithms in Chapters 5 and 6.

One can see that matrix–vector multiplications of the form $W(x)^T s$ and $W(x)s_u$ involve only the solution of linear systems with the matrices $C_y(x)$ and $C_y(x)^T$. In fact we have

$$W(x)s_u = \begin{pmatrix} -C_y(x)^{-1}C_u(x)s_u \\ s_u \end{pmatrix}$$

for which we need to solve the form of the (discretized) linearized state equation:

$$C_y(x)v_y = C_u(x)s_u.$$

Moreover,

$$W(x)^T s = -C_u(x)^T C_y(x)^{-T} s_y + s_u$$

and this requires the solution of the adjoint equation of the (discretized) linearized state equation given above, i.e. it requires the solution of:

$$C_y(x)^T v_y = s_y. \tag{4.6}$$

4.2 All–At–Once rather than Black Box

The point we want to convey in this section has nothing to do with the presence or absence of the bound constraints $a \leq u \leq b$. Therefore we consider the simpler case where there are no bounds, i.e. where $\mathcal{B} = \mathbb{R}^{n-m}$. The purpose of this section is to discuss some of the basic relationships between the problem

$$\begin{array}{ll} \text{minimize} & f(y,u) \\ \text{subject to} & C(y,u) = 0, \end{array} \tag{4.7}$$

and a reduced formulation of this problem that can be obtained by applying the implicit function theorem. In fact, suppose there exists an open set $\mathcal{U}$ such that for all $u \in \mathcal{U}$ there exists a solution y of $C(y,u) = 0$ and such that the matrix $C_y(x)$ is invertible for all $x = (y^T, u^T)^T$ with $u \in \mathcal{U}$ and $C(y,u) = 0$. Then the implicit function theorem guarantees the existence of a differentiable function

$$y(\cdot) : \mathcal{U} \to \mathbb{R}^m$$

defined by

$$C(y(u), u) = 0$$

and problem (4.7) is equivalent to

$$\text{minimize} \quad \hat{f}(u) \equiv f(y(u), u)\,. \tag{4.8}$$

This leads to the so–called *black box* approach in which the nonlinear constraint $C(y,u) = 0$ is not visible to the optimizer. Its solution is part of the evaluation of the objective function $\hat{f}(u)$. The reduced problem can be solved by a Newton–like method. For optimal control problems, many algorithms follow this approach and use projection techniques [70], [119] to handle the bounds on the variables u.

The reduced problem (4.8) is important since it leads us to the use of reduced SQP algorithms. The relation between problem (4.7) and the reduced problem (4.8) gives important insight into the structure of (4.7) and allows us to extend techniques successfully applied to problems of the form (4.8). To see why this is true we need to study the derivatives of the function $\hat{f}$.

Since $y(\cdot)$ is differentiable, the function $\hat{f}$ is differentiable and its gradient is given by

$$\nabla \hat{f}(u) = \nabla_u y \, \nabla_y f(y(u), u) + \nabla_u f(y(u), u), \tag{4.9}$$

where $\nabla_u y = \frac{dy}{du}^T$. The derivative of $y(u)$ with respect to u can be obtained from taking derivatives on both sides of $C(y(u), u) = 0$:

$$C_y(y(u), u)\frac{dy}{du} + C_u(y(u), u) = 0. \tag{4.10}$$

Thus, from (4.9) and (4.10) we see that

$$\nabla \hat{f}(u) = W(y(u), u)^T \nabla f(y(u), u). \tag{4.11}$$

Moreover, it can be shown that the Hessian of $\hat{f}$ is equal to the reduced Hessian

$$\nabla^2 \hat{f}(u) = W(y(u), u)^T \nabla^2_{xx} \ell(y(u), u, \lambda)\, W(y(u), u),$$

provided that the Lagrange multipliers are computed by (4.14) given below.

The so–called *all–at–once* approach treats both y and u as independent variables. All–at–once approaches were proposed to solve optimal control problems among many others in [75], [79], [82], [83], [85], [86], [87], [89]. For the optimal control problems that we consider in this thesis, the all–at–once approach is based on the formulation (4.7) (actually (4.1) if we include the bound constraints on the controls). The goal is to move towards optimality and feasibility at the same time, and this offers significant advantages. SQP algorithms are of particular interest since they allow use of the structure of optimal control problems, see e.g. [87], [88]. As we saw in Chapter 3 for equality–constrained optimization they do not require the (possibly very expensive) solution of the nonlinear state equation $C(y, u) = 0$ at every iteration.

If we solve (4.7) by the SQP Algorithm 3.2.1, then the quadratic programming subproblem we have to solve at every iteration is of the form

$$\begin{aligned} \text{minimize} \quad & \nabla f(x)^T s + \tfrac{1}{2} s^T \nabla^2_{xx} \ell(x, \lambda)\, s \\ \text{subject to} \quad & C_y(x) s_y + C_u(x) s_u + C(x) = 0. \end{aligned} \tag{4.12}$$

If the reduced Hessian $W(x)^T \nabla^2_{xx} \ell(x, \lambda)\, W(x)$ is nonsingular, the solution s of (4.12) is given by (4.3) and (4.4) with

$$s_u = -\Big(W(x)^T \nabla^2_{xx} \ell(x, \lambda)\, W(x) \Big)^{-1} W(x)^T \Big(\nabla^2_{xx} \ell(x, \lambda) s^{\mathsf{q}} + \nabla f(x) \Big). \tag{4.13}$$

Such s_u is also the solution of the quadratic programming subproblem of the Reduced SQP Algorithm 3.2.2.

In practice the Hessian $\nabla^2_{xx} \ell(x, \lambda)$ or the reduced Hessian $W(x)^T \nabla^2_{xx} \ell(x, \lambda)\, W(x)$ are often approximated using secant updates. In the latter case, when an approximation to $\nabla^2_{xx} \ell(x, \lambda)$ is not available, then the cross–term $W(x)^T \nabla^2_{xx} \ell(x, \lambda) s^{\mathsf{q}}$ has also to be approximated. This term can be approximated by zero, by finite differences, or by secant updates. In the case where this cross term is approximated by zero, the right hand side of the linear system (4.13) defining s_u can be written as

$$W(x)^T \nabla f(x) = -C_u(x)^T C_y(x)^{-T} \nabla_y f(x) + \nabla_u f(x).$$

Thus, if the Lagrange multipliers are computed by the adjoint formula

$$\lambda = -C_y(x)^{-T} \nabla_y f(x), \tag{4.14}$$

then

$$W(x)^T \nabla f(x) = C_u(x)^T \lambda + \nabla_u f(x) = \nabla_u \ell(x, \lambda). \tag{4.15}$$

One can see that the gradient and the Hessian information in the SQP algorithm for (4.7) and in the Newton method for (4.8) are the same if y and u solve $C(y,u) = 0$. Thus, if Newton–like methods are applied to the solution of (4.8), then one has all the ingredients available necessary to implement an SQP algorithm for the solution of (4.7). The important difference, of course, is that in the SQP algorithm we do not have to solve the nonlinear constraints $C(y,u) = 0$ at every iteration. Thus we combine the possible implementational advantages of a black–box approach with the generally more efficient all–at–once approach.

Specifically, our consequent use of the structure of the optimal control problems leads to our family of trust–region interior–point reduced SQP algorithms (see Chapter 5). These algorithms only require information that the user has to provide anyway if a black–box approach is used with a Newton–like method for the solution of the nonlinear state equation $C(y,u) = 0$ and adjoint equations of the form (4.6) for the computation of the reduced gradient (4.11). Furthermore, the inexact analysis for these algorithms presented in Chapter 6 provides practical rules to solve inexactly the linearized state and adjoint equations that guarantee global convergence.

In these considerations we neglected the bound constraints $a \leq u \leq b$. We have already pointed out that these relationships between (4.7) and (4.8) are basically the same with or without the bound constraints on the control variables. (See also Section 5.1.)

4.3 The Oblique Projection

In this section we show how the quasi–normal decomposition (4.3) differs from the normal decomposition (3.6) for problem (4.1). The normal decomposition applied to problem (4.1) has the form

$$s = s^{\mathsf{n}} + Z(x)\bar{s}^{\mathsf{t}},$$

where $Z(x)$ is a matrix whose columns form an orthogonal basis for $\mathcal{N}(J(x))$. We showed in Section 3.2 how to compute this decomposition from the QR factorization of $J(x)^T$.

A major difference between the decompositions (3.6) and (4.3) lies in the form of the basis of the null space $\mathcal{N}(J(x))$. It is reasonable in this class of problems to access to the basis $W(x)$ given in (4.5) since it exploits the structure of the problem and allows the use of linear solvers available from the application. The use of the QR factorization for this class of problems is problematic: it depends strongly on the sparsity pattern of $J(x)$, it might cause unnecessary fill–in, and it requires the user to do an involved computation of no value except in the optimization algorithm. Furthermore, the normal component s^{n} has a nonzero u component (see Figure 4.1) and this means that the bounds on the variables u would have to interfere somehow in the computation of s^{n}. These problems do not arise if the quasi–normal component (4.4) is used.

One other major difference is the type of projection associated with both decompositions. The quasi–normal decomposition (4.3) offers an oblique projector onto $\mathcal{N}(J(x))$:

$$P_{obl}(x) = W(x)W(x)^T, \tag{4.16}$$

where $W(x)$ is given by (4.5). The normal decomposition (3.6) when applied to the equality constraints of our problem (4.1) provides an orthogonal projector

$$P_{ort}(x) = Z(x)Z(x)^T. \tag{4.17}$$

It can be easily proved that

$$P_{ort}(x) = Z(x)Z(x)^T = W(x)\left(W(x)^T W(x)\right)^{-1} W(x)^T. \tag{4.18}$$

In Figure 4.1 we depict the action of the projectors $P_{obl}(x)$ and $P_{ort}(x)$ on a given vector v. The following proposition provides an explanation for the form of $P_{obl}(x)v$ given in Figure 4.1.

Proposition 4.3.1 Given a vector v in $\mathbb{R}^n$,

$$P_{ort}(x)v = P_{ort}(x)\begin{pmatrix} 0 \\ W(x)^T v \end{pmatrix}. \tag{4.19}$$

In addition, $\begin{pmatrix} 0 \\ W(x)^T v \end{pmatrix}$ is the unique vector in the vector space $\{x = (y^T, u^T)^T \in \mathbb{R}^n : \; y = 0\}$ for which (4.19) holds.

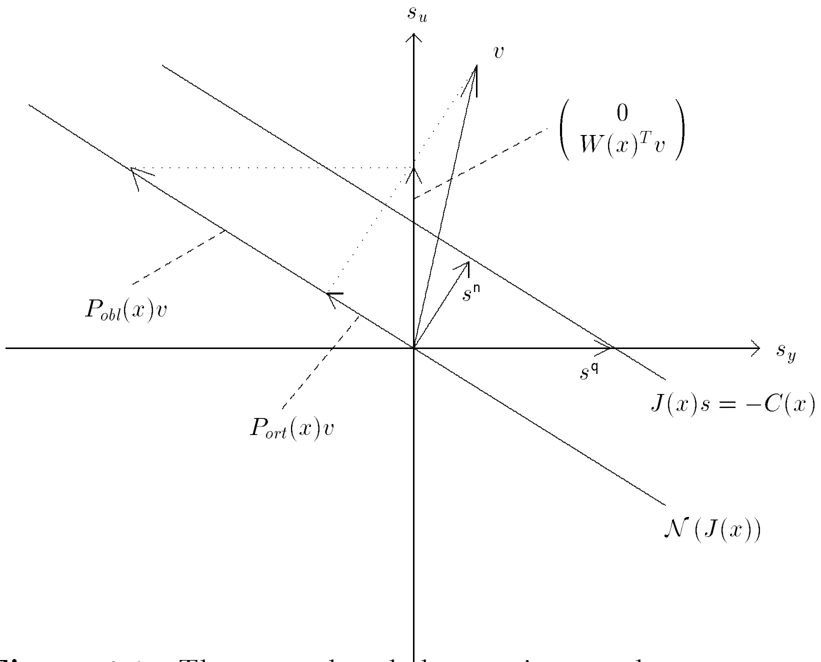

Figure 4.1 The normal and the quasi–normal components and the action of the orthogonal and oblique projectors.

Proof The proof of the first part is the following:

$$
\begin{aligned}
& P_{ort}(x) \begin{pmatrix} 0 \\ W(x)^T v \end{pmatrix} \\
&= W(x)\left(W(x)^T W(x)\right)^{-1} \begin{pmatrix} -C_y(x)^{-1}C_u(x) \\ I_{n-m} \end{pmatrix}^T \begin{pmatrix} 0 \\ W(x)^T v \end{pmatrix} \\
&= W(x)\left(W(x)^T W(x)\right)^{-1} W(x)^T v \\
&= P_{ort}(x)v,
\end{aligned}
$$

where we used (4.5) and the form of $P_{ort}(x)$ given in (4.18). To prove the uniqueness suppose that $x_1 = \begin{pmatrix} 0 \\ u_1 \end{pmatrix}$ and $x_2 = \begin{pmatrix} 0 \\ u_2 \end{pmatrix}$ satisfy $P_{ort}(x)x_1 = P_{ort}(x)x_2$. From $P_{ort}(x)(x_1 - x_2) = 0$ we conclude that $x_1 - x_2$ is orthogonal to $\mathcal{N}(J(x))$, i.e. $W(x)^T(x_1 - x_2) = 0$. But this is just

$$
\begin{pmatrix} -C_y(x)^{-1}C_u(x) \\ I_{n-m} \end{pmatrix}^T \begin{pmatrix} 0 \\ u_1 - u_2 \end{pmatrix} = 0
$$

and $u_1 = u_2$. □

From this proposition we know how to depict $W(x)^T v$ along the u axis. Note that

$$P_{obl}(x)v = W(x)W(x)^T v = \begin{pmatrix} -C_y(x)^{-1}C_u(x)\,W(x)^T v \\ W(x)^T v \end{pmatrix}$$

lies in the null space $\mathcal{N}(J(x))$.

4.4 Optimality Conditions

In this section we apply the first–order necessary and the second–order necessary and sufficient optimality conditions to problem (4.1). These conditions provide a powerful characterization of local minimizers in nonlinear programming and are used in many different fields of mathematics and science. They were discovered independently by Karush [80] in 1939 and by Kuhn and Tucker [84] in 1951. One can see these conditions as an extension of the Lagrange multiplier theory for problems with equality constraints (see Propositions 3.1.1 and 3.1.2) to problems with both equality and inequality constraints. By a general nonlinear programming problem we mean the problem

$$\begin{aligned} \text{minimize} \quad & f(x) \\ \text{subject to} \quad & h_i(x) = 0,\ i = 1, \ldots, p\,, \\ & g_i(x) \geq 0,\ i = 1, \ldots, l\,, \end{aligned} \tag{4.20}$$

where it is assumed that f, h_i, and g_i are twice continuously differentiable functions defined from $\mathbb{R}^n$ to $\mathbb{R}$. In order to describe the form of the optimality conditions that we use, we need to introduce the notion of regularity for both equalities and inequalities.

Definition 4.4.1 A point x_* is regular for problem (4.20) if the the set of vectors

$$\begin{aligned} &\Big\{\nabla h_i(x_*),\ i = 1, \ldots, p\Big\} \ \cup \\ &\Big\{\nabla g_i(x_*),\ \text{for all } i \in \{1, \ldots, l\} \text{ such that } g_i(x_*) = 0\Big\} \end{aligned} \tag{4.21}$$

is linearly independent.

The inequality constraints $g_i(x) \geq 0$, for all $i \in \{1, \ldots, l\}$ such that $g_i(x_*) = 0$, are said to be active or binding at x_*.

The optimality conditions for nonlinear programming are stated in the two following propositions using regularity as a constraint qualification.

Proposition 4.4.1 *(Karush–Kuhn–Tucker)* If the regular point x_* is a local minimizer of problem (4.20), then there exist $\lambda_* \in \mathbb{R}^p$ and $\mu_* \in \mathbb{R}^l$ such that

$$h_i(x_*) = 0,\ i = 1, \ldots, p\,,$$

$$g_i(x_*) \geq 0,\ i = 1, \ldots, l\,,$$

$$\nabla f(x_*) + \sum_{i=1}^{p} (\lambda_*)_i \nabla h_i(x_*) + \sum_{i=1}^{l} (\mu_*)_i \nabla g_i(x_*) = 0\,,$$

$$g_i(x_*)\,(\mu_*)_i = 0,\ i = 1, \ldots, l, \quad \text{and}$$

$$\mu_* \geq 0.$$

These conditions are called the first–order necessary optimality conditions.

The vectors λ_* and μ_* are the Lagrange multipliers. The Lagrangian function associated with problem (4.20) is $f(x) + \sum_{i=1}^{p} \lambda_i h_i(x) + \sum_{i=1}^{l} \mu_i g_i(x)$.

Proposition 4.4.2 *(Karush–Kuhn–Tucker)* If x_* is a regular point for problem (4.20), then the second–order necessary optimality conditions for x_* to be a local minimizer are the existence of $\lambda_* \in \mathbb{R}^p$ and $\mu_* \in \mathbb{R}^l$ such that the first–order necessary optimality conditions hold and

$$\nabla^2 f(x_*) + \sum_{i=1}^{p} (\lambda_*)_i \nabla^2 h_i(x_*) + \sum_{i=1}^{l} (\mu_*)_i \nabla^2 g_i(x_*) \qquad (4.22)$$

is positive semi–definite on the null space of the set of vectors in (4.21).

The second–order sufficient optimality conditions include the first–order necessary optimality conditions and require the matrix (4.22) to be positive definite for every nonzero vector $z \in \mathbb{R}^n$ that satisfies

$$z^T \nabla h_i(x_*) = 0,\ i = 1, \ldots, p\,,$$

$$z^T \nabla g_i(x_*) = 0,\ \text{for } i \in \{1, \ldots, l\} \text{ such that } (\mu_*)_i > 0\,,$$

$$z^T \nabla g_i(x_*) \geq 0,\ \text{for } i \in \{1, \ldots, l\} \text{ such that } (\mu_*)_i = 0\,.$$

We can apply Propositions 4.4.1 and 4.4.2 to problem (4.1) and simplify the result by using the structure of (4.1). This is what we actually do in the rest of this section. The resulting optimality conditions for problem (4.1) are stated in Propositions 4.4.3 and 4.4.4.

A point x_* satisfies the first–order necessary optimality conditions for problem (4.1) if there exist $\lambda_* \in \mathbb{R}^m$ and $\mu_*^a, \mu_*^b \in \mathbb{R}^{n-m}$ such that

$$
\begin{aligned}
& C(x_*) = 0, \quad a \le u_* \le b, \\
& \begin{pmatrix} \nabla_y f(x_*) \\ \nabla_u f(x_*) \end{pmatrix} + \begin{pmatrix} C_y(x_*)^T \lambda_* \\ C_u(x_*)^T \lambda_* \end{pmatrix} - \begin{pmatrix} 0 \\ \mu_*^a \end{pmatrix} + \begin{pmatrix} 0 \\ \mu_*^b \end{pmatrix} = 0, \\
& ((u_*)_i - a_i)\,(\mu_*^a)_i = (b_i - (u_*)_i)\,(\mu_*^b)_i = 0, \quad i = 1, \ldots, n-m\,, \text{ and} \\
& \mu_*^a \ge 0,\ \mu_*^b \ge 0.
\end{aligned}
\tag{4.23}
$$

These conditions are necessary for x_* to be a local solution of (4.1) since the invertibility of $C_y(x_*)$ and the form of the bound constraints on the controls u imply the linear independence of the equality and active inequality constraints (see Definition 4.4.1). We can use the structure of the problem to rewrite the first–order necessary optimality conditions:

$$
\begin{aligned}
& C(x_*) = 0, \quad a \le u_* \le b, \\
& \lambda_* = -C_y(x_*)^{-T} \nabla_y f(x_*), \\
& a_i < (u_*)_i < b_i \quad \Longrightarrow \quad (\nabla_u \ell(x_*, \lambda_*))_i = 0, \\
& (u_*)_i = a_i \qquad\quad \Longrightarrow \quad (\nabla_u \ell(x_*, \lambda_*))_i \ge 0, \text{ and} \\
& (u_*)_i = b_i \qquad\quad \Longrightarrow \quad (\nabla_u \ell(x_*, \lambda_*))_i \le 0.
\end{aligned}
$$

One can obtain a useful form of these conditions by noting that

$$\nabla_u \ell(x_*, \lambda_*) = W(x_*)^T \nabla f(x_*).$$

(See equations (4.14) and (4.15).) In other words, $\nabla_u \ell(x_*, \lambda_*)$ is just the reduced gradient corresponding to the u variables. Hence x_* satisfies the first–order necessary optimality conditions if

$$C(x_*) = 0, \quad a \le u_* \le b,$$

$$\begin{array}{lll} a_i < (u_*)_i < b_i & \Longrightarrow & \left(W(x_*)^T \nabla f(x_*)\right)_i = 0, \\ (u_*)_i = a_i & \Longrightarrow & \left(W(x_*)^T \nabla f(x_*)\right)_i \geq 0, \quad \text{and} \\ (u_*)_i = b_i & \Longrightarrow & \left(W(x_*)^T \nabla f(x_*)\right)_i \leq 0. \end{array}$$

Furthermore, x_* satisfies the second–order necessary optimality conditions for problem (4.1) if it satisfies the first–order necessary optimality conditions, and if the principal submatrix of the reduced Hessian $W(x_*)^T \nabla^2_{xx} \ell(x_*, \lambda_*) W(x_*)$ corresponding to indices i such that $a_i < (u_*)_i < b_i$ is positive semi–definite, where $\lambda_* = -C_y(x_*)^{-T} \nabla_y f(x_*)$.

Now we adapt the idea of Coleman and Li [23] to this context and define $D(x) \in \mathbb{R}^{(n-m)\times(n-m)}$ to be the diagonal matrix with diagonal elements given by

$$(D(x))_{ii} = \begin{cases} (b-u)_i^{\frac{1}{2}} & \text{if} \quad \left(W(x)^T \nabla f(x)\right)_i < 0 \text{ and } b_i < +\infty, \\ 1 & \text{if} \quad \left(W(x)^T \nabla f(x)\right)_i < 0 \text{ and } b_i = +\infty, \\ (u-a)_i^{\frac{1}{2}} & \text{if} \quad \left(W(x)^T \nabla f(x)\right)_i \geq 0 \text{ and } a_i > -\infty, \\ 1 & \text{if} \quad \left(W(x)^T \nabla f(x)\right)_i \geq 0 \text{ and } a_i = -\infty, \end{cases} \tag{4.24}$$

for $i = 1, \ldots, n-m$. In the following proposition we give the form of the first–order and the second–order necessary optimality conditions that we use in Chapters 5 and 6. To us, they indicate the suitability of (4.24) as an affine scaling for (4.1).

Proposition 4.4.3 A point x_* satisfies the first–order necessary optimality conditions for problem (4.1) if and only if

$$C(x_*) = 0\,, \quad a \leq u_* \leq b\,, \quad \text{and}$$

$$D(x_*) W(x_*)^T \nabla f(x_*) = 0.$$

A point x_* satisfies the second–order necessary optimality conditions for problem (4.1) if and only if it satisfies the first–order necessary optimality conditions and

$$D(x_*) W(x_*)^T \nabla^2_{xx} \ell(x_*, \lambda_*) W(x_*) D(x_*)$$

is positive semi–definite. The corresponding Lagrange multipliers are given by $\lambda_* = -C_y(x_*)^{-T} \nabla_y f(x_*)$.

Proposition 4.4.3 remains valid for a larger class of diagonal matrices $D(x)$. The scalar 1 in the definition (4.24) of $D(x)$ can be replaced by any other positive scalar and Proposition 4.4.3 also remains valid with $D(x)$ replaced by $D(x)^p$, $p > 0$. Most of our convergence results in Chapters 5 and 6 still hold if $D(x)$ is replaced by $D(x)^p$, $p \geq 1$. See also Remark 5.5.1. However, the square roots in the definition of $D(x)$ are necessary for the proof of local q–quadratic convergence of our trust–region interior–point reduced SQP algorithms.

The form of the sufficient optimality conditions that we use requires the definition of nondegeneracy or strict complementarity.

Definition 4.4.2 A point x in $\mathcal{B}$ is said to be nondegenerate if

$$\left(W(x)^T\nabla f(x)\right)_i = 0 \quad \Longrightarrow \quad a_i < u_i < b_i \text{ for all } i \in \{1,\ldots,n-m\}.$$

We now define a diagonal $(n-m)\times(n-m)$ matrix $E(x)$ with diagonal elements given by

$$(E(x))_{ii} = \begin{cases} \left|\left(W(x)^T\nabla f(x)\right)_i\right| & \text{if } \left(W(x)^T\nabla f(x)\right)_i < 0 \text{ and } b_i < +\infty, \text{ or} \\ & \text{if } \left(W(x)^T\nabla f(x)\right)_i > 0 \text{ and } a_i > -\infty, \\ 0 & \text{in all other cases,} \end{cases} \tag{4.25}$$

for $i = 1,\ldots,n-m$. The significance of this matrix becomes clear in Section 5.1 when we apply Newton's method to the system of nonlinear equations arising from the first–order necessary optimality conditions. From the definitions of $D(x)$ and $E(x)$ we have the following property. The proof is simple and we omit it.

Proposition 4.4.4 A nondegenerate point x_* satisfies the second–order sufficient optimality conditions for problem (4.1) if and only if it satisfies the first–order necessary optimality conditions and

$$D(x_*)W(x_*)^T\nabla^2_{xx}\ell(x_*,\lambda_*)W(x_*)D(x_*) + E(x_*)$$

is positive definite, where $\lambda_* = -C_y(x_*)^{-T}\nabla_y f(x_*)$.

4.5 Optimal Control Examples

The two examples that we describe in this section are used in Chapters 5 and 6 to test our trust–region interior–point reduced SQP algorithms.

4.5.1 Boundary Control of a Nonlinear Heat Equation

An application that has the structure described in this chapter is the control of a heating process. In this section we introduce a simplified model for the heating of a probe in a kiln discussed by Burger and Pogu [12]. The temperature $y(x,t)$ inside the probe is governed by a nonlinear parabolic partial differential equation. The spatial domain is given by $(0,1)$. The boundary $x = 1$ is the inside of the probe and $x = 0$ is the boundary of the probe¶.

The goal is to control the heating process in such a way that the temperature inside the probe follows a certain desired temperature profile $y_d(t)$. The control $u(t)$ acts on the boundary $x = 0$. The problem can be formulated as follows [12]:

$$\text{minimize} \quad \frac{1}{2}\int_0^T \Big((y(1,t) - y_d(t))^2 + \gamma u^2(t)\Big)dt \tag{4.26}$$

subject to

$$\begin{aligned}
\tau(y(x,t))\tfrac{\partial y}{\partial t}(x,t) - \partial_x(\kappa(y(x,t))\partial_x y(x,t)) &= q(x,t), \quad (x,t) \in (0,1) \times (0,T),\\
\kappa(y(0,t))\partial_x y(0,t) &= g\Big(y(0,t) - u(t)\Big), \quad t \in (0,T),\\
\kappa(y(1,t))\partial_x y(1,t) &= 0, \quad t \in (0,T),\\
y(x,0) &= y_0(x), \quad x \in (0,1),\\
u_{low} \le \quad u & \le u_{upp},
\end{aligned}$$

where $y \in L^2(0,T;H^1(0,1))$, and $u \in L^2(0,T)$. The functions τ, $\kappa \in C^1(\mathbb{R})$ denote the specific heat capacity and the heat conduction, respectively. $y_0 \in H^1(0,1)$ is the initial temperature distribution, $q \in L^2(0,T;H^1(0,1))$ is the source term, g is a given scalar, and γ is a positive regularization parameter. Here $u_{low}, u_{upp} \in L^\infty(0,T)$ are given functions. It is shown in [12] that if the functions τ and κ satisfy

$$\begin{aligned}
&0 < \tau_1 \le \tau(t) \le \tau_2\,, \quad |\tau'(t)| \le \tau_3\,,\\
&0 < \kappa_1 \le \kappa(t) \le \kappa_2\,, \quad |\kappa'(t)| \le \kappa_3\,, \quad \text{for all } t > 0\,,
\end{aligned}$$

then the state equation has a unique solution in the state space

$$\Big\{y \;:\; y \in L^\infty(0,T;H^1(0,1)),\; y' \in L^2(0,T;H^1(0,1)')\Big\}$$

¶The notation x used here for the spatial variables should not be confused with the n dimensional vector x formed by the y and u components.

and there exists a solution for the control problem with no bound constraints.

If the partial differential equation and the integral are discretized, we obtain an optimization problem of the form (4.1). The discretization uses finite elements and was introduced in [12] (see also [75], [89]). The spatial domain $(0,1)$ is divided into N_x subintervals of equidistant length, and the spatial discretization is done using piecewise linear finite elements. The time discretization is performed by partitioning the interval $[0,T]$ into N_t equidistant subintervals. Then the backward Euler method is used to approximate the state space in time, and piecewise constant functions are used to approximate the control space. With this discretization scheme, $C_y(x)$ is a block bidiagonal matrix with tridiagonal blocks resulting from stiffness and mass matrices. Hence linear systems with $C_y(x)$ and $C_y(x)^T$ can be solved efficiently. It is shown in [89][Lemma 3.1] that if

$$\frac{\Delta t}{h^2} < \frac{1}{6}\left(\frac{\tau_2}{\kappa_1} - \frac{\tau_1}{\kappa_2}\right)^{-1},$$

where $\Delta t = \frac{T}{N_t}$ and $h = \frac{1}{N_x}$, then these tridiagonal blocks are nonsingular. Thus $C_y(x)$ is also nonsingular.

4.5.2 Distributed Control of a Semi–Linear Elliptic Equation

The second example is the distributed control of a semi–linear elliptic equation discussed by Heinkenschloss and Vicente [77]. The control problem is given by

$$\text{minimize} \quad \frac{1}{2}\int_\Omega \left((y-y_d)^2 + \gamma u^2\right) dx \tag{4.27}$$

over all y and u satisfying the state equation

$$\begin{aligned} -\Delta y + g(y) &= u, && \text{in } \Omega, \\ y &= d, && \text{on } \partial\Omega, \end{aligned} \tag{4.28}$$

and the control constraints

$$u_{low} \le u \le u_{upp}, \tag{4.29}$$

where $y \in H^1(\Omega)$, $u \in L^2(\Omega)$, $u_{low}, u_{upp} \in L^\infty(\Omega)$ are given functions, and Ω is a bounded domain of $\mathbb{R}^N$, $N = 1,2,3$, with boundary $\partial\Omega$.

The state equation (4.27) is related to the time dependent problem $\frac{\partial y}{\partial t} = \Delta y + e^y$, $t > 0$, that arises in thermal self–ignition of a chemically active mixture of gases in a vessel as described in Gel'fand [57].

For $g(y) = -\lambda e^y$, $u = 0$, and $d = 0$, the state equation (4.27) reduces to the Bratu problem:

$$\begin{aligned} -\Delta y &= \lambda e^y, && \text{in } \Omega, \\ y &= 0, && \text{on } \partial\Omega. \end{aligned} \tag{4.30}$$

This problem models diffusion phenomena in combustion and semiconductors and has become a standard test problem for solvers of systems of nonlinear equations (see the description by Glowinski and Keller in the collection of nonlinear model problems assembled by Moré [104].) The numerical treatment by finite element methods and the solvability of the Bratu problem is discussed in [62, Section IV.2], [63].

For the discretization of this optimal control problem one can use piecewise linear finite elements for both the states and the controls. This leads to a discretized optimal control problem of the form (4.1).

4.6 Problem Scaling

An important numerical issue, that is addressed in our implementation of the algorithms presented in Chapter 5 is the problem scaling inherent in optimal control problems. As we pointed out, the problems we are primarily interested in are discretizations of optimal control problems governed by partial differential equations. The infinite dimensional problem structure greatly influences the finite dimensional problem. In our implementation, we take this into account by allowing the scalar products for the states y, the controls u, and the duality pairing needed to represent $\lambda^T C(y,u)$ to be chosen so that they are discretizations of proper infinite dimensional ones. It is beyond the scope of this thesis to give a comprehensive theoretical study of these issues, but it is important to notice that the formulation of the algorithms in Chapters 5 and 6 fully support the use of such scalar products without any changes. This is a great advantage. In some of the numerical experiments reported in [22], [72], this improved the performance of our algorithms significantly, it avoided artificial ill–conditioning, and it enhanced the quality of the solution computed for a given stopping tolerance.

Chapter 5

Trust–Region Interior–Point Reduced SQP Algorithms for a Class of Nonlinear Programming Problems

Nonlinear programming problems of the form (4.1) originating from optimal control problems governed by large systems of differential equations are the targets of the algorithms introduced in this chapter.

Our algorithms are reduced SQP algorithms that use trust–region interior–point (TRIP) techniques to guarantee global convergence and to handle the bound constraints on the controls (see also Dennis, Heinkenschloss, and Vicente [36]). As we described in Chapter 4, the structure of optimal control problems given in Section 4.1 can be used to implement and analyze SQP algorithms. In particular, to implement reduced SQP algorithms, it is sufficient to compute quantities of the form $C_y(x)v_y$, $C_y(x)^T v_y$, $C_u(x)v_u$, $C_u(x)^T v_y$, and to compute solutions of the linearized state equation $C_y(x)v_y = r$, and of the adjoint equation $C_y(x)^T v_y = r$. This is an important observation, because these are tasks that arise naturally in the context of optimal control problems. In fact, all of the early SQP algorithms, and many of the recent ones rely on matrix factorizations, like the QR, of the Jacobian $J(x)$ of $C(x)$. For the applications we have in mind this is not feasible. As we discussed in Section 4.3, the involved matrices are too large to perform such computations and very often these matrices are not even available in explicit form. On the other hand, matrix–vector multiplications $C_y(x)v_y$, $C_y(x)^T v_y$, $C_u(x)v_u$, $C_u(x)^T v_y$ can be performed and efficient solvers for the linearized state equation $C_y(x)v_y = r$, and the adjoint equation $C_y(x)^T v_y = r$ often are available.

A purely local analysis for the case with no bounds constraints has being given in [83], [86], [87], [89]. However, we consider here the much more difficult issue of incorporating all this structure into an algorithm that converges globally and handles bound constraints on the control variables u.

The global convergence of our algorithms is guaranteed by a trust–region strategy. In our framework the trust region serves a dual purpose. Besides ensuring global convergence, trust regions also introduce a regularization of the subproblems which is related to the Tikhonov regularization [138] as we saw in Section 2.3.4. For the solution of optimal control problems, the partitioning of the variables into states y and controls u motivates a partial decoupling of step components that leads to interesting alternatives for the choice of the trust regions. In Section 5.2.2, we use the structure of problem (4.1) and adapt to this case the decoupled and coupled trust–region approaches introduced in Section 3.4.2 for equality–constrained optimization. As indicated by the names, in the decoupled approach the trust region acts on step components separately. This allows a more efficient implementation of algorithms for the computation of these steps. However, for problems with ill–conditioned state equations, this decoupling does not give an accurate estimate of the size of the steps and might lead to poor performance. In this situation the coupled approach is better, and so we include both.

For the treatment of the bound constraints on u we use a primal–dual affine scaling interior–point algorithm introduced by Coleman and Li [24] for problems with simple bounds. Interior–point approaches are attractive for problems with a large number of bounds. In our context, the affine scaling interior–point algorithm is also of interest, because it does not interfere with the structure of the problem. To apply this algorithm, no additional information is required from the user. This or similar interior–point approaches have recently also been used e.g. in [7], [25], [94], [95], [118]. The advantage of the approach in [24] is that the scaling matrix is determined by the distance of the iterates to the bounds and by the direction of the gradient. This dependence on the direction of the gradient is important for global convergence and its good effect can be seen in numerical examples, see e.g. Figures 5.5 and 5.6.

We believe that the features and strong theoretical properties of these algorithms make them very attractive and powerful tools for the solution of optimal control problems. We applied them to a boundary nonlinear parabolic control problem, see Section 5.8, and a distributed nonlinear elliptic control problem, see Section 6.5. The numerical results are quite satisfactory. Our algorithms have also been applied successfully to optimal control problems arising in fluid flow [22], [72].

This chapter is organized as follows. In Section 5.1, we discuss the application of Newton's method to the system of nonlinear equations arising from the first–order necessary optimality conditions. This is important for the derivation of our TRIP reduced SQP algorithms. We describe these algorithms in Section 5.2. Sections 5.2.1

and 5.2.2 contain a description of the quasi–normal component and of the tangential component. As noticed previously, the partial decoupling of the step components motivated by the partitioning of the variables into states y and controls u and the roles of the decoupled and coupled trust–region approaches are exposed in Section 5.2.2. A complete statement of the TRIP reduced SQP algorithms is given in Section 5.2.4.

The convergence theory for these algorithms is given in Sections 5.3, 5.4, 5.5, and 5.6. Section 5.3 contains some technical results. In Section 5.4, Theorem 5.4.1, we establish global convergence of the iterates to solutions of the first–order necessary optimality conditions. This result is established under very mild assumptions on the steps, the quadratic models, and the Lagrange multipliers. It simultaneously extends the results presented recently by Coleman and Li [24] for simple bounds and those of Dennis, El–Alem, and Maciel [35] (see Theorem 3.6.1 in this thesis) for equality constraints. Under additional conditions, we show convergence of the iterates to non-degenerate solutions of the second–order necessary optimality conditions in Theorem 5.5.2, Section 5.5. This latter result simultaneously extends those by Coleman and Li [24] for simple bounds and those by Dennis and Vicente [42] (see Theorem 3.6.3 in this thesis) for equality constraints. See Figures 1.1 and 1.2. A q–quadratic rate of convergence is proven in Section 5.6. Our analysis allows the application of a variety of methods for the computation of the step components s^{q} and $s^{\mathsf{t}} = W(x)s_u$. In Section 5.7, we discuss practical algorithms for the computation of steps and the Lagrange multipliers that are currently used in our implementation. Numerical results are reported in Section 5.8.

5.1 Application of Newton's Method

One way to motivate the algorithms described in this chapter is to apply Newton's method to the system of nonlinear equations

$$\begin{aligned} C(x) &= 0, \\ D(x)^2 W(x)^T \nabla f(x) &= 0, \end{aligned} \tag{5.1}$$

where $x = (y^T, u^T)^T$ is strictly feasible with respect to the bounds on the variables u, i.e. $a < u < b$. This is related to Goodman's approach [68] for an orthogonal null–space basis and equality constraints (see the discussion at the end of Section 3.2). Although $D(x)^2$ is usually discontinuous at points where $\left(W(x)^T \nabla f(x)\right)_i = 0$, the function $D(x)^2 W(x)^T \nabla f(x)$ is continuous (but not differentiable) at such points. This can be observed in Figures 5.1 and 5.2. The application of Newton's method to this type of systems of nondifferentiable equations has first been suggested by Coleman and Li [23] in the context of nonlinear optimization problems with simple bounds. They showed that this type of nondifferentiability still allows the Newton process to achieve local q–quadratic convergence. In order to apply Newton's method we first need to compute some derivatives.

To calculate the Jacobian of the reduced gradient $W(x)^T \nabla f(x)$, we write

$$W(x)^T \nabla f(x) = \nabla_u f(x) + C_u(x)^T \lambda,$$

where λ is given by $C_y(x)^T \lambda = -\nabla_y f(x)$ and has derivatives

$$\begin{aligned}
\tfrac{\partial \lambda}{\partial y} &= -C_y(x)^{-T} \left(\textstyle\sum_{i=1}^m \nabla^2_{yy} c_i(x) \lambda_i + \nabla^2_{yy} f(x) \right) \\
&= -C_y(x)^{-T} \nabla^2_{yy} \ell(x, \lambda), \\
\tfrac{\partial \lambda}{\partial u} &= -C_y(x)^{-T} \left(\textstyle\sum_{i=1}^m \nabla^2_{yu} c_i(x) \lambda_i + \nabla^2_{yu} f(x) \right) \\
&= -C_y(x)^{-T} \nabla^2_{yu} \ell(x, \lambda).
\end{aligned}$$

This implies the equalities

$$\begin{aligned}
\tfrac{\partial}{\partial y} \left(W(x)^T \nabla f(x) \right) &= C_u(x)^T \tfrac{\partial \lambda}{\partial y} + \nabla^2_{uy} f(x) + \textstyle\sum_{i=1}^m \nabla^2_{uy} c_i(x) \lambda_i \\
&= W(x)^T \begin{pmatrix} \nabla^2_{yy} \ell(x, \lambda) \\ \nabla^2_{uy} \ell(x, \lambda) \end{pmatrix}, \\
\tfrac{\partial}{\partial u} \left(W(x)^T \nabla f(x) \right) &= C_u(x)^T \tfrac{\partial \lambda}{\partial u} + \nabla^2_{uu} f(x) + \textstyle\sum_{i=1}^m \nabla^2_{uu} c_i(x) \lambda_i \\
&= W(x)^T \begin{pmatrix} \nabla^2_{yu} \ell(x, \lambda) \\ \nabla^2_{uu} \ell(x, \lambda) \end{pmatrix},
\end{aligned}$$

and we can conclude that

$$\frac{d}{dx} \left(W(x)^T \nabla f(x) \right) = W(x)^T \nabla^2_{xx} \ell(x, \lambda),$$

where $\lambda = -C_y(x)^{-T} \nabla_y f(x)$.

A linearization of (5.1) gives

$$\begin{aligned}
C_y(x) s_y + C_u(x) s_u &= -C(x), & (5.2) \\
\left(D(x)^2 W(x)^T \nabla^2_{xx} \ell(x, \lambda) + \begin{pmatrix} 0 & E(x) \end{pmatrix} \right) \begin{pmatrix} s_y \\ s_u \end{pmatrix} &= -D(x)^2 W(x)^T \nabla f(x), & (5.3)
\end{aligned}$$

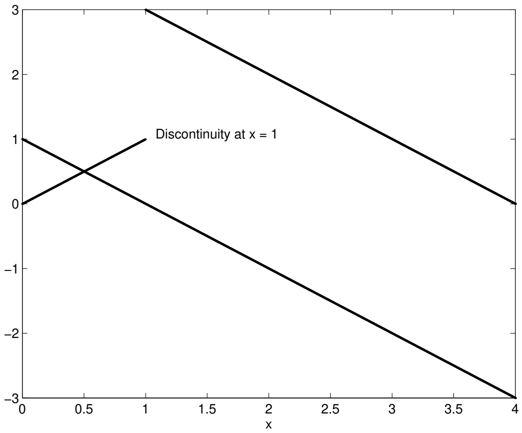

Figure 5.1 Plots of $D(x)^2$ and $W(x)^T\nabla f(x)$ for $W(x)^T\nabla f(x) = -x+1$ and $x \in [0,4]$.

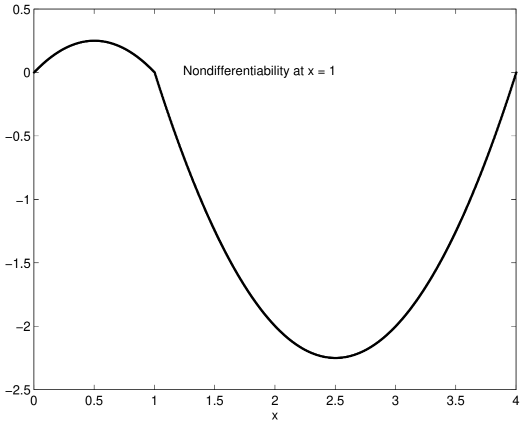

Figure 5.2 Plot of $D(x)^2 W(x)^T\nabla f(x)$ for $W(x)^T\nabla f(x) = -x+1$ and $x \in [0,4]$.

where 0 denotes the $(n-m)\times m$ matrix with zero entries. Equation (5.2) is the linearized state equation. The matrix $E(x)$ was defined in (4.25), Section 4.4. The diagonal elements of $E(x)$ are the product of the derivative of the diagonal elements of $D(x)^2$ and the components of the reduced gradient $W(x)^T\nabla f(x)$. The derivative of $(D(x)^2)_{ii}$ does not exist if $\left(W(x)^T\nabla f(x)\right)_i = 0$. In this case we set the corresponding quantities in the Jacobian to zero (see references [24], [23]). This gives the equation (5.3).

By using (4.3) we can rewrite the linear system (5.2)–(5.3) as

$$\begin{aligned} s &= s^{\mathsf{q}} + W(x)s_u, \\ &\Big(D(x)^2 W(x)^T \nabla^2_{xx}\ell(x,\lambda)W(x) + E(x)\Big)\, s_u \\ &\quad = -D(x)^2 W(x)^T \left(\nabla^2_{xx}\ell(x,\lambda)s^{\mathsf{q}} + \nabla f(x)\right). \end{aligned} \tag{5.4}$$

We define our Newton–like step as the solution of

$$s = s^{\mathsf{q}} + W(x)s_u, \tag{5.5}$$

$$\begin{aligned} &\Big(\bar{D}(x)^2 W(x)^T \nabla^2_{xx}\ell(x,\lambda)W(x) + E(x)\Big)\, s_u \\ &\quad = -\bar{D}(x)^2 W(x)^T \left(\nabla^2_{xx}\ell(x,\lambda)s^{\mathsf{q}} + \nabla f(x)\right), \end{aligned} \tag{5.6}$$

where $\bar{D}(x) \in \mathbb{R}^{(n-m)\times(n-m)}$ is the diagonal matrix defined by

$$(\bar{D}(x))_{ii} = \begin{cases} (b-u)_i^{\frac{1}{2}} & \text{if} \quad \left(W(x)^T\left(\nabla_{xx}^2 \ell(x,\lambda)s^{\mathsf{q}} + \nabla f(x)\right)\right)_i < 0 \text{ and } b_i < +\infty, \\ 1 & \text{if} \quad \left(W(x)^T\left(\nabla_{xx}^2 \ell(x,\lambda)s^{\mathsf{q}} + \nabla f(x)\right)\right)_i < 0 \text{ and } b_i = +\infty, \\ (u-a)_i^{\frac{1}{2}} & \text{if} \quad \left(W(x)^T\left(\nabla_{xx}^2 \ell(x,\lambda)s^{\mathsf{q}} + \nabla f(x)\right)\right)_i \geq 0 \text{ and } a_i > -\infty, \\ 1 & \text{if} \quad \left(W(x)^T\left(\nabla_{xx}^2 \ell(x,\lambda)s^{\mathsf{q}} + \nabla f(x)\right)\right)_i \geq 0 \text{ and } a_i = -\infty, \end{cases} \tag{5.7}$$

for $i = 1, \ldots, n-m$. This change of the diagonal scaling matrix is based on the form of the right hand side of (5.4).

If x is close to a nondegenerate point x_* satisfying the second–order sufficient optimality conditions and if $W(x)^T \nabla_{xx}^2 \ell(x,\lambda)s^{\mathsf{q}}$ is sufficiently small, a step s defined in this way is a Newton step on the following system of nonlinear equations

$$\begin{aligned} &C(x) = 0, \\ &D(x)_u^2 W(x)^T \nabla f(x) = 0, \end{aligned} \tag{5.8}$$

where $D(x)_u$ depends on x_* as follows:

$$(D(x)_u)_{ii} = \begin{cases} 1 \text{ or } (b-u)_i^{\frac{1}{2}} \text{ or } (u-a)_i^{\frac{1}{2}} & \text{if } \left(W(x_*)^T \nabla f(x_*)\right)_i = 0, \\ (b-u)_i^{\frac{1}{2}} & \text{if } \left(W(x_*)^T \nabla f(x_*)\right)_i < 0, \\ (u-a)_i^{\frac{1}{2}} & \text{if } \left(W(x_*)^T \nabla f(x_*)\right)_i > 0, \end{cases}$$

for $i = 1, \ldots, n-m$. If $\left(W(x_*)^T \nabla f(x_*)\right)_i = 0$, the i–th principal diagonal element of $D(x)_u$ has to be chosen so that $D(x)_u$ and $\bar{D}(x)$ are the same matrices. Of course, this depends on the sign of $\left(W(x)^T(\nabla_{xx}^2 \ell(x,\lambda)s^{\mathsf{q}} + \nabla f(x))\right)_i$. As Coleman and Li [23] pointed out, $D(x)_u$ is just of theoretical use since x_* is unknown. One can see that $D(x)_u^2 W(x)^T \nabla f(x)$ is continuously differentiable with Lipschitz continuous derivatives in an open neighborhood of x_*, that $D(x_*)_u^2 W(x_*)^T \nabla f(x_*) = 0$, and that the Jacobian of $D(x)_u^2 W(x)^T \nabla f(x)$ at x_* is nonsingular, for all choices of $D(x)_u$. These conditions are those typically required to get q–quadratic convergence for the Newton iteration (see [39][Theorem 5.2.1]). The interior–point process damps the Newton step so that it stays strictly feasible but this does affect the rate of convergence. The details are provided in Corollary 5.6.1.

5.2 Trust–Region Interior–Point Reduced SQP Algorithms

The algorithms that we propose generate a sequence of iterates $\{x_k\}$ where

$$x_k = \begin{pmatrix} y_k \\ u_k \end{pmatrix},$$

and u_k is strictly feasible with respect to the bounds, i.e. $a < u_k < b$. At iteration k we are given x_k, and we need to compute a step s_k. If s_k is accepted, we set $x_{k+1} = x_k + s_k$. Otherwise, we set x_{k+1} to x_k, reduce the trust–region radius, and compute a new step.

Following the application of Newton's method (5.5), each step s_k is decomposed as

$$s_k = s_k^{\mathsf{q}} + s_k^{\mathsf{t}} = s_k^{\mathsf{q}} + W_k (s_k)_u,$$

where s_k^{q} is called the quasi–normal component and s_k^{t} is the tangential component. The role of s_k^{q} is to move towards feasibility whereas the role of s_k^{t} is to move towards optimality. The definition of the quasi–normal component, the tangential component, as well as the complete formulation of our algorithms is the content of this section.

5.2.1 The Quasi–Normal Component

Let δ_k be the trust radius at iteration k. The quasi–normal component s_k^{q} is related to the trust–region subproblem for the linearized constraints

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\|J_k s^{\mathsf{q}} + C_k\|^2 \\ \text{subject to} & \|s^{\mathsf{q}}\| \le \delta_k, \end{array}$$

and it is required to have the form

$$s_k^{\mathsf{q}} = \begin{pmatrix} (s_k^{\mathsf{q}})_y \\ 0 \end{pmatrix}. \tag{5.9}$$

Thus the displacement along s_k^{q} is made only in the y variables, and as a consequence, x_k and $x_k + s_k^{\mathsf{q}}$ have the same u components. The calculation of the quasi–normal component is illustrated in Figures 5.3 and 5.4. Since $(s_k^{\mathsf{q}})_u = 0$, the trust–region subproblem introduced above can be rewritten as

$$\text{minimize} \quad \frac{1}{2}\|C_y(x_k)(s^{\mathsf{q}})_y + C_k\|^2 \tag{5.10}$$

$$\text{subject to} \quad \|(s^{\mathsf{q}})_y\| \le \delta_k. \tag{5.11}$$

Thus, the quasi–normal component s_k^{q} is a trust–region globalization of the component s^{q} given in (4.4) of the Newton step (5.5). We do not have to solve (5.10)–(5.11) exactly, we only have to assume that the quasi–normal component satisfies the conditions

$$\|s_k^{\mathsf{q}}\| \leq \kappa_1 \|C_k\| \tag{5.12}$$

and

$$\begin{aligned} &\|C_k\|^2 - \|C_y(x_k)(s_k^{\mathsf{q}})_y + C_k\|^2 \geq \kappa_2 \|C_k\| \min\{\kappa_3 \|C_k\|, \delta_k\}\,, \\ &\|(s_k^{\mathsf{q}})_y\| \leq \delta_k\,, \end{aligned} \tag{5.13}$$

where κ_1, κ_2, and κ_3 are positive constants independent of k. In Section 6.3, we describe several ways of computing a quasi–normal component that satisfies the requirements (5.9), (5.12), and (5.13). Condition (5.12) tells us that the quasi–normal component is small close to feasible points. The decrease condition (5.13) is a form of Cauchy decrease or simple decrease for the trust–region subproblem (5.10)–(5.11). See Section 3.4.1 for more details.

5.2.2 The Tangential Component

The computation of the tangential component $(s_k)_u$ follows a trust–region globalization of the Newton step (5.6). Following Coleman and Li [24] we symmetrize (5.6) and get

$$\left(\bar{D}_k W_k^T H_k W_k \bar{D}_k + E_k\right) \bar{D}_k^{-1} s_u = -\bar{D}_k W_k^T \left(H_k s_k^{\mathsf{q}} + \nabla f_k\right),$$

where $\bar{D}_k = \bar{D}(x_k)$, $E_k = E(x_k)$, and H_k denotes a symmetric approximation to the Hessian matrix $\nabla_{xx}^2 \ell_k$. This suggests the change of variables $\hat{s}_u = \bar{D}_k^{-1} s_u$ and the consideration in the scaled space $\hat{s}_u$ of the trust–region subproblem:

$$\begin{array}{ll} \text{minimize} & \left(\bar{D}_k W_k^T \left(H_k s_k^{\mathsf{q}} + \nabla f_k\right)\right)^T \hat{s}_u + \dfrac{1}{2} \hat{s}_u^T \left(\bar{D}_k W_k^T H_k W_k \bar{D}_k + E_k\right) \hat{s}_u \\ \text{subject to} & \|\hat{s}_u\| \leq \delta_k. \end{array}$$

Now we can rewrite the previous subproblem in the unscaled space s_u as

$$\begin{array}{ll} \text{minimize} & \left(W_k^T \left(H_k s_k^{\mathsf{q}} + \nabla f_k\right)\right)^T s_u + \frac{1}{2} s_u^T \left(W_k^T H_k W_k + E_k \bar{D}_k^{-2}\right) s_u \\ \text{subject to} & \|\bar{D}_k^{-1} s_u\| \leq \delta_k. \end{array} \tag{5.14}$$

Of course, we also have to require that the new iterate is in the interior of the box constraints. To ensure that $u_k + s_k$ is strictly feasible with respect to the box constraints we choose $\sigma_k \in [\sigma, 1)$, $\sigma \in (0,1)$, and compute s_u with $\sigma_k(a - u_k) \leq s_u \leq \sigma_k(b - u_k)$. However, one of the strength of this trust–region approach is that we can allow for approximate solutions of this subproblem with or without the bound constraints. In particular, it is not necessary to solve the full trust–region subproblem including the box constraints. For example, one can compute the solution of the trust–region subproblem without the box constraints and then scale the computed solution back so that the resulting damped s_u obeys $\sigma_k(a-u_k) \leq s_u \leq \sigma_k(b-u_k)$. We show that under suitable assumptions this strategy guarantees global convergence and local q–quadratic convergence. Another way to compute an approximate u component of the step is to use a modification of the Conjugate–Gradient Algorithm 2.3.2 for the trust–region subproblem that is truncated if one of the bounds $\sigma_k(a - u_k) \leq s_u \leq \sigma_k(b - u_k)$ is violated. See Section 5.7.1. More ways to compute the tangential component are possible. The conditions on the tangential component necessary to guarantee global convergence are stated later in this section.

We now introduce a quadratic model

$$\Psi_k(s_u) = q_k(s_k^{\mathrm{q}} + W_k s_u) + \frac{1}{2} s_u^T \left(E_k \bar{D}_k^{-2} \right) s_u, \tag{5.15}$$

where, as in Section 3.2,

$$q_k(s_k^{\mathrm{q}} + W_k s_u) = q_k(s_k^{\mathrm{q}}) + \bar{g}_k^T s_u + \frac{1}{2} {s_u}^T W_k^T H_k W_k s_u \tag{5.16}$$

is a quadratic model of $\ell(x_k + s, \lambda_k)$ about (x_k, λ_k), and

$$\bar{g}_k = W_k^T \nabla q_k(s_k^{\mathrm{q}}) = W_k^T \left(H_k s_k^{\mathrm{q}} + \nabla f_k \right).$$

The Decoupled Trust–Region Approach

We can restate the trust–region subproblem (5.14) as

$$\text{minimize} \quad \Psi_k(s_u) \tag{5.17}$$

$$\text{subject to} \quad \|\bar{D}_k^{-1} s_u\| \leq \delta_k. \tag{5.18}$$

We refer to the approach based on this subproblem as the decoupled approach. In this decoupled approach the trust–region constraint is of the form $\|\bar{D}_k^{-1} s_u\| \leq \delta_k$ corresponding to the constraint $\|\hat{s}_u\| \leq \delta_k$ in the scaled space. One can see from (5.11) and (5.18) that we are imposing the trust region separately on the y part of the quasi–normal component and on the u part of the tangential component. (In Figure 5.3, the tangential component is depicted for the decoupled approach.) Moreover, if the cross–term $W_k^T H_k s_k^{\mathsf{q}}$ is set to zero, then the trust–region subproblems for the quasi–normal component and for the tangential component are completely separated.

The Coupled Trust–Region Approach

The approach we now present forces both the y and the u components of the tangential component $s_k^{\mathsf{t}} = W_k(s_k)_u$ to lie inside a trust region of radius δ_k. See Figure 5.4. The reference trust–region subproblem is given by

$$\text{minimize} \quad \Psi_k(s_u) \tag{5.19}$$

$$\text{subject to} \quad \left\| \begin{pmatrix} -C_y(x_k)^{-1}C_u(x_k)s_u \\ \bar{D}_k^{-1}s_u \end{pmatrix} \right\| \le \delta_k. \tag{5.20}$$

Recall from Section 3.4.2 that in the case where there are no bounds on u this trust–region constraint is of the form

$$\left\| \begin{pmatrix} -C_y(x_k)^{-1}C_u(x_k)s_u \\ s_u \end{pmatrix} \right\| \; = \; \|W_k s_u\| \; \le \; \delta_k.$$

As opposed to the decoupled case, one can see that the term $C_y(x_k)^{-1}C_u(x_k)s_u$ is present in the trust–region constraint (5.20). If W_k^+ denotes the Moore–Penrose pseudo inverse of W_k (see [66][Section 5.5.4]), then

$$\frac{1}{\|W_k^+\|}\|s_u\| \le \|W_k s_u\| \le \|W_k\|\|s_u\|.$$

Thus, if the condition number $\kappa(W_k) = \|W_k^+\| \, \|W_k\|$ is small, then the decoupled and the coupled approach generate similar iterates. In this case, the decoupled approach is more efficient since it uses fewer linear system solves with the system matrix $C_y(x_k)$. See Section 5.7.1. However, if $\kappa(W_k)$ is large, e.g. if $C_y(x_k)$ is ill–conditioned, then the coupled approach uses the size of the tangential component s^{t}, whereas the decoupled approach may underestimate vastly the size of this step component. This can lead to poor performance of the decoupled approach when steps are rejected and the trust–region radius is reduced based on the incorrect estimate $\|s_u\|$ of the norm of $s^{\mathsf{t}} = W_k s_u$. This indicates that when $C_y(x)$ is ill–conditioned the coupled approach offers a better regularization of the step.

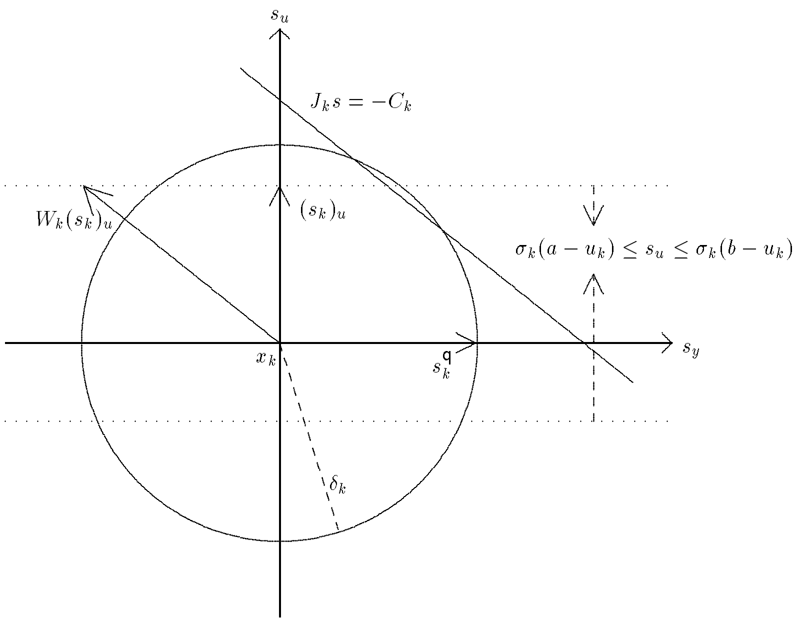

Figure 5.3 The quasi–normal and tangential components of the step for the decoupled approach. We assume for simplicity that $\bar{D}_k = (\,1\,)$.

Cauchy Decrease for the Tangential Component

To assure global convergence to a point satisfying the first–order necessary optimality conditions, we consider analogs for the subproblems (5.17)–(5.18) and (5.19)–(5.20) of the fraction of Cauchy decrease condition (2.7) for the unconstrained optimization problem.

First we consider the decoupled trust–region subproblem (5.17)–(5.18). The Cauchy step c_k^{d} for this case is defined as the solution of

$$\begin{array}{ll} \text{minimize} & \Psi_k(c^{\mathsf{d}}) \\ \text{subject to} & \|\bar{D}_k^{-1} c^{\mathsf{d}}\| \le \delta_k, \quad c^{\mathsf{d}} \in span\{-\bar{D}_k^2 \bar{g}_k\}, \\ & \sigma_k(a - u_k) \le c^{\mathsf{d}} \le \sigma_k(b - u_k), \end{array}$$

where $-\bar{D}_k^2 \bar{g}_k$ is the steepest–descent direction for $\Psi_k(s_u)$ at $s_u = 0$ in the norm $\|\bar{D}_k^{-1} \cdot \|$. (See Section 2.3 for general definitions of Cauchy steps and steepest–descent directions.) Here $\sigma_k \in [\sigma, 1)$ ensures that the Cauchy step c_k^{d} remains strictly feasible with respect to the box constraints. The parameter $\sigma \in (0,1)$ is fixed for all k. We

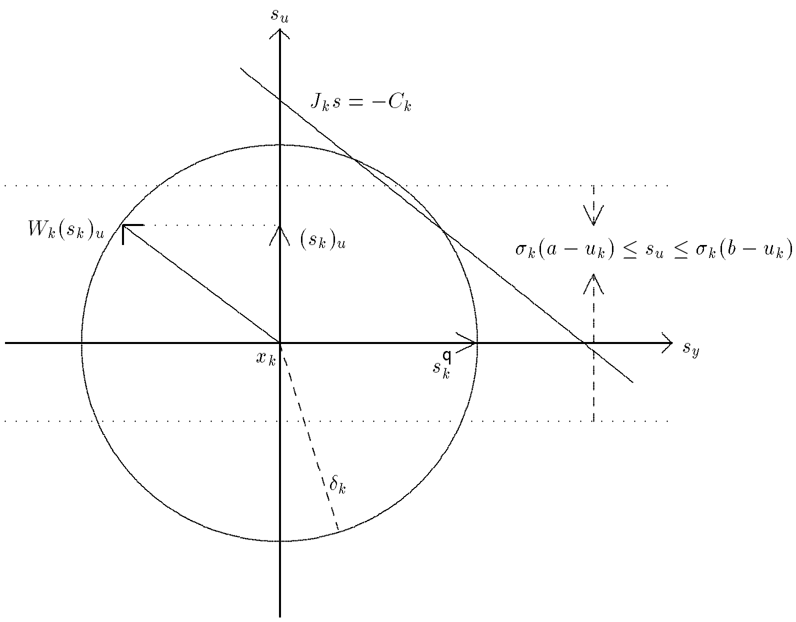

Figure 5.4 The quasi–normal and tangential components of the step for the coupled approach. We assume for simplicity that $\bar{D}_k = (\,1\,)$.

require the tangential component $(s_k)_u$ with $\sigma_k(a - u_k) \leq (s_k)_u \leq \sigma_k(b - u_k)$ to give a decrease on $\Psi_k(s_u)$ smaller than a uniform fraction of the decrease given by c_k^{d} for the same function $\Psi_k(s_u)$. This fraction of Cauchy decrease condition can be stated as

$$\begin{aligned} &\Psi_k(0) - \Psi_k((s_k)_u) \geq \beta_1^{\mathsf{d}} \left(\Psi_k(0) - \Psi_k(c_k^{\mathsf{d}})\right) , \\ &\|(s_k)_u\| \leq \delta_k , \end{aligned} \tag{5.21}$$

where β_1^{d} is positive and fixed across all iterations. It is not difficult to see that dogleg or conjugate–gradient algorithms of the type 2.3.1 and 2.3.2 can compute components $(s_k)_u$ conveniently that satisfy condition (5.21) with $\beta_1^{\mathsf{d}} = 1$. We leave these issues to Section 5.7.1.

In a similar way, the component $(s_k)_u$ with $\sigma_k(a-u_k) \leq (s_k)_u \leq \sigma_k(b-u_k)$ satisfies a fraction of Cauchy decrease for the coupled trust–region subproblem (5.19)–(5.20) if

$$\begin{aligned} &\Psi_k(0) - \Psi_k((s_k)_u) \geq \beta_1^{\mathsf{c}}\left(\Psi_k(0) - \Psi_k(c_k^{\mathsf{c}})\right) , \\ &\left\| \begin{pmatrix} -C_y(x_k)^{-1}C_u(x_k)(s_k)_u \\ \bar{D}_k^{-1}(s_k)_u \end{pmatrix} \right\| \leq \delta_k , \end{aligned} \tag{5.22}$$

for some β_1^{c} independent of k, where the Cauchy step c_k^{c} is the solution of

$$\begin{array}{ll} \text{minimize} & \Psi_k(c^{\mathsf{c}}) \\ \text{subject to} & \left\| \begin{pmatrix} -C_y(x_k)^{-1}C_u(x_k)c^{\mathsf{c}} \\ \bar{D}_k^{-1}c^{\mathsf{c}} \end{pmatrix} \right\| \le \delta_k, \quad c^{\mathsf{c}} \in span\{-\bar{D}_k^2\bar{g}_k\}, \\ & \sigma_k(a-u_k) \le c^{\mathsf{c}} \le \sigma_k(b-u_k). \end{array}$$

In Section 5.7.1, we show how to use conjugate–gradient type algorithms to compute components $(s_k)_u$ satisfying the condition (5.22).

One final comment is in order. In the coupled approach, the Cauchy step c_k^{c} was defined along the direction $-\bar{D}_k^2\bar{g}_k$. To simplify this discussion, suppose that there are no bounds on u. In this case the trust–region constraint is of the form $\|W_k s_u\| \le \delta_k$. The presence of W_k gives the trust region an ellipsoidal shape. The steepest–descent direction for the quadratic (5.15) in the norm $\|W_k \cdot\|$ at $s_u = 0$ is given by $-(W_k^T W_k)^{-1}\bar{g}_k$. Our analysis still holds for this case since $\{\|(W_k^T W_k)^{-1}\|\}$ is a bounded sequence. See the discussion in Section 3.4.2 for the coupled approach. The reason why we avoid the term $(W_k^T W_k)^{-1}$ is that in many applications there is no reasonable way to solve systems with $W_k^T W_k$. We show in Section 5.7.1 how this affects the use of conjugate gradients (see Remark 5.7.1). Finally, we point out that this problem does not arise if the decoupled approach is used.

Optimal Decrease for the Tangential Component

The conditions in the previous subsection are sufficient to guarantee global convergence to a point satisfying first–order necessary optimality conditions, but they are too weak to guarantee global convergence to a point satisfying second–order necessary optimality conditions. To accomplish this, just as in the unconstrained case [106], [132], in the box–constrained case [24], and in the equality–constrained case [42], [48], we need to make sure that s_u satisfies an appropriate fraction of optimal decrease condition.

First we consider the decoupled approach and let o_k^{d} be an optimal solution of the trust–region subproblem (5.17)–(5.18). It follows from the application of Proposition 2.3.3 that there exists $\gamma_k \ge 0$ such that

$$W_k^T H_k W_k + E_k\bar{D}_k^{-2} + \gamma_k\bar{D}_k^{-2} \text{ is positive semi–definite,} \tag{5.23}$$

$$\Big(W_k^T H_k W_k + E_k \bar{D}_k^{-2} + \gamma_k \bar{D}_k^{-2}\Big) o_k^{\mathsf{d}} = -\bar{g}_k, \quad \text{and} \tag{5.24}$$

$$\gamma_k\Big(\delta_k - \|\bar{D}_k^{-1} o_k^{\mathsf{d}}\|\Big) = 0.$$

Since $u_k + o_k^{\mathsf{d}}$ might not be strictly feasible, we consider $\tau_k o_k^{\mathsf{d}}$, where τ_k is given by

$$\tau_k = \sigma_k \; \min\left\{1, \; \max\left\{\frac{b_i - (u_k)_i}{(o_k^{\mathsf{d}})_i}, \; \frac{(u_k)_i - a_i}{(o_k^{\mathsf{d}})_i}\right\}, \; i = 1, \ldots, n-m\right\}. \tag{5.25}$$

With this choice of τ_k, $u_k + \tau_k o_k^{\mathsf{d}}$ is strictly inside the box constraints $\mathcal{B}$.

The tangential component $(s_k)_u$ then is required to satisfy the following fraction of optimal decrease condition

$$\begin{aligned} &\Psi_k(0) - \Psi_k((s_k)_u) \geq \beta_2^{\mathsf{d}} \left(\Psi_k(0) - \Psi_k(\tau_k o_k^{\mathsf{d}})\right), \\ &\|\bar{D}_k^{-1}(s_k)_u\| \leq \beta_3^{\mathsf{d}} \delta_k, \end{aligned} \tag{5.26}$$

where $\beta_2^{\mathsf{d}}, \beta_3^{\mathsf{d}}$ are positive constants independent of k.

From conditions (5.23), (5.24), (5.26), and $\tau_k < 1$, we can write

$$\begin{aligned} \Psi_k(0) - \Psi_k((s_k)_u) &\geq \beta_2^{\mathsf{d}} \left(-\tau_k \bar{g}_k^T o_k^{\mathsf{d}} - \frac{1}{2}\tau_k^2 (o_k^{\mathsf{d}})^T \left(W_k^T H_k W_k + E_k \bar{D}_k^{-2}\right)(o_k^{\mathsf{d}})\right) \\ &\geq \beta_2^{\mathsf{d}} \tau_k \Big(- \bar{g}_k^T o_k^{\mathsf{d}} - \frac{1}{2}(o_k^{\mathsf{d}})^T (W_k^T H_k W_k + E_k \bar{D}_k^{-2} \\ &\qquad + \gamma_k \bar{D}_k^{-2})(o_k^{\mathsf{d}})\Big) + \frac{1}{2}\beta_2^{\mathsf{d}} \tau_k^2 \gamma_k (o_k^{\mathsf{d}})^T \bar{D}_k^{-2}(o_k^{\mathsf{d}}) \\ &\geq \frac{1}{2}\beta_2^{\mathsf{d}} \tau_k \|R_k o_k^{\mathsf{d}}\|^2 + \frac{1}{2}\beta_2^{\mathsf{d}} \tau_k^2 \gamma_k \delta_k^2 \\ &\geq \frac{1}{2}\beta_2^{\mathsf{d}} \tau_k^2 \gamma_k \delta_k^2, \end{aligned} \tag{5.27}$$

where $W_k^T H_k W_k + E_k \bar{D}_k^{-2} + \gamma_k \bar{D}_k^{-2} = R_k^T R_k$.

Now let us focus on the coupled approach and let o_k^{c} be the optimal solution of the trust–region subproblem (5.19)–(5.20). In this case o_k^{c} satisfies

$$W_k^T H_k W_k + E_k \bar{D}_k^{-2} + \gamma_k \left(\bar{D}_k^{-2} + W_k^T W_k - I_{n-m}\right)$$

is positive semi–definite, (5.28)

$$\Big(W_k^T H_k W_k + E_k \bar{D}_k^{-2} + \gamma_k \left(\bar{D}_k^{-2} + W_k^T W_k - I_{n-m}\right)\Big) o_k^{\mathsf{c}} = -\bar{g}_k, \quad \text{and} \tag{5.29}$$

$$\gamma_k \left(\delta_k - \left\| \begin{pmatrix} -C_y(x_k)^{-1} C_u(x_k) o_k^{\mathsf{c}} \\ \bar{D}_k^{-1} o_k^{\mathsf{c}} \end{pmatrix} \right\| \right) = 0.$$

Now we damp o_k^{c} with τ_k given as in (5.25) but with o_k^{d} replaced by o_k^{c}. Thus, the resulting step $u_k + \tau_k o_k^{\mathsf{c}}$ is strictly feasible. We impose the following fraction of optimal decrease condition on the tangential component $(s_k)_u$:

$$\begin{aligned} \Psi_k(0) - \Psi_k((s_k)_u) &\geq \beta_2^{\mathsf{c}} \Big(\Psi_k(0) - \Psi_k(\tau_k o_k^{\mathsf{c}}) \Big) \,, \\ \left\| \begin{pmatrix} -C_y(x_k)^{-1} C_u(x_k)(s_k)_u \\ \bar{D}_k^{-1}(s_k)_u \end{pmatrix} \right\| &\leq \beta_3^{\mathsf{c}} \delta_k \,, \end{aligned} \tag{5.30}$$

where $\beta_2^{\mathsf{c}}, \beta_3^{\mathsf{c}}$ are positive and independent of k. In this case it can be shown in a way similar to (5.27) that

$$\Psi_k(0) - \Psi((s_k)_u) \geq \frac{1}{2} \beta_2^{\mathsf{c}} \tau_k^2 \gamma_k \delta_k^2. \tag{5.31}$$

5.2.3 Reduced and Full Hessians

In the previous section we considered an approximation H_k to the full Hessian $\nabla^2_{xx}\ell_k$. The algorithms and theory presented in this and in the following chapters are also valid if we use an approximation $\tilde{H}_k$ to the reduced Hessian $W_k^T \nabla^2_{xx}\ell_k W_k$. In this case we set

$$H_k = \begin{pmatrix} 0 & 0 \\ 0 & \tilde{H}_k \end{pmatrix}. \tag{5.32}$$

Due to the form of W_k, we have

$$W_k^T H_k W_k = \tilde{H}_k.$$

This allows us to obtain the expansion (5.16) in the context of a reduced Hessian approximation.

For the algorithms with reduced Hessian approximation the following observations are useful:

$$\begin{aligned} H_k d &= \begin{pmatrix} 0 \\ \tilde{H}_k d_u \end{pmatrix}, \\ d^T H_k d &= d_u^T \tilde{H}_k d_u, \\ W_k^T H_k d &= \tilde{H}_k d_u, \end{aligned} \tag{5.33}$$

where $d = \begin{pmatrix} d_y \\ d_u \end{pmatrix} \in \mathbb{R}^n$.

5.2.4 Outline of the Algorithms

We need to introduce the merit function and the corresponding actual and predicted decreases. The merit function used is the augmented Lagrangian

$$L(x, \lambda; \rho) = f(x) + \lambda^T C(x) + \rho C(x)^T C(x).$$

The actual decrease at iteration k is defined as

$$ared(s_k; \rho_k) = L(x_k, \lambda_k; \rho_k) - L(x_k + s_k, \lambda_{k+1}; \rho_k),$$

and the predicted decrease as

$$pred(s_k; \rho_k) = L(x_k, \lambda_k; \rho_k) - \left(q_k(s_k) + \Delta\lambda_k^T (J_k s_k + C_k) + \rho_k \|J_k s_k + C_k\|^2\right),$$

with $\Delta\lambda_k = \lambda_{k+1} - \lambda_k$.

These choices of actual and predicted decreases are the same as in Section 3.4.3 for equality–constrained optimization. A possible redefinition of the actual and predicted decreases is obtained by subtracting the term $\frac{1}{2}(s_k)_u^T \left(E_k \bar{D}_k^{-2}\right)(s_k)_u$ from both $ared(s_k; \rho_k)$ and $pred(s_k; \rho_k)$. This type of modification was suggested in [24] for optimization with simple bounds, and it does not affect the global and local results given in this and in the following chapters.

To decide whether to accept or reject a step s_k, we evaluate the ratio

$$\frac{ared(s_k; \rho_k)}{pred(s_k; \rho_k)}.$$

To update the penalty parameter ρ_k we use the scheme proposed in [47] and already used in Algorithm 3.4.1 for equality constraints.

We now can outline the main steps of the trust–region interior–point (TRIP) reduced sequential quadratic programming (SQP) algorithms. We leave the practical computation of s_k^{q}, $(s_k)_u$, and λ_k to Section 5.7.

Algorithm 5.2.1 *(TRIP Reduced SQP Algorithms)*

1 Choose x_0 such that $a < u_0 < b$, pick $\delta_0 > 0$, and calculate λ_0. Choose α_1, η_1, σ, δ_{min}, δ_{max}, $\bar{\rho}$, and ρ_{-1} such that $0 < \alpha_1, \eta_1, \sigma < 1$, $0 < \delta_{min} \leq \delta_{max}$, $\bar{\rho} > 0$, and $\rho_{-1} \geq 1$.

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

2.1 Stop if (x_k, λ_k) satisfies the stopping criterion.

2.2 Compute s_k^{q} based on the subproblem (5.10)–(5.11).
Compute $(s_k)_u$ based on the subproblem (5.17)–(5.18) (or (5.19)–(5.20) for the coupled approach) satisfying

$$\sigma_k(a - u_k) \leq (s_k)_u \leq \sigma_k(b - u_k),$$

with $\sigma_k \in [\sigma, 1)$. Set $s_k = s_k^{\mathsf{q}} + s_k^{\mathsf{t}} = s_k^{\mathsf{q}} + W_k(s_k)_u$.

2.3 Compute λ_{k+1} and set $\Delta\lambda_k = \lambda_{k+1} - \lambda_k$.

2.4 Compute $pred(s_k; \rho_{k-1})$:

$$q_k(0) - q_k(s_k) - \Delta\lambda_k^T(J_k s_k + C_k) + \rho_{k-1}\Big(\|C_k\|^2 - \|J_k s_k + C_k\|^2\Big).$$

If $pred(s_k; \rho_{k-1}) \geq \frac{\rho_{k-1}}{2}\Big(\|C_k\|^2 - \|J_k s_k + C_k\|^2\Big)$ then set $\rho_k = \rho_{k-1}$. Otherwise set

$$\rho_k = \frac{2\left(q_k(s_k) - q_k(0) + \Delta\lambda_k^T(J_k s_k + C_k)\right)}{\|C_k\|^2 - \|J_k s_k + C_k\|^2} + \bar{\rho}.$$

2.5 If $\frac{ared(s_k;\rho_k)}{pred(s_k;\rho_k)} < \eta_1$, set

$$\delta_{k+1} = \alpha_1 \max\left\{\|s_k^{\mathsf{q}}\|, \|\bar{D}_k^{-1}(s_k)_u\|\right\} \text{ in the decoupled case or}$$

$$\delta_{k+1} = \alpha_1 \max\left\{\|s_k^{\mathsf{q}}\|, \left\|\begin{pmatrix} -C_y(x_k)^{-1}C_u(x_k)(s_k)_u \\ \bar{D}_k^{-1}(s_k)_u \end{pmatrix}\right\|\right\} \text{ in the}$$

coupled case, and reject s_k.
Otherwise accept s_k and choose δ_{k+1} such that

$$\max\{\delta_{min}, \delta_k\} \leq \delta_{k+1} \leq \delta_{max}.$$

2.6 If s_k was rejected set $x_{k+1} = x_k$ and $\lambda_{k+1} = \lambda_k$. Otherwise set $x_{k+1} = x_k + s_k$ and $\lambda_{k+1} = \lambda_k + \Delta\lambda_k$.

A reasonable stopping criterion for global convergence to a stationary point is $\|\bar{D}_k\bar{g}_k\| + \|C_k\| \leq \epsilon_{tol}$ for a given $\epsilon_{tol} > 0$. If global convergence to a point satisfying the second–order necessary optimality conditions is the goal of the algorithms, then the stopping criterion should look like $\|\bar{D}_k\bar{g}_k\| + \|C_k\| + \gamma_k \leq \epsilon_{tol}$, where γ_k is the

Lagrange multiplier associated with the trust–region constraint in (5.18) (or (5.20)) that satisfies equation (5.23) (or (5.28)).

Once again we point out that the rules to update the trust radius in the previous algorithm can be much more involved to enhance efficiency, but the above suffices for our presentation.

5.2.5 General Assumptions

In order to establish local and global convergence results we need some general assumptions. We list these assumptions below. They extend for our problem (4.1) the assumptions in Chapter 3 for equality–constrained optimization. Let Ω be an open subset of $\mathbb{R}^n$ such that for all iterations k, x_k and $x_k + s_k$ are in Ω.

Assumptions 5.1–5.6

5.1 The same as Assumption 3.1.
(The functions f, c_i, $i = 1, \ldots, m$ are twice continuously differentiable functions in Ω.)

5.2 The partial Jacobian $C_y(x)$ is nonsingular for all $x \in \Omega$.
(This implies Assumption 3.2.)

5.3 The same as Assumption 3.3.
(The functions f, ∇f, $\nabla^2 f$, C, J, and $\nabla^2 c_i$, $i = 1, \ldots, m$, are bounded in Ω.)

5.4 The same as Assumption 3.4.
(The sequences $\{W_k\}$, $\{H_k\}$, and $\{\lambda_k\}$ are bounded.)

5.5 The matrix $C_y^{-1}(x)$ is uniformly bounded in Ω.
(This implies Assumption 3.5.)

5.6 The sequence $\{u_k\}$ is bounded.

It is not difficult to see that when the equality constraints of problem (3.1) reduce to the equality constraints of problem (4.1), Assumptions 5.1–5.5 given above imply Assumptions 3.1–3.5 given in Chapter 3. Assumption 5.6 is used by Coleman and Li [24] for optimization problems with simple bounds.

It is equivalent to Assumptions 5.3–5.6, that there exist positive constants $\nu_0, \ldots, \nu_9$ independent of k such that

$$|f(x)| \le \nu_0, \qquad \|\nabla f(x)\| \le \nu_1, \quad \|\nabla^2 f(x)\| \le \nu_2, \quad \|C(x)\| \le \nu_3, \qquad \|J(x)\| \le \nu_4,$$
$$\|\nabla^2 c_i(x)\| \le \nu_5, \quad i = 1, \ldots, m, \qquad \text{and} \qquad \|C_y(x)^{-1}\| \le \nu_6$$

for all $x \in \Omega$, and

$$\|W_k\| \le \nu_6, \quad \|H_k\| \le \nu_7, \quad \|\lambda_k\| \le \nu_8, \quad \text{and} \quad \|\bar{D}_k\| \le \nu_9,$$

for all k.

For the rest of this chapter we suppose that Assumptions 5.1–5.6 are always satisfied.

As we pointed out earlier, our approach is related to the Newton method presented in Section 5.1. The u component $(s_k^{\mathsf{N}})_u$ of the Newton step $s_k^{\mathsf{N}} = s_k^{\mathsf{q}} + W_k (s_k^{\mathsf{N}})_u$, whenever it is defined, is given by

$$\begin{aligned} (s_k^{\mathsf{N}})_u &= -\left(\bar{D}_k^2\, W_k^T H_k W_k + E_k\right)^{-1} \bar{D}_k^2 \bar{g}_k \\ &= -\bar{D}_k \left(\bar{D}_k\, W_k^T H_k W_k \bar{D}_k + E_k\right)^{-1} \bar{D}_k \bar{g}_k, \end{aligned} \tag{5.34}$$

where

$$s_k^{\mathsf{q}} = \begin{pmatrix} -C_y(x_k)^{-1} C_k \\ 0 \end{pmatrix} \tag{5.35}$$

and $\bar{g}_k = W_k^T \big(H_k s_k^{\mathsf{q}} + \nabla f_k\big)$. From (5.34) we see that that the Newton step is well defined in a neighborhood of a nondegenerate point that satisfies the second–order sufficient optimality conditions and for which $W_k^T H_k s_k^{\mathsf{q}}$ is sufficiently small. To guarantee strict feasibility of this step we consider a damped Newton step given by

$$s_k^{\mathsf{q}} + W_k \tau_k^{\mathsf{N}} (s_k^{\mathsf{N}})_u, \tag{5.36}$$

where $(s_k^{\mathsf{N}})_u$ and s_k^{q} are given by (5.34) and (5.35) respectively, and

$$\tau_k^{\mathsf{N}} = \sigma_k \min\left\{1,\ \max\left\{\frac{b_i - (u_k)_i}{((s_k^{\mathsf{N}})_u)_i}, \frac{a_i - (u_k)_i}{((s_k^{\mathsf{N}})_u)_i}\right\},\ i = 1, \ldots, n-m\right\}. \tag{5.37}$$

If Algorithms 5.2.1 are particularized to satisfy the following conditions on the steps, on the quadratic model, and on the Lagrange multipliers, then we can prove global and local convergence.

Conditions 5.1–5.4

5.1 The quasi–normal component s_k^{q} satisfies conditions (5.9), (5.12), and (5.13). The tangential component $(s_k)_u$ satisfies the fraction of Cauchy decrease condition (5.21) ((5.22) for the coupled approach).
The parameter σ_k is chosen in $[\sigma, 1)$, where $\sigma \in (0,1)$ is fixed for all k.

5.2 The tangential component $(s_k)_u$ satisfies the fraction of optimal decrease condition (5.26) ((5.30) for the coupled approach).

5.3 The second derivatives of f and c_i, $i = 1, \ldots, m$ are Lipschitz continuous in Ω. The approximation to the Hessian matrix is exact, i.e. $H_k = \nabla^2_{xx}\ell(x_k, \lambda_k)$ with Lagrange multiplier $\lambda_k = -C_y(x_k)^{-T}\nabla_y f_k$.

5.4 The step s_k is given by (5.36) provided $(s_k^{\mathsf{N}})_u$ exists, $(s_k^{\mathsf{q}})_y$ lies inside the trust region (5.11), and $\tau_k^{\mathsf{N}}(s_k^{\mathsf{N}})_u$ lies inside the trust region (5.18) ((5.20) for the coupled approach).
The parameter σ_k is chosen such that $\sigma_k \geq \sigma$ and $|\sigma_k - 1|$ is $\mathcal{O}\left(\left\|\bar{D}_k \bar{g}_k\right\|\right)$.

Condition 5.1 assures global convergence to a point satisfying the first–order necessary optimality conditions. Global convergence to a nondegenerate point that satisfies second–order necessary optimality conditions requires Conditions 5.1–5.3. To prove local q–quadratic convergence, we need Conditions 5.1, 5.3, and 5.4.

> **Remark 5.2.1** A very important point here is that there is no need to add to Conditions 5.1–5.3 the condition (3.19) on the quasi–normal component s_k^{q}. We recall that this latter condition was required in Chapter 3 to prove global convergence to a point satisfying the second–order necessary optimality conditions. In fact, given the form (5.9) of s_k^{q} imposed in Condition 5.1 and the adjoint update of λ_k described in Condition 5.3, we have $\nabla_x \ell_k^T s_k^{\mathsf{q}} = 0$ (see expression (5.53)).

5.3 Intermediate Results

We start by pointing out that, as in Section 3.5, (5.13) with the fact that the tangential component lies in the null space of J_k together imply that

$$\|C_k\|^2 - \|J_k s_k + C_k\|^2 \geq \kappa_2 \|C_k\| \min\{\kappa_3\|C_k\|, \delta_k\}. \tag{5.38}$$

We calculated the first derivatives of $\lambda(x) = -C_y(x)^{-T}\nabla_y f(x)$ in Section 5.1. It is clear that under Assumptions 5.3 and 5.5 these derivatives are bounded in Ω. Thus, if λ_k is computed as stated in Condition 5.3, then there exists a positive constant ν_{10} independent of k such that

$$\|\Delta\lambda_k\| \leq \nu_{10}\|s_k\|. \tag{5.39}$$

From $\|s_k^{\mathsf{q}}\| \leq \delta_{max}$ and Assumptions 5.3–5.4 we also have

$$\|\bar{g}_k\| = \|W_k^T \left(H_k s_k^{\mathsf{q}} + \nabla f_k\right)\| \leq \nu_{11}, \tag{5.40}$$

where $\nu_{11} = \nu_6(\nu_7\delta_{max} + \nu_1)$.

The following result is a direct consequence of the scheme that updates ρ_k in Step 2.4 of Algorithms 5.2.1. This result is exactly the same as in Lemma 3.5.1 and we just state it for completeness.

Lemma 5.3.1 The sequence $\{\rho_k\}$ satisfies

$$\begin{aligned} &\rho_k \geq \rho_{k-1} \geq 1 \quad \text{and} \\ &pred(s_k;\rho_k) \geq \frac{\rho_k}{2}\Big(\|C_k\|^2 - \|J_k s_k + C_k\|^2\Big). \end{aligned} \tag{5.41}$$

The following lemma relating the sizes of $\|s_k\|$ and δ_k is required also for the convergence theory.

Lemma 5.3.2 Let Assumptions 5.1–5.6 hold. Every step satisfies

$$\|s_k\| \leq \kappa_4\delta_k \tag{5.42}$$

and, if s_k is rejected in Step 2.5 of Algorithms 5.2.1, then

$$\delta_{k+1} \geq \kappa_5\|s_k\|, \tag{5.43}$$

where κ_4 and κ_5 are positive constants independent of k.

Proof In the coupled trust–region approach we bound s_k^{t} as follows:

$$\begin{aligned} \left\|\begin{pmatrix} -C_y(x_k)^{-1}C_u(x_k)s_u \\ s_u \end{pmatrix}\right\| &\leq \left\|\begin{pmatrix} I_m & 0 \\ 0 & \bar{D}_k \end{pmatrix}\right\| \left\|\begin{pmatrix} -C_y(x_k)^{-1}C_u(x_k)s_u \\ \bar{D}_k^{-1}s_u \end{pmatrix}\right\| \\ &\leq (1+\nu_9)\,\delta_k\,, \end{aligned}$$

where ν_9 is a uniform bound for $\|\bar{D}_k\|$, see Assumption 5.6. Since $\|s_k^{\mathsf{q}}\| \leq \delta_k$, we obtain $\|s_k\| \leq (2+\nu_9)\,\delta_k$. It is not difficult to see now that in Step 2.5 we have $\delta_{k+1} \geq \frac{\alpha_1}{2} \min\left\{1, \frac{1}{1+\nu_9}\right\} \|s_k\|$.

In the decoupled approach, $\|s_k\| = \|s_k^{\mathsf{q}} + W_k(s_k)_u\| \leq (1+\nu_6\nu_9)\delta_k$ and similarly $\delta_{k+1} \geq \frac{\alpha_1}{2} \min\left\{1, \frac{1}{\nu_6\nu_9}\right\} \|s_k\|$, where ν_6 is a uniform bound for $\|W_k\|$, see Assumption 5.4.

We can combine these bounds to obtain

$$\begin{aligned} \|s_k\| &\leq \max\{2+\nu_9, 1+\nu_6\nu_9\}\,\delta_k, \\ \delta_{k+1} &\geq \tfrac{\alpha_1}{2} \min\left\{1, \tfrac{1}{1+\nu_9}, \tfrac{1}{\nu_6\nu_9}\right\} \|s_k\|. \end{aligned}$$

In the case where fraction of optimal decrease (5.26) or (5.30) is imposed on $(s_k)_u$, the constants κ_4 and κ_5 depend also on β_3^{d} and β_3^{c}. □

In the following lemma we rewrite the fraction of Cauchy decrease conditions (5.21) and (5.22) in a more useful form for the analysis.

Lemma 5.3.3 Let Assumptions 5.1–5.6 hold. If $(s_k)_u$ satisfies Condition 5.1 then

$$q_k(s_k^{\mathsf{q}}) - q_k(s_k^{\mathsf{q}} + W_k(s_k)_u) \geq \kappa_6 \|\bar{D}_k\bar{g}_k\| \min\left\{\kappa_7\|\bar{D}_k\bar{g}_k\|, \kappa_8\delta_k\right\}, \qquad (5.44)$$

where κ_6, κ_7, and κ_8 are positive constants independent of the iteration k.

Proof From the definition (5.15) of Ψ_k we find

$$\begin{aligned} q_k(s_k^{\mathsf{q}}) - q_k(s_k^{\mathsf{q}} + W_k(s_k)_u) &\geq q_k(s_k^{\mathsf{q}}) - q_k(s_k^{\mathsf{q}} + W_k(s_k)_u) - \frac{1}{2}(s_k)_u^T\left(E_k\bar{D}_k^{-2}\right)(s_k)_u \\ &= \Psi_k(0) - \Psi_k((s_k)_u). \end{aligned} \qquad (5.45)$$

Let $\tilde{\delta}_k$ be the maximum $\|\bar{D}_k^{-1}\ \cdot\|$ norm of a step, say $(\tilde{s}_k)_u$, along $-\bar{D}_k\frac{\hat{g}_k}{\|\hat{g}_k\|}$ allowed inside the trust region. Here $\hat{g}_k = \bar{D}_k\bar{g}_k$.

If the trust region is given by (5.18), then

$$\delta_k = \tilde{\delta}_k. \qquad (5.46)$$

If the trust region is given by (5.20), then we can use Assumptions 5.4–5.6 to deduce the inequality

$$\begin{aligned}\delta_k^2 = \left\| \begin{pmatrix} -C_y(x_k)^{-1}C_u(x_k)(\tilde{s}_k)_u \\ \bar{D}_k^{-1}(\tilde{s}_k)_u \end{pmatrix} \right\|^2 &= \| - C_y(x_k)^{-1}C_u(x_k)\bar{D}_k\bar{D}_k^{-1}(\tilde{s}_k)_u\|^2 \\ &\quad + \|\bar{D}_k^{-1}(\tilde{s}_k)_u\|^2 \\ &\le (\nu_6^2\nu_9^2+1)\|\bar{D}_k^{-1}(\tilde{s}_k)_u\|^2 \\ &= (\nu_6^2\nu_9^2+1)\,\tilde{\delta}_k^2,\end{aligned}$$

or, equivalently,

$$\tilde{\delta}_k \ge \frac{1}{\sqrt{\nu_6^2\nu_9^2+1}}\,\delta_k. \tag{5.47}$$

Define $\psi : \mathbb{R}^+ \longrightarrow \mathbb{R}$ as $\psi(t) = \Psi_k\left(-t\bar{D}_k\frac{\hat{g}_k}{\|\hat{g}_k\|}\right) - \Psi_k(0)$. Then $\psi(t) = -\|\hat{g}_k\|t + \frac{r_k}{2}t^2$, where $r_k = \frac{\hat{g}_k^T\widehat{H}_k\hat{g}_k}{\|\hat{g}_k\|^2}$ and $\widehat{H}_k = \bar{D}_k\left(W_k^TH_kW_k + E_k\bar{D}_k^{-2}\right)\bar{D}_k$. Now we need to minimize ψ in $[0, T_k]$ where T_k is given by

$$T_k = \min\left\{\tilde{\delta}_k,\ \sigma_k\min\left\{\frac{\|\bar{D}_k\bar{g}_k\|}{(\bar{g}_k)_i} : (\bar{g}_k)_i > 0\right\},\ \sigma_k\min\left\{-\frac{\|\bar{D}_k\bar{g}_k\|}{(\bar{g}_k)_i} : (\bar{g}_k)_i < 0\right\}\right\}.$$

Let t_k^* be the minimizer of ψ in $[0, T_k]$. As in the proof of Lemma 2.3.1 (see equations (2.8) and (2.9)), it easily can be proved that

$$\psi(t_k^*) \le -\frac{1}{2}\|\hat{g}_k\|\min\left\{\frac{\|\hat{g}_k\|}{\|\widehat{H}_k\|},\ T_k\right\}. \tag{5.48}$$

We can combine (5.45) and (5.48) with

$$\Psi_k(0) - \Psi_k((s_k)_u) \ge \beta_1^{\mathsf{d}}\left(\Psi_k(0) - \Psi_k(c_k^{\mathsf{d}})\right) = -\beta_1^{\mathsf{d}}\psi(t_k^*)$$

to get

$$q_k(s_k^{\mathsf{q}}) - q_k(s_k^{\mathsf{q}} + W_k(s_k)_u) \ge \frac{1}{2}\beta_1^{\mathsf{d}}\|\hat{g}_k\|\min\left\{\frac{\|\hat{g}_k\|}{\|\widehat{H}_k\|}, T_k\right\}.$$

The facts that $\sigma_k \ge \sigma$ and $\|\bar{g}_k\| \le \nu_{11}$ (see (5.40)) imply that

$$\begin{aligned}&\Psi_k(0) - \Psi_k((s_k)_u) \\ &\quad\ge \frac{1}{2}\beta_1^{\mathsf{d}}\|\bar{D}_k\bar{g}_k\|\min\left\{\frac{\|\bar{D}_k\bar{g}_k\|}{\|\bar{D}_k^T\left(W_k^TH_kW_k + E_k\bar{D}_k^{-2}\right)\bar{D}_k\|}, \min\left\{\tilde{\delta}_k, \frac{\sigma}{\nu_{11}}\|\bar{D}_k\bar{g}_k\|\right\}\right\}.\end{aligned}$$

To complete the proof, we use (5.46), (5.47), Assumptions 5.1–5.6, and $\delta_k \leq \delta_{max}$ to establish (5.44) with $\kappa_6 = \frac{1}{2}\min\{\beta_1^{\mathsf{d}}, \beta_1^{\mathsf{c}}\}$, $\kappa_7 = \min\left\{\frac{1}{\nu_6^2\nu_7\nu_9^2+\nu_1\nu_6}, \frac{\sigma}{\nu_{11}}\right\}$, and $\kappa_8 = \min\left\{1, \frac{1}{\sqrt{\nu_6^2\nu_9^2+1}}\right\}$. □

Now we state the convenient form of the fraction of optimal decrease conditions (5.26) and (5.30).

Lemma 5.3.4 Let Assumptions 5.1–5.6 hold. If $(s_k)_u$ satisfies Condition 5.2 then

$$q_k(s_k^{\mathsf{q}}) - q_k(s_k^{\mathsf{q}} + W_k(s_k)_u) \geq \kappa_9 \tau_k^2 \gamma_k \delta_k^2, \tag{5.49}$$

where κ_9 is a positive constant independent of the iteration k.

Proof The proof follows immediately from observation (5.45) and conditions (5.27) and (5.31). □

We also need the following two inequalities. (See Lemma 3.5.3 for a similar result.)

Lemma 5.3.5 Let Assumptions 5.1–5.6 hold. Under Condition 5.1 there exists a positive constant κ_{10} such that

$$q_k(0) - q_k(s_k^{\mathsf{q}}) - \Delta\lambda_k^T(J_k s_k + C_k) \geq -\kappa_{10}\|C_k\|. \tag{5.50}$$

Moreover, if we assume Condition 5.3, then

$$q_k(0) - q_k(s_k^{\mathsf{q}}) - \Delta\lambda_k^T(J_k s_k + C_k) \geq -\kappa_{11}\|C_k\|\left(\|s_k^{\mathsf{q}}\| + \|s_k\|\right). \tag{5.51}$$

Proof The term $q_k(0) - q_k(s_k^{\mathsf{q}})$ can be bounded using (5.12) and $\|s_k^{\mathsf{q}}\| \leq \delta_k$ in the following way:

$$\begin{aligned} q_k(0) - q_k(s_k^{\mathsf{q}}) &= -\nabla_x \ell_k^T s_k^{\mathsf{q}} - \tfrac{1}{2}(s_k^{\mathsf{q}})^T H_k (s_k^{\mathsf{q}}) \\ &\geq -\kappa_1\left(\|\nabla_x \ell_k\| + \tfrac{1}{2}\delta_k\|H_k\|\right)\|C_k\|. \end{aligned}$$

On the other hand, it follows from $\|J_k s_k + C_k\| \leq \|C_k\|$ that

$$-\Delta\lambda_k^T(J_k s_k + C_k) \geq -\|\Delta\lambda_k\|\,\|C_k\|. \tag{5.52}$$

Combining these two bounds with Assumptions 5.3–5.4 we get (5.50).

To prove (5.51) we first observe that, due to the definition of λ_k in Condition 5.3 and to the form (5.9) of the quasi–normal component s_k^{q},

$$\nabla_x \ell_k^T s_k^{\mathsf{q}} = \left(\begin{array}{c} 0 \\ \nabla_u f_k + C_u(x_k)^T \lambda_k \end{array} \right)^T \left(\begin{array}{c} (s_k^{\mathsf{q}})_y \\ 0 \end{array} \right) = 0. \tag{5.53}$$

Thus

$$q_k(0) - q_k(s_k^{\mathsf{q}}) \geq -\frac{1}{2}\kappa_1 \|H_k\| \, \|C_k\| \, \|s_k^{\mathsf{q}}\| \geq -\frac{1}{2}\kappa_1 \nu_7 \, \|C_k\| \, \|s_k^{\mathsf{q}}\|. \tag{5.54}$$

Also, by appealing to (5.39) and (5.52),

$$-\Delta\lambda_k^T (J_k s_k + C_k) \geq -\nu_{10} \|s_k\| \, \|C_k\|. \tag{5.55}$$

The proof of (5.51) is complete by combining (5.54) and (5.55). □

The convergence theory for trust–region algorithms traditionally requires consistency of actual and predicted decreases. This is given in the following lemma.

Lemma 5.3.6 Let Assumptions 5.1–5.6 hold. Under Condition 5.1 there exists a positive constant κ_{12} such that

$$|ared(s_k; \rho_k) - pred(s_k; \rho_k)| \leq \kappa_{12} \left(\|s_k\|^2 + \rho_k \left(\|s_k\|^3 + \|C_k\| \, \|s_k\|^2 \right) \right). \tag{5.56}$$

Moreover, assume also Condition 5.3, and then

$$|ared(s_k; \rho_k) - pred(s_k; \rho_k)| \leq \kappa_{13} \rho_k \left(\|s_k\|^3 + \|C_k\| \, \|s_k\|^2 \right). \tag{5.57}$$

Proof The bulk of the proof is the same as the proof of Lemma 3.5.4. The estimate (5.56) is a direct consequence of (3.37) and of the boundedness of $\{\|\Delta\lambda_k\|\}$ (see Assumption 5.4) whereas estimate (5.57) comes from (3.38) and inequality (5.39). □

5.4 Global Convergence to a First–Order Point

The proof of global convergence to a point satisfying the first–order necessary optimality conditions (Theorem 5.4.1) established in this section follows the structure of the convergence theory presented in [35] for the equality–constrained optimization problem. This proof is by contradiction and is based on Condition 5.1. We show that the supposition

$$\|\bar{D}_k\bar{g}_k\| + \|C_k\| > \epsilon_{tol},$$

for all k, leads to a contradiction.

The following three lemmas are necessary to bound the predicted decrease.

Lemma 5.4.1 Let Assumptions 5.1–5.6 hold. Under Condition 5.1 the predicted decrease in the merit function satisfies

$$\begin{aligned} pred(s_k;\rho) \ \geq\ & \kappa_6\|\bar{D}_k\bar{g}_k\| \min\left\{\kappa_7\|\bar{D}_k\bar{g}_k\|, \kappa_8\delta_k\right\} \\ & -\kappa_{10}\|C_k\| + \rho(\|C_k\|^2 - \|J_k s_k + C_k\|^2), \end{aligned} \tag{5.58}$$

for every $\rho > 0$.

Proof The inequality (5.58) follows from a direct application of (5.50) and from the lower bound (5.44). □

Lemma 5.4.2 Let Assumptions 5.1–5.6 hold. Assume that Condition 5.1 and $\|\bar{D}_k\bar{g}_k\| + \|C_k\| > \epsilon_{tol}$ are satisfied. If $\|C_k\| \leq \theta\delta_k$, where θ is a positive constant satisfying

$$\theta \leq \min\left\{\frac{\epsilon_{tol}}{3\delta_{max}}, \frac{\kappa_6\epsilon_{tol}}{3\kappa_{10}} \min\left\{\frac{2\kappa_7\epsilon_{tol}}{3\delta_{max}}, \kappa_8\right\}\right\}, \tag{5.59}$$

then

$$\begin{aligned} pred(s_k;\rho) \ \geq\ & \frac{\kappa_6}{2}\|\bar{D}_k\bar{g}_k\| \min\left\{\kappa_7\|\bar{D}_k\bar{g}_k\|, \kappa_8\delta_k\right\} \\ & + \rho\left(\|C_k\|^2 - \|J_k s_k + C_k\|^2\right), \end{aligned} \tag{5.60}$$

for every $\rho > 0$.

Proof From $\|\bar{D}_k\bar{g}_k\| + \|C_k\| > \epsilon_{tol}$ and the first bound on θ given by (5.59), we get

$$\|\bar{D}_k\bar{g}_k\| > \frac{2}{3}\epsilon_{tol}. \tag{5.61}$$

If we use this, (5.58), and the second bound on θ given by (5.59), we obtain

$$\begin{aligned} pred(s_k;\rho) \ \geq\ & \tfrac{\kappa_6}{2}\|\bar{D}_k\bar{g}_k\| \min\left\{\kappa_7\|\bar{D}_k\bar{g}_k\|, \kappa_8\delta_k\right\} + \tfrac{\kappa_6\epsilon_{tol}}{3} \min\{\tfrac{2\kappa_7\epsilon_{tol}}{3}, \kappa_8\delta_k\} \\ & -\kappa_{10}\|C_k\| + \rho\Big(\|C_k\|^2 - \|J_k s_k + C_k\|^2\Big) \\ \geq\ & \tfrac{\kappa_6}{2}\|\bar{D}_k\bar{g}_k\| \min\left\{\kappa_7\|\bar{D}_k\bar{g}_k\|, \kappa_8\delta_k\right\} + \rho\Big(\|C_k\|^2 - \|J_k s_k + C_k\|^2\Big). \end{aligned}$$

□

We can use Lemma 5.4.2 with $\rho = \rho_{k-1}$ and conclude that if $\|\bar{D}_k\bar{g}_k\| + \|C_k\| > \epsilon_{tol}$ and $\|C_k\| \leq \theta\delta_k$, then the penalty parameter at the current iteration does not need to be increased. See Step 2.4 of Algorithms 5.2.1.

Lemma 5.4.3 Let Assumptions 5.1–5.6 hold. Assume that Condition 5.1 and $\|\bar{D}_k\bar{g}_k\| + \|C_k\| > \epsilon_{tol}$ are satisfied. If $\|C_k\| \leq \theta\delta_k$, where θ satisfies (5.59), then there exists a positive constant $\kappa_{14} > 0$ such that

$$pred(s_k; \rho_k) \geq \kappa_{14}\delta_k. \tag{5.62}$$

Proof From (5.60) with $\rho = \rho_k$ and $\|\bar{D}_k\bar{g}_k\| \geq \frac{2}{3}\epsilon_{tol}$, cf. (5.61), we obtain

$$\begin{aligned} pred(s_k; \rho_k) &\geq \tfrac{\kappa_6\epsilon_{tol}}{3} \min\{\tfrac{2\kappa_7\epsilon_{tol}}{3}, \kappa_8\delta_k\} \\ &\geq \tfrac{\kappa_6\epsilon_{tol}}{3} \min\{\tfrac{2\kappa_7\epsilon_{tol}}{3\delta_{max}}, \kappa_8\}\delta_k. \end{aligned}$$

Hence (5.62) holds with

$$\kappa_{14} = \frac{\kappa_6\epsilon_{tol}}{3} \min\left\{\frac{2\kappa_7\epsilon_{tol}}{3\delta_{max}}, \kappa_8\right\}.$$

□

The following lemma is also required.

Lemma 5.4.4 Let Assumptions 5.1–5.6 hold. Under Condition 5.1, if $\|\bar{D}_k\bar{g}_k\| + \|C_k\| > \epsilon_{tol}$ for all k then the sequences $\{\rho_k\}$ and $\{L_k\}$ are bounded and δ_k is uniformly bounded away from zero.

Proof See Lemmas 7.9–7.13, 8.2 in [35]. □

Our first global convergence result follows.

Theorem 5.4.1 Under Assumptions 5.1–5.6 and Condition 5.1 the sequences of iterates generated by the TRIP Reduced SQP Algorithms 5.2.1 satisfy

$$\liminf_{k\to+\infty} \left(\|D_k W_k^T \nabla f_k\| + \|C_k\|\right) = 0. \tag{5.63}$$

Proof The proof is by contradiction. Suppose that for all k

$$\|\bar{D}_k \bar{g}_k\| + \|C_k\| > \epsilon_{tol}. \tag{5.64}$$

At each iteration k either $\|C_k\| \leq \theta\delta_k$ or $\|C_k\| > \theta\delta_k$, where θ satisfies (5.59). In the first case, we appeal to Lemmas 5.4.3 and 5.4.4 and obtain

$$pred(s_k;\rho_k) \geq \kappa_{14}\delta_*,$$

where δ_* is the lower bound on δ_k given by Lemma 5.4.4. If $\|C_k\| > \theta\delta_k$, we have from $\rho_k \geq 1$, (5.38), (5.41), and Lemma 5.4.4, that

$$pred(s_k;\rho_k) \geq \frac{\kappa_2}{2}\theta \min\{\kappa_3\theta, 1\}\delta_*.$$

Hence $pred(s_k;\rho_k) \geq \kappa_{15}$ for all k, where the positive constant κ_{15} does not depend on k. From this and (5.56) we establish

$$\left|\frac{ared(s_k;\rho_k) - pred(s_k;\rho_k)}{pred(s_k;\rho_k)}\right| \leq \frac{\kappa_{12}}{\kappa_{15}}\left(\|s_k\|^2 + \rho_*\left(\|s_k\|^3 + \|C_k\|\,\|s_k\|^2\right)\right) \leq \kappa_{16}\delta_k^2,$$

where ρ_* is the upper bound on ρ_k guaranteed by Lemma 5.4.4. From the rules that update δ_k in Step 2.5 of Algorithms 5.2.1 this inequality tells us that an acceptable step always is found after a finite number of unsuccessful iterations. Using this fact, we can ignore the rejected steps and work only with successful iterates. So, without loss of generality, we have

$$L_k - L_{k+1} = ared(s_k;\rho_k) \geq \eta_1 pred(s_k;\rho_k) \geq \eta_1\kappa_{15}.$$

Now, if we let k go to infinity, this contradicts the boundedness of $\{L_k\}$ guaranteed by Lemma 5.4.4. Hence the supposition (5.64) is false, and we must have that

$$\liminf_{k\to+\infty}\left(\|\bar{D}_k\bar{g}_k\| + \|C_k\|\right) = 0. \tag{5.65}$$

To establish the desired result, we note that $D(x)W(x)^T\nabla f(x)$ is a continuous function of x. For a given bounded $H(x,\lambda)$, let us consider $\bar{D}(x)W(x)^T\Big(H(x,\lambda)s^{\mathsf{q}} + \nabla f(x)\Big)$, where $\bar{D}(x)$ is defined with the reduced gradient $W(x)^T\Big(H(x,\lambda)s^{\mathsf{q}} + \nabla f(x)\Big)$. It is then clear that $\bar{D}(x)W(x)^T\Big(H(x,\lambda)s^{\mathsf{q}} + \nabla f(x)\Big)$ is a continuous function on the pair (x, s^{q}) at $(x, 0)$. From this observation, and since $\|s_k^{\mathsf{q}}\| \leq \kappa_1\|C_k\|$ and $\liminf_{k\to+\infty}\|C_k\| = 0$, we see that the limit (5.65) implies the limit (5.63). $\square$

If $\{x_k\}$ is a bounded sequence we conclude from Theorem 5.4.1 and the continuity of $C(x)$ and $D(x)W(x)^T\nabla f(x)$, that $\{x_k\}$ has a limit point satisfying the first–order necessary optimality conditions.

5.5 Global Convergence to a Second–Order Point

In this section we establish global convergence to a point that satisfies the second–order necessary optimality conditions.

Theorem 5.5.1 Under Assumptions 5.1–5.6 and Conditions 5.1–5.3, the sequences of iterates generated by the TRIP Reduced SQP Algorithms 5.2.1 satisfy

$$\liminf_{k \to +\infty} \left(\|\bar{D}_k \bar{g}_k\| + \|C_k\| + \tau_k^2 \gamma_k \right) = 0, \tag{5.66}$$

where γ_k is the Lagrange multiplier corresponding to the trust–region constraint, see (5.23) and (5.28), and τ_k is the damping parameter defined in (5.25).

Proof The proof is again by contradiction. Suppose that for all k,

$$\|\bar{D}_k \bar{g}_k\| + \|C_k\| + \tau_k^2 \gamma_k > \frac{5}{3} \epsilon_{tol}. \tag{5.67}$$

(i) Suppose that $\|C_k\| \le \theta' \delta_k$, where

$$\theta' = \min \left\{ \theta, \frac{\kappa_9 \epsilon_{tol}}{3\kappa_{11}(1+\kappa_4)} \right\} \tag{5.68}$$

and θ satisfies (5.59). From the first bound on θ in (5.59) we get

$$\|\bar{D}_k \bar{g}_k\| + \tau_k^2 \gamma_k > \frac{4}{3} \epsilon_{tol}.$$

Thus, either $\|\bar{D}_k \bar{g}_k\| > \frac{2}{3}\epsilon_{tol}$ or $\tau_k^2 \gamma_k > \frac{2}{3}\epsilon_{tol}$. In the first case, we proceed exactly as in Lemmas 5.4.2 and 5.4.3 and obtain

$$\begin{aligned} pred(s_k; \rho) &\ge \frac{\kappa_6}{2} \|\bar{D}_k \bar{g}_k\| \min \left\{ \kappa_7 \|\bar{D}_k \bar{g}_k\|, \kappa_8 \delta_k \right\} \\ &\quad + \rho \left(\|C_k\|^2 - \|J_k s_k + C_k\|^2 \right) \\ &\ge \frac{\kappa_{14}}{\delta_{max}} \delta_k^2 \end{aligned} \tag{5.69}$$

for any $\rho > 0$. If $\tau_k^2 \gamma_k > \frac{2}{3}\epsilon_{tol}$ then from (5.42), (5.49), (5.51), $\|s_k^{\mathsf{q}}\| \le \delta_k$, and the second bound on θ' given in (5.68), we can write

$$pred(s_k; \rho) = q_k(s_k^{\mathsf{q}}) - q_k(s_k^{\mathsf{q}} + W_k (s_k)_u) + q_k(0) - q_k(s_k^{\mathsf{q}}) - \Delta\lambda_k^T (J_k s_k + C_k)$$

$$
\begin{aligned}
&\quad + \rho\left(\|C_k\|^2 - \|J_k s_k + C_k\|^2\right) \\
&\geq \frac{1}{2}\kappa_9 \tau_k^2 \gamma_k \delta_k^2 + \left(\frac{1}{3}\kappa_9 \epsilon_{tol}\delta_k - \kappa_{11}\|C_k\|(1+\kappa_4)\right)\delta_k \\
&\quad + \rho\left(\|C_k\|^2 - \|J_k s_k + C_k\|^2\right) \\
&\geq \frac{1}{2}\kappa_9 \tau_k^2 \gamma_k \delta_k^2 + \rho\left(\|C_k\|^2 - \|J_k s_k + C_k\|^2\right) \qquad (5.70)\\
&\geq \frac{\kappa_9 \epsilon_{tol}}{3}\delta_k^2
\end{aligned}
$$

for any $\rho > 0$. From the two bounds (5.69) and (5.70) we conclude that if $\|C_k\| \leq \theta' \delta_k$, then the penalty parameter does not increase. See Step 2.4 of Algorithms 5.2.1. Moreover, these two bounds on $pred(s_k; \rho_k)$ show the existence of a positive constant κ_{17} independent of k such that

$$
pred(s_k; \rho_k) \geq \kappa_{17}\delta_k^2, \qquad (5.71)
$$

provided $\|C_k\| \leq \theta' \delta_k$.

(ii) Now we prove that $\{\rho_k\}$ is bounded. If ρ_k is increased at iteration k, then it is updated according to the rule

$$
\rho_k = 2\left(\frac{q_k(s_k) - q_k(0) + \Delta\lambda_k^T(J_k s_k + C_k)}{\|C_k\|^2 - \|J_k s_k + C_k\|^2}\right) + \bar{\rho}.
$$

We can write

$$
\begin{aligned}
\tfrac{\rho_k}{2}\left(\|C_k\|^2 - \|J_k s_k + C_k\|^2\right) &= q_k(s_k) - q_k(s_k^{\mathsf{q}}) \\
&\quad -\left(q_k(0) - q_k(s_k^{\mathsf{q}})\right) + \Delta\lambda_k^T(J_k s_k + C_k) \\
&\quad +\tfrac{\bar{\rho}}{2}\left(\|C_k\|^2 - \|J_k s_k + C_k\|^2\right).
\end{aligned}
$$

By applying (5.38) to the left hand side and (5.42), (5.49), (5.51), and $\|s_k^{\mathsf{q}}\| \leq \delta_k$ to the right hand side, we obtain

$$
\begin{aligned}
\frac{\rho_k}{2}\kappa_2\|C_k\|\min\{\kappa_3\|C_k\|, \delta_k\} &\leq \kappa_{11}(1+\kappa_4)\delta_k\|C_k\| + \frac{\bar{\rho}}{2}\left(-2(J_k^T C_k)^T s_k - \|J_k s_k\|^2\right) \\
&\leq (\kappa_{11}(1+\kappa_4) + \bar{\rho}\nu_4\kappa_4)\delta_k\|C_k\|. \qquad (5.72)
\end{aligned}
$$

If ρ_k is increased at iteration k, then, because of part (i), $\|C_k\| > \theta'\delta_k$. Now we use this fact to establish that

$$
\left(\frac{\kappa_2}{2}\min\{\kappa_3\theta', 1\}\right)\rho_k \leq \kappa_{11}(1+\kappa_4) + \bar{\rho}\nu_4\kappa_4.
$$

This and Assumptions 5.1–5.6 prove that $\{\rho_k\}$ and $\{L_k\}$ are bounded sequences.

(iii) The next step is to prove that δ_k is bounded away from zero.

If s_{k-1} was an acceptable step, then $\delta_k \geq \delta_{min}$, see Step 2.5 in Algorithm 5.2.1.

If s_{k-1} was rejected, then $\delta_k \geq \kappa_5 \|s_{k-1}\|$, see (5.43). We consider two cases. In both cases we use the fact that

$$1 - \eta_1 \leq \left| \frac{ared(s_{k-1};\rho_{k-1})}{pred(s_{k-1};\rho_{k-1})} - 1 \right|.$$

In the first case, we assume that $\|C_{k-1}\| \leq \theta' \delta_{k-1}$. From (5.71) we have

$$pred(s_{k-1};\rho_{k-1}) \geq \kappa_{17} \delta_{k-1}^2.$$

Thus we can use $\|s_{k-1}\| \leq \kappa_4 \delta_{k-1}$, see (5.42), and (5.57) with $k = k-1$ to obtain

$$\left| \frac{ared(s_{k-1};\rho_{k-1})}{pred(s_{k-1};\rho_{k-1})} - 1 \right| \leq \frac{\kappa_{13}\rho_* \left(\kappa_4^2 \delta_{k-1}^2 + \theta' \kappa_4 \delta_{k-1}^2\right)}{\kappa_{17}\delta_{k-1}^2} \|s_{k-1}\|.$$

This gives $\delta_k \geq \kappa_5 \|s_{k-1}\| \geq \frac{\kappa_5(1-\eta_1)\kappa_{17}}{\kappa_{13}\rho_*(\kappa_4^2 + \theta'\kappa_4)} \equiv \kappa_{18}$.

The other case is $\|C_{k-1}\| > \theta' \delta_{k-1}$. In this case we get from (5.38) and (5.41) with $k = k-1$ that

$$\begin{aligned} pred(s_{k-1};\rho_{k-1}) &\geq \frac{\rho_{k-1}}{2}\kappa_2 \|C_{k-1}\| \min\{\kappa_3 \|C_{k-1}\|, \delta_{k-1}\} \\ &\geq \rho_{k-1}\kappa_{19}\delta_{k-1}\|C_{k-1}\| \\ &\geq \rho_{k-1}\theta'\kappa_{19}\delta_{k-1}^2, \end{aligned}$$

where $\kappa_{19} = \frac{\kappa_2}{2}\min\{\kappa_3\theta', 1\}$. Again we use $\rho_{k-1} \geq 1$ and (5.42) and (5.57) with $k = k-1$, this time with the last two lower bounds on $pred(s_{k-1};\rho_{k-1})$, and we write

$$\begin{aligned} \left| \frac{ared(s_{k-1};\rho_{k-1})}{pred(s_{k-1};\rho_{k-1})} - 1 \right| &\leq \frac{\kappa_{13}\rho_{k-1}\|s_{k-1}\|^3}{|pred(s_{k-1};\rho_{k-1})|} + \frac{\kappa_{13}\rho_{k-1}\|C_{k-1}\|\,\|s_{k-1}\|^2}{|pred(s_{k-1};\rho_{k-1})|} \\ &\leq \left(\frac{\kappa_{13}\rho_{k-1}\kappa_4^2\delta_{k-1}^2}{\rho_{k-1}\theta'\kappa_{19}\delta_{k-1}^2} + \frac{\kappa_{13}\rho_{k-1}\kappa_4\delta_{k-1}\|C_{k-1}\|}{\rho_{k-1}\kappa_{19}\delta_{k-1}\|C_{k-1}\|} \right) \|s_{k-1}\|. \end{aligned}$$

Hence $\delta_k \geq \kappa_5 \|s_{k-1}\| \geq \frac{\kappa_5(1-\eta_1)\theta'\kappa_{19}}{\kappa_{13}(\kappa_4^2+\kappa_4\theta')} \equiv \kappa_{20}$.

Combining the two cases yields

$$\delta_k \geq \delta_* = \min\{\delta_{min}, \kappa_{18}, \kappa_{20}\}$$

for all k.

(iv) The rest of the proof consists of proving that an acceptable step always is found after a finite number of iterations and then from this concluding that the supposition (5.67) is false. The proof of these facts is exactly the proof of Theorem 5.4.1 where θ is now θ' and $\kappa_{14}\delta_*$ is replaced by $\kappa_{17}\delta_*^2$. □

It is worthwhile to compare the limit (5.66) given by this theorem with the limit (3.42) given in Theorem 3.6.2 for equality–constrained optimization. In the former, we have $\liminf_{k\to+\infty} \tau_k^2\gamma_k = 0$ whereas in the latter we just have $\liminf_{k\to+\infty} \gamma_k = 0$. The presence of τ_k in (5.66) is due to the presence of the bound constraints on the variables u. One other difference is that in (5.66) the reduced gradient is scaled by the matrix D_k reflecting the first–order necessary optimality conditions.

The following result finally establishes global convergence to a nondegenerate point satisfying the second–order necessary optimality conditions. If no equality constraints are considered, the proof reduces to the proof of Lemma 3.8 of Coleman and Li [24].

> **Theorem 5.5.2** Let $\{x_k\}$ be a bounded sequence of iterates generated by the TRIP Reduced SQP Algorithms 5.2.1 under Assumptions 5.1–5.6 and Conditions 5.1–5.3. Then $\{x_k\}$ has a limit point x_* satisfying the first–order necessary optimality conditions. Furthermore, if x_* is nondegenerate, then x_* satisfies the second–order necessary optimality conditions.

Proof Consider the subsequence of $\{x_k\}$ for which the limit in (5.66) is zero. Since this subsequence is bounded we can use the same arguments as in the proof of Theorem 5.4.1 to show that it has a convergent subsequence indexed by $\{k_j\}$ such that

$$\lim_{j\to+\infty} \|\bar{D}_{k_j}\bar{g}_{k_j}\| + \|C_{k_j}\| = \lim_{j\to+\infty} \|D_{k_j} W_{k_j}^T \nabla f_{k_j}\| + \|C_{k_j}\| = 0. \tag{5.73}$$

Moreover,

$$\lim_{j\to+\infty} \tau_{k_j}^2 \gamma_{k_j} = 0, \tag{5.74}$$

where τ_{k_j} is given by (5.25). Let x_* denote the limit of $\{x_{k_j}\}$. It follows from (5.73) and the continuity of $C(x)$ and $D(x)W(x)^T\nabla f(x)$ that x_* satisfies the first–order necessary optimality conditions.

Now we assume that x_* is nondegenerate, and we prove that $\lim_{j\to+\infty} \gamma_{k_j} = 0$. First we consider the decoupled trust–region approach.

From (5.12), Assumptions 5.3–5.4, and the limit $\lim_{j\to+\infty} \|C_{k_j}\| = 0$, we get the limit

$$\lim_{j\to+\infty} \|W_{k_j}^T H_{k_j} s_{k_j}^{\mathsf{q}}\| = 0.$$

Since x_* is nondegenerate and $\lim_{j\to+\infty} \|W_{k_j}^T H_{k_j} s_{k_j}^{\mathsf{q}}\| = 0$, there exists $\epsilon_0 \in (0,1)$ such that

$$\min\left\{(u_{k_j})_i - a_i,\ b_i - (u_{k_j})_i\right\} + \left|\left(\bar{g}_{k_j}\right)_i\right| > 2\epsilon_0, \qquad i = 1, \ldots, n-m \tag{5.75}$$

for large enough j, and

$$2\epsilon_0 < \min\{b_i - a_i,\ i = 1, \ldots, n-m\}.$$

Without loss of generality, we only consider the cases where $\tau_{k_j} \le \sigma_{k_j} < 1$. In the following the index i is the index defining τ_{k_j} in (5.25). (The index i is really i_j but we drop the j from i_j to simplify the notation.) We also assume that j is large enough such that

$$\left|\left(\bar{D}_{k_j}^2 \bar{g}_{k_j}\right)_i\right| < \epsilon_0^2. \tag{5.76}$$

Multiplying both sides of (5.24) by $\bar{D}_{k_j}^2$ gives

$$\left(E_{k_j} + \gamma_{k_j} I_{n-m}\right) o_{k_j}^{\mathsf{d}} = \bar{D}_{k_j}^2 \left(-\bar{g}_{k_j} - W_{k_j}^T H_{k_j} W_{k_j} o_{k_j}^{\mathsf{d}}\right),$$

which in turn yields

$$\gamma_{k_j} |(o_{k_j}^{\mathsf{d}})_i| \le (\bar{D}_{k_j}^2)_{ii} \left|\left(-\bar{g}_{k_j} - W_{k_j}^T H_{k_j} W_{k_j} o_{k_j}^{\mathsf{d}}\right)_i\right|. \tag{5.77}$$

Also, Assumption 5.6 implies $\|o_{k_j}^{\mathsf{d}}\| \le \nu_9 \delta_{k_j} \le \nu_9 \delta_{max}$. From this, (5.40), and Assumptions 5.3–5.4, we can write

$$\frac{1}{(o_{k_j}^{\mathsf{d}})_i} \ge \frac{\gamma_{k_j}}{\kappa_{21} (\bar{D}_{k_j})_{ii}^2} \tag{5.78}$$

for some κ_{21} independent of k. Now we distinguish between two cases.

In the first case, we consider $\left|\left(\bar{g}_{k_j}\right)_i\right| \le \epsilon_0$ and appeal to (5.75) to get $\min\{(u_{k_j})_i - a_i,\ b_i - (u_{k_j})_i\} > \epsilon_0$. Thus from (5.78) and the definition (5.25) of τ_{k_j} we obtain

$$\tau_{k_j} \ge \frac{\sigma_{k_j} \gamma_{k_j} \epsilon_0}{\kappa_{21} (\bar{D}_{k_j})_{ii}^2}. \tag{5.79}$$

Now we analyze the case $\left|\left(\bar{g}_{k_j}\right)_i\right| > \epsilon_0$. Two possibilities can occur.

(i) The first possibility is that the value of the numerator defining τ_{k_j} in (5.25) is equal to $(\bar{D}_{k_j})^2_{ii}$. In this situation (5.78) immediately implies

$$\tau_{k_j} \geq \frac{\sigma_{k_j}\gamma_{k_j}}{\kappa_{21}}. \tag{5.80}$$

(ii) The other possibility is that the value of the numerator defining τ_{k_j} is not equal to $(\bar{D}_{k_j})^2_{ii}$. In this case we have from (5.76) that $(\bar{D}_{k_j})^2_{ii} < \epsilon_0$ and since $b_i - a_i > 2\epsilon_0$, the numerator in the definition (5.25) of τ_{k_j} is bigger than ϵ_0. Thus

$$\tau_{k_j} \geq \frac{\sigma_{k_j}\gamma_{k_j}\epsilon_0}{\kappa_{21}(\bar{D}_{k_j})^2_{ii}}. \tag{5.81}$$

Using (5.74), (5.79), (5.80), (5.81), $\sigma_{k_j} \geq \sigma$, and the boundedness of $\bar{D}_{k_j}$ this proves that

$$\lim_{j\to+\infty} \gamma_{k_j} = 0.$$

By (5.23) we know that

$$\bar{D}_{k_j} W_{k_j}^T H_{k_j} W_{k_j} \bar{D}_{k_j} + E_{k_j} + \gamma_{k_j} I_{n-m}$$

is positive semi–definite. Hence condition (5.73) and the limits $\lim_{j\to+\infty} \|W_{k_j}^T H_{k_j} s_{k_j}^{\mathsf{q}}\| = 0$ and $\lim_{j\to+\infty} \gamma_{k_j} = 0$ imply that the principal submatrix of $W_{k_j}^T H_{k_j} W_{k_j}$ corresponding to indices l such that $a_l < (u_*)_l < b_l$ is positive semi–definite for j large enough. Since $W(x)^T \nabla^2_{xx}\ell(x,\lambda)W(x)$ is continuous, the second–order necessary optimality conditions are satisfied at x_*. This completes the proof for the decoupled approach.

The proof for the coupled trust–region approach differs only from the proof for the decoupled approach in the use of equations (5.28) and (5.29) and in the use of $\|W_{k_j} o_{k_j}^{\mathsf{c}}\| \leq (1+\nu_9)\delta_{max}$ to bound the right hand side of inequality (5.77). $\square$

Remark 5.5.1 The global convergence results of Sections 5.4 and 5.5 hold if the quadratic $\Psi_k(s_u)$ is redefined as $\Psi_k(s_u) = q_k(s_k^{\mathsf{q}} + W_k s_u)$ (see the definitions (5.15) and (5.16)) without the Newton augmentation term $\frac{1}{2}s_u^T\left(E_k\bar{D}_k^{-2}\right)s_u$. They are valid also if the matrices D_k and $\bar{D}_k$ are redefined respectively as D_k^p and $\bar{D}_k^p$ with $p \geq 1$. In [41], different forms for this affine scaling matrices are discussed.

5.6 Local Rate of Convergence

We now analyze the local behavior of Algorithms 5.2.1 under Conditions 5.1, 5.3, and 5.4. We start by looking at the behavior of the trust radius close to a nondegenerate point that satisfies the second–order sufficient optimality conditions. For this purpose we require the following lemma.

Lemma 5.6.1 Let Assumptions 5.1–5.6 hold. Under Condition 5.1 the quasi–normal component satisfies

$$\|s_k^{\mathsf{q}}\| \leq \kappa_{22}\|s_k\|, \tag{5.82}$$

where κ_{22} is positive and independent of the iteration counter k.

Proof From $s_k = s_k^{\mathsf{q}} + W_k(s_k)_u$, we obtain

$$\|s_k^{\mathsf{q}}\| \leq \|s_k\| + \|W_k\|\,\|(s_k)_u\|.$$

But since $\|s_k\|^2 = \|(s_k)_y\|^2 + \|(s_k)_u\|^2$, we use Assumption 5.4 to obtain

$$\|s_k^{\mathsf{q}}\| \leq (1+\nu_6)\,\|s_k\|\,,$$

and (5.82) holds with $\kappa_{22} = 1 + \nu_6$. □

Theorem 5.6.1 Let $\{x_k\}$ be a sequence of iterates generated by the TRIP Reduced SQP Algorithms 5.2.1 under Assumptions 5.1–5.6 and Conditions 5.1 and 5.3. If x_k converges to a nondegenerate point x_* satisfying the second–order sufficient optimality conditions, then $\{\rho_k\}$ is a bounded sequence, δ_k is uniformly bounded away from zero, and eventually all the iterations are successful.

Proof It follows from $\lim_{k\to+\infty} x_k = x_*$ and $C(x_*) = 0$ that $\lim_{k\to+\infty}\|C_k\| = 0$. This fact, condition (5.12), and Assumptions 5.3–5.4, together imply the limit $\lim_{k\to+\infty}\|W_k^T H_k s_k^{\mathsf{q}}\| = 0$. Since x_k converges to a nondegenerate point that satisfies the second–order sufficient optimality conditions and $\lim_{k\to+\infty}\|W_k^T H_k s_k^{\mathsf{q}}\| = 0$, there exists a $\bar{\gamma} > 0$ such that the smallest eigenvalue of $\bar{D}_k W_k^T H_k W_k \bar{D}_k + E_k$ is greater than $\bar{\gamma}$ for k sufficiently large.

First we prove that $\{\rho_k\}$ is a bounded sequence. Since $\Psi_k(0) - \Psi_k((s_k)_u) \geq 0$, we obtain

$$\begin{aligned} \tfrac{1}{2}(\bar{D}_k^{-1}(s_k)_u)^T \left(\bar{D}_k W_k^T H_k W_k \bar{D}_k + E_k\right) (\bar{D}_k^{-1}(s_k)_u) &\leq -(\bar{D}_k^{-1}(s_k)_u)^T(\bar{D}_k \bar{g}_k) \\ &\leq \|\bar{D}_k^{-1}(s_k)_u\| \, \|\bar{D}_k \bar{g}_k\|, \end{aligned}$$

which, by using the upper bounds on W_k and $\bar{D}_k$ given by Assumptions 5.4–5.6, implies

$$\|s_k^{\mathsf{t}}\| = \|W_k(s_k)_u\| \leq \frac{2\nu_6\nu_9}{\bar{\gamma}} \|\bar{D}_k \bar{g}_k\|. \tag{5.83}$$

Using (5.44) and (5.83), we find that

$$\begin{aligned} q_k(s_k^{\mathsf{q}}) - q_k(s_k^{\mathsf{q}} + W_k(s_k)_u) &\geq \kappa_6 \|\bar{D}_k \bar{g}_k\| \min\left\{\kappa_7 \|\bar{D}_k \bar{g}_k\|, \kappa_8 \delta_k\right\} \\ &\geq \kappa_{23} \|s_k^{\mathsf{t}}\|^2, \end{aligned} \tag{5.84}$$

where $\kappa_{23} = \frac{\kappa_6\bar{\gamma}}{2\nu_6\nu_9} \min\{\frac{\kappa_7\bar{\gamma}}{2\nu_6\nu_9}, \frac{\kappa_8}{\nu_6\nu_9}, \frac{\kappa_8}{1+\nu_9}\}$ accounts for the decoupled and coupled cases.

Next, we prove that if $\|C_k\| \leq \theta''\|s_k\|$, where θ'' satisfies (5.86) below, then the penalty parameter does not need to be increased. From (5.12) and $\|C_k\| \leq \theta''\|s_k\|$, we get

$$\begin{aligned} \|s_k\|^2 \leq \left(\|s_k^{\mathsf{q}}\| + \|s_k^{\mathsf{t}}\|\right)^2 &\leq 2\|s_k^{\mathsf{q}}\|^2 + 2\|s_k^{\mathsf{t}}\|^2 \\ &\leq 2\theta''\kappa_1^2 \|C_k\| \, \|s_k\| + 2\|s_k^{\mathsf{t}}\|^2. \end{aligned}$$

This estimate, (5.12), (5.42), (5.51), (5.84), and $\|C_k\| \leq \theta''\|s_k\|$ yield

$$\begin{aligned} pred(s_k;\rho) &= q_k(s_k^{\mathsf{q}}) - q_k(s_k^{\mathsf{q}} + W_k(s_k)_u) + q_k(0) - q_k(s_k^{\mathsf{q}}) - \Delta\lambda_k^T(J_k s_k + C_k) \\ &\quad + \rho\left(\|C_k\|^2 - \|J_k s_k + C_k\|^2\right) \\ &\geq \frac{1}{4}\kappa_{23}\|s_k\|^2 + \left(\frac{1}{4}\kappa_{23}\|s_k\| - (\theta''\kappa_1^2\kappa_{23} + \kappa_{11}(\theta''\kappa_1 + 1))\|C_k\|\right)\|s_k\| \\ &\quad + \rho\left(\|C_k\|^2 - \|J_k s_k + C_k\|^2\right), \end{aligned} \tag{5.85}$$

for all $\rho > 0$. If $\|C_k\| \leq \theta''\|s_k\|$, where θ'' satisfies

$$(4\kappa_{11})\,\theta'' + \left(4\kappa_1^2\kappa_{23} + 4\kappa_1\kappa_{11}\right)(\theta'')^2 \leq \kappa_{23}, \tag{5.86}$$

then we set $\rho = \rho_{k-1}$ in (5.85) and deduce that the penalty parameter does not need to be increased. See Step 2.4 of Algorithms 5.2.1. Hence if ρ_k is increased then the inequality $\|C_k\| > \theta''\|s_k\|$ must hold, and we can proceed as in Theorem 5.5.1, equation (5.72), and write

$$\frac{\rho_k}{2}\kappa_2\|C_k\| \min\left\{\kappa_3\|C_k\|, \frac{1}{\kappa_4}\|s_k\|\right\} \le (\kappa_{11}(\kappa_{22}+1) + \bar{\rho}\nu_4)\|s_k\|\,\|C_k\|,$$

(here we used inequality (5.82)) which in turn implies

$$\left(\frac{\kappa_2}{2}\min\left\{\kappa_3\theta'', \frac{1}{\kappa_4}\right\}\right)\ \rho_k \le \kappa_{11}(\kappa_{22}+1) + \bar{\rho}\nu_4.$$

This gives the uniform boundedness of the penalty parameter:

$$\rho_k \le \rho_*$$

for all k.

Given the boundedness of $\{\rho_k\}$ we can complete the proof of the theorem. If $\|C_k\| > \theta''\|s_k\|$, where θ'' satisfies (5.86), then from (5.38), (5.41), and (5.42) we find that

$$pred(s_k;\rho_k) \ge \rho_k\frac{\kappa_2}{2}\|C_k\|\min\{\kappa_3\|C_k\|, \delta_k\} \ge \rho_k\kappa_{24}\|s_k\|^2, \tag{5.87}$$

where $\kappa_{24} = \frac{\kappa_2\theta''}{2}\min\{\kappa_3\theta'', \frac{1}{\kappa_4}\}$. In this case it follows from (5.57) and (5.87) that

$$\left|\frac{ared(s_k;\rho_k)}{pred(s_k;\rho_k)} - 1\right| \le \frac{\kappa_{13}}{\kappa_{24}}\left(\|s_k\| + \|C_k\|\right). \tag{5.88}$$

Now, suppose that $\|C_k\| \le \theta''\|s_k\|$. From (5.85) with $\rho = \rho_k$ we obtain $pred(s_k;\rho_k) \ge \frac{\kappa_{23}}{4}\|s_k\|^2$. Now we use (5.57) and $\rho_k \le \rho_*$ to get

$$\left|\frac{ared(s_k;\rho_k)}{pred(s_k;\rho_k)} - 1\right| \le \frac{4\kappa_{13}\rho_*}{\kappa_{23}}\left(\|s_k\| + \|C_k\|\right). \tag{5.89}$$

Finally from (5.88), (5.89), $\lim_{k\to+\infty} x_k = x_*$, and $\lim_{k\to+\infty}\|C_k\| = 0$, we get

$$\lim_{k\to+\infty}\frac{ared(s_k;\rho_k)}{pred(s_k;\rho_k)} = 1,$$

which by the rules for updating the trust radius given in Step 2.5 of Algorithms 5.2.1, shows that δ_k is uniformly bounded away from zero. $\square$

We use the following straightforward globalization of the quasi–normal component s_k^{q} of the Newton step given in (5.35). The new quasi–normal component is given by:

$$s_k^{\mathsf{q}} = \begin{pmatrix} -\xi_k C_y(x_k)^{-1} C_k \\ 0 \end{pmatrix}, \tag{5.90}$$

where

$$\xi_k = \begin{cases} 1 & \text{if } \|C_y(x_k)^{-1} C_k\| \le \delta_k, \\ \frac{\delta_k}{\|C_y(x_k)^{-1} C_k\|} & \text{otherwise.} \end{cases} \tag{5.91}$$

Before we state the q–quadratic rate of convergence we prove the following important result.

Lemma 5.6.2 Let Assumptions 5.1–5.6 hold. The quasi–normal component (5.90) satisfies conditions (5.9), (5.12), and (5.13) for some positive κ_1, κ_2, and κ_3 independent of k.

Proof It is obvious that (5.9) holds. Condition (5.12) is a direct consequence of the condition (5.13). In fact, using $\|C_y(x_k)(s_k^{\mathsf{q}})_y + C_k\| \le \|C_k\|$ and the boundedness of $\{C_y(x_k)^{-1}\}$ we find that

$$\begin{aligned} \|s_k^{\mathsf{q}}\| &= \|s_k^{\mathsf{q}} + C_y(x_k)^{-1} C_k - C_y(x_k)^{-1} C_k\| \\ &\le \|C_y(x_k)^{-1}\| \Big(\|C_y(x_k)(s_k^{\mathsf{q}})_y + C_k\| + \|C_k\| \Big) \;\le\; 2\nu_6 \|C_k\| . \end{aligned} \tag{5.92}$$

So, let us prove (5.13). A simple manipulation shows that

$$\begin{aligned} \|C_k\|^2 \;-\; & \|C_y(x_k)(s_k^{\mathsf{q}})_y + C_k\|^2 \\ \ge\; & \|C_k\|^2 - \| - \xi_k C_y(x_k) C_y(x_k)^{-1} C_k + C_k\|^2 \\ =\; & \|C_k\|^2 - \Big((1-\xi_k)\|C_k\| \Big)^2 \\ =\; & \xi_k(2-\xi_k)\|C_k\|^2 \;\ge\; \xi_k \|C_k\|^2 . \end{aligned}$$

We need to consider two cases. If $\xi_k = 1$, then

$$\|C_k\|^2 - \|C_y(x_k)(s_k^{\mathsf{q}})_y + C_k\|^2 \ge \|C_k\| \min\{\|C_k\|, \delta_k\}.$$

Otherwise, $\xi_k = \frac{\delta_k}{\|C_y(x_k)^{-1} C_k\|}$. In this case we get

$$\|C_k\|^2 - \|C_y(x_k)(s_k^{\mathsf{q}})_y + C_k\|^2 \ge \frac{1}{\nu_6}\|C_k\|\, \delta_k \ge \frac{1}{\nu_6}\|C_k\| \min\{\|C_k\|, \delta_k\}.$$

Thus the result holds with $\kappa_2 = \min\{1, \frac{1}{\nu_6}\}$ and $\kappa_3 = 1$. $\square$

Corollary 5.6.1 Let $\{x_k\}$ be a sequence of iterates generated by the TRIP Reduced SQP Algorithms 5.2.1 under Assumptions 5.1–5.6 and Conditions 5.1, 5.3, and 5.4. If x_k converges to a nondegenerate point x_* satisfying the second–order sufficient optimality conditions, then x_k converges q–quadratically.

Proof We start by showing that $|\tau_k^{\mathsf{N}} - 1|$ is $\mathcal{O}\left(\|x_k - x_*\|\right)$, where τ_k^{N} is given by (5.37). Since $\lim_{k\to+\infty} \|W_k^T H_k s_k^{\mathsf{q}}\| = 0$, we have that $\left|\frac{\tau_k^{\mathsf{N}}}{\sigma_k} - 1\right|$ is $\mathcal{O}(\|(s_k^{\mathsf{N}})_u\|)$ (see [23, Equation (6.4) and Lemma 12]). Also since by Condition 5.4 $|\sigma_k - 1|$ is $\mathcal{O}\left(\left\|\bar{D}_k \bar{g}_k\right\|\right)$, and $\bar{D}_k \bar{g}_k$ is $\mathcal{O}\left(\|(s_k^{\mathsf{N}})_u\|\right)$ (see (5.34)), we can see that $|\sigma_k - 1|$ is also $\mathcal{O}\left(\|(s_k^{\mathsf{N}})_u\|\right)$. Furthermore,

$$|\tau_k^{\mathsf{N}} - 1| \le \sigma_k \left|\frac{\tau_k^{\mathsf{N}}}{\sigma_k} - 1\right| + |\sigma_k - 1|.$$

Hence $|\tau_k^{\mathsf{N}} - 1|$ is $\mathcal{O}\left(\|(s_k^{\mathsf{N}})_u\|\right)$. But $(s_k^{\mathsf{N}})_u$ is $\mathcal{O}\left(\|x_k + s_k^{\mathsf{q}} - x_*\|\right)$ and s_k^{q} is $\mathcal{O}\left(\|x_k - x_*\|\right)$ and this shows that $|\tau_k^{\mathsf{N}} - 1|$ is $\mathcal{O}\left(\|x_k - x_*\|\right)$.

We need to prove that Condition 5.4 does not conflict with Condition 5.1 so that Theorem 5.6.1 can be applied. In other words, we need to show that the decrease conditions given in Condition 5.1 hold for the Newton damped step (5.36) whenever it is taken. In Lemma 5.6.2 we showed that the quasi–normal component s_k^{q} given in (5.90) satisfies (5.9), (5.12), and (5.13). From Condition 5.4, s_k^{q} given by (5.35) is used when it coincides with the s_k^{q} given by (5.90). Thus s_k^{q} given by (5.35) satisfies also (5.9), (5.12), and (5.13). It remains to prove that $\tau_k^{\mathsf{N}}(s_k^{\mathsf{N}})_u$ satisfies the Cauchy decrease condition (5.21) ((5.22) for the coupled approach). This is indeed the case since

$$\begin{aligned}
&\Psi_k(0) - \Psi_k(\tau_k^{\mathsf{N}}(s_k^{\mathsf{N}})_u) \\
&\quad \ge \; -\tau_k^{\mathsf{N}} \bar{g}_k^T (s_k^{\mathsf{N}})_u - \frac{1}{2}(\tau_k^{\mathsf{N}})^2 ((s_k^{\mathsf{N}})_u)^T \left(W_k^T H_k W_k + E_k \bar{D}_k^{-2}\right)((s_k^{\mathsf{N}})_u) \\
&\quad \ge \; \tau_k^{\mathsf{N}} \left(-\bar{g}_k^T (s_k^{\mathsf{N}})_u - \frac{1}{2}((s_k^{\mathsf{N}})_u)^T \left(W_k^T H_k W_k + E_k \bar{D}_k^{-2}\right)((s_k^{\mathsf{N}})_u)\right) \\
&\quad \ge \; \tau_k^{\mathsf{N}} \left(\Psi_k(0) - \Psi_k(c_k^{\mathsf{d}})\right),
\end{aligned}$$

and $|\tau_k^{\mathsf{N}} - 1|$ is $\mathcal{O}\left(\|x_k - x_*\|\right)$.

Now we need to show that eventually s_k is given by (5.36). Since $\{x_k\}$ converges to a nondegenerate point satisfying the second–order sufficient optimality conditions,

$(s_k^{\mathsf{N}})_u$ exists for k sufficiently large. Furthermore $(s_k^{\mathsf{q}})_y = -C_y(x_k)^{-1}C_k$ for k large enough because $\lim_{k\to+\infty} \|C_y(x_k)^{-1}C_k\| = 0$, and from Theorem 5.6.1, δ_k is eventually bounded away from zero. Using a similar argument we see that $\tau_k^{\mathsf{N}}(s_k^{\mathsf{N}})_u$ is inside the trust region (5.18) for the decoupled approach or (5.20) for the coupled approach. So, from Condition 5.4 we conclude that there exists a positive integer $\bar{k}$ such that s_k is given by (5.36) for $k \geq \bar{k}$.

Using the fact that $(s_k^{\mathsf{N}})_u$ is $\mathcal{O}\left(\|x_k - x_*\|\right)$, we conclude that $\tau_k^{\mathsf{N}}(s_k^{\mathsf{N}})_u - (s_k^{\mathsf{N}})_u$ is $\mathcal{O}\left(\|x_k - x_*\|^2\right)$. Thus

$$s_k - s_k^{\mathsf{N}} = \begin{pmatrix} s_k^{\mathsf{q}} - C_y(x_k)^{-1}C_u(x_k)\tau_k^{\mathsf{N}}(s_k^{\mathsf{N}})_u \\ \tau_k^{\mathsf{N}}(s_k^{\mathsf{N}})_u \end{pmatrix} - \begin{pmatrix} s_k^{\mathsf{q}} - C_y(x_k)^{-1}C_u(x_k)(s_k^{\mathsf{N}})_u \\ (s_k^{\mathsf{N}})_u \end{pmatrix}$$

is $\mathcal{O}\left(\|x_k - x_*\|^2\right)$. This completes the proof since s_k^{N} can be seen as a Newton step on a given vector function of the type (5.8). This function vanishes at x_* and is continuously differentiable with Lipschitz continuous derivatives and a nonsingular Jacobian matrix in an open neighborhood of x_*. See the discussion at the end of Section 5.1. Thus the q–quadratic rate of convergence follows from [39][Theorem 5.2.1] and from the fact that $s_k - s_k^{\mathsf{N}}$ is $\mathcal{O}\left(\|x_k - x_*\|^2\right)$. $\square$

5.7 Computation of Steps and Multiplier Estimates

When we described the TRIP reduced SQP algorithms in Section 5.2, we deferred the practical computation of the quasi–normal and tangential components and of the Lagrange multipliers. In this section we address these issues.

The quasi–normal component s_k^{q} is an approximate solution of the trust–region subproblem (5.10)–(5.11). To guarantee global convergence to a point that satisfies the necessary optimality conditions, the component s_k^{q} is required to satisfy (5.9), (5.12), and (5.13). As we saw in equation (5.92) of the proof of Lemma 5.6.2, property (5.12) is a consequence of (5.13). Whether property (5.13) holds depends on the way in which the quasi–normal component is computed. We showed in Lemma 5.6.2 that the quasi–normal component given by (5.90) satisfies conditions (5.9), (5.12), and (5.13). We show in Section 6.3 that these conditions are satisfied for many other reasonable ways to compute s_k^{q}.

5.7.1 Computation of the Tangential Component

In this section we show how to derive conjugate–gradient algorithms to compute $(s_k)_u$. In [8], Branch, Coleman, and Li propose other practical algorithms to compute steps for trust–region subproblems that come from optimization problems with simple bounds. They use three dimensional subspace approximations and conjugate gradients.

Let us consider first the decoupled trust–region approach given in Section 5.2.2. If we ignore the bound constraints for the moment, we can apply the Conjugate–Gradient Algorithm 2.3.2 proposed by Steihaug [134] and Toint [139] to solve the problem

$$\begin{array}{ll} \text{minimize} & \Psi_k(s_u) \\ \text{subject to} & \|\bar{D}_k^{-1} s_u\| \le \delta_k. \end{array}$$

However we also need to incorporate the constraints

$$\sigma_k(a - u_k) \le s_u \le \sigma_k(b - u_k).$$

This leads to the following algorithm:

Algorithm 5.7.1 *(Computation of $s_k = s_k^{\mathsf{q}} + W_k(s_k)_u$ (Decoupled Case))*

1 Set $s_u^0 = 0$, $r^0 = -\bar{g}_k = -W_k^T \nabla q_k(s_k^{\mathsf{q}})$, $q^0 = \bar{D}_k^2 r^0$, $d^0 = q^0$, and $\epsilon > 0$.

2 For $i = 0, 1, 2, \ldots$ do

2.1 Compute $\gamma^i = \frac{(r^i)^T(q^i)}{(d^i)^T(W_k^T H_k W_k + E_k \bar{D}_k^{-2})(d^i)}$.

2.2 Compute

$$\begin{aligned} \tau^i = \max\{\tau > 0 \ : \ & \|\bar{D}_k^{-1}(s_u^i + \tau d^i)\| \le \delta_k, \\ & \sigma_k(a - u_k) \le s_u^i + \tau d^i \le \sigma_k(b - u_k)\}. \end{aligned}$$

2.3 If $\gamma^i \le 0$, or if $\gamma^i > \tau^i$, then set $(s_k)_u = s_u^i + \tau^i d^i$, where τ^i is given as in 2.2 and go to 3; otherwise set $s_u^{i+1} = s_u^i + \gamma^i d^i$.

2.4 Update the residuals: $r^{i+1} = r^i - \gamma^i (W_k^T H_k W_k + E_k \bar{D}_k^{-2}) d^i$ and $q^{i+1} = \bar{D}_k^2 r^{i+1}$.

2.5 Check truncation criteria: if $\sqrt{\frac{(r^{i+1})^T(q^{i+1})}{(r^0)^T(q^0)}} \le \epsilon$, set $(s_k)_u = s_u^{i+1}$ and go to 3.

2.6 Compute $\alpha^i = \frac{(r^{i+1})^T(q^{i+1})}{(r^i)^T(q^i)}$ and set $d^{i+1} = q^{i+1} + \alpha^i d^i$.

3 Compute $s_k = s_k^{\mathsf{q}} + W_k(s_k)_u$ and stop.

Step 2 of Algorithm 5.7.1 iterates entirely in the vector space of the u variables. After the u component of the step s_k has been computed, Step 3 finds its y component. The decoupled approach allows an efficient use of an approximation $\tilde{H}_k$ to the reduced Hessian $W_k^T \nabla_{xx}^2 \ell_k W_k$. In this case, only two linear system solves are required, one with $C_y(x_k)^T$ to compute $\bar{g}_k$ in Step 1, and the other with $C_y(x_k)$ to compute $W_k(s_k)_u$ in Step 3. If it is the Hessian H_k that is being approximated, then the total number of linear systems is $2I(k)+2$, where $I(k)$ is the number of conjugate–gradient iterations. See Table 5.1.

One can transform this algorithm to work in the whole space rather then in the reduced space by considering the coupled trust–region approach given in Section 5.2.2. This alternative is presented below.

Algorithm 5.7.2 *(Computation of $s_k = s_k^{\mathsf{q}} + W_k(s_k)_u$ (Coupled Case))*

1 Set $s^0 = 0$, $r^0 = -\bar{g}_k = -W_k^T \nabla q_k(s_k^{\mathsf{q}})$, $q^0 = \bar{D}_k^2 r^0$, $d^0 = W_k q^0$, and $\epsilon > 0$.

2 For $i = 0, 1, 2, \ldots$ do

2.1 Compute $\gamma^i = \frac{(r^i)^T(q^i)}{(d^i)^T H_k (d^i) + (d^i)_u^T E_k \bar{D}_k^{-2} (d^i)_u}$.

2.2 Compute

$$\tau^i = \max\Big\{\tau > 0 \ : \ \left\| \begin{pmatrix} -C_y(x_k)^{-1} C_u(x_k) \tau (d^i)_u \\ \bar{D}_k^{-1} \tau (d^i)_u \end{pmatrix} \right\| \le \delta_k\,, \\ \sigma_k(a - u_k) \le s_u^i + \tau (d^i)_u \le \sigma_k(b - u_k)\Big\}.$$

2.3 If $\gamma^i \le 0$, or if $\gamma^i > \tau^i$, then $s_k^{\mathsf{t}} = s^i + \tau^i d^i$, where τ^i is given as in 2.2 and go to 3; otherwise set $s^{i+1} = s^i + \gamma^i d^i$.

2.4 Update the residuals: $r^{i+1} = r^i - \gamma^i \left(W_k^T H_k d^i + E_k \bar{D}_k^{-2} (d^i)_u \right)$ and $q^{i+1} = \bar{D}_k^2 r^{i+1}$.

2.5 Check truncation criteria: if $\sqrt{\frac{(r^{i+1})^T(q^{i+1})}{(r^0)^T(q^0)}} \le \epsilon$, set $s_k^{\mathsf{t}} = s^{i+1}$ and go to 3.

2.6 Compute $\alpha^i = \frac{(r^{i+1})^T(q^{i+1})}{(r^i)^T(q^i)}$ and set $d^{i+1} = W_k(q^{i+1} + \alpha^i d^i)$.

3 Compute $s_k = s_k^{\mathsf{q}} + s_k^{\mathsf{t}}$ and stop.

Note that in Step 2 of Algorithm 5.7.2 both the y and the u components of the tangential component are being computed. The coupled approach is suitable particularly when an approximation to the full Hessian H_k is used. The coupled approach can be used also with an approximation $\tilde{H}_k$ to the reduced Hessian $W_k^T \nabla_{xx}^2 \ell_k W_k$. In this case, we consider H_k that is given by (5.32) and use equalities (5.33) to compute the terms involving H_k in Algorithm 5.7.2. If the Hessian H_k is approximated, the total number of linear system solves is $2I(k)+2$. If the reduced Hessian $W_k^T \nabla_{xx}^2 \ell_k W_k$ is approximated, this number is $I(k)+2$, where $I(k)$ is the number of conjugate–gradient iterations. See Table 5.1.

Linear solver	Decoupled		Coupled	
	Reduced $\tilde{H}_k$	Full H_k	Reduced $\tilde{H}_k$	Full H_k
$C_y(x_k)$	1	$I(k)+1$	$I(k)+1$	$I(k)+1$
$C_y(x_k)^T$	1	$I(k)+1$	1	$I(k)+1$

Table 5.1 Number of linearized state and adjoint solvers to compute the tangential component. ($I(k)$ denotes the number of conjugate–gradient iterations.)

Since Conjugate–Gradient Algorithms 5.7.1 and 5.7.2 start by minimizing the quadratic function $\Psi_k(s_u)$ along the direction $-\bar{D}_k^2 \bar{g}_k$, it is quite clear from Proposition 2.3.1 that they produce reduced tangential components $(s_k)_u$ that satisfy (5.21) and (5.22), respectively, with $\beta_1^{\mathsf{d}} = \beta_1^{\mathsf{c}} = 1$.

We end this section with the following remark.

Remark 5.7.1 For simplicity let us consider the case $\mathcal{B} = \mathbb{R}^{n-m}$. If $W_k^T W_k$ was included as a preconditioner in the Algorithm 5.7.2 derived for the coupled approach, then the conjugate–gradient iterates would monotonically increase in the norm $\|W_k \cdot\|$. Dropping this preconditioner means that the conjugate–gradient iterates do not necessarily increase in this norm (see [134]). As a result, if the quasi–Newton step $-\left(W_k^T H_k W_k\right)^{-1} \bar{g}_k$ exists and is inside the trust region, Algorithm 5.7.2 can terminate prematurely by stopping at the boundary of the trust region. This problem does not arise using Algorithm 5.7.1 for the decoupled approach.

5.7.2 Computation of Multiplier Estimates

A convenient estimate for the Lagrange multipliers is the adjoint update

$$\lambda_k = -C_y(x_k)^{-T}\nabla_y f_k, \tag{5.93}$$

which we use after each successful step. However we also consider the following update:

$$\lambda_{k+1} = -C_y(x_k)^{-T}\nabla_y q_k(s_k^{\mathfrak{q}}) = -C_y(x_k)^{-T}\Big((H_k s_k^{\mathfrak{q}})_y + \nabla_y f_k\Big). \tag{5.94}$$

Here the use of (5.94) instead of

$$\lambda_{k+1} = -C_y(x_k + s_k)^{-T}\nabla_y f(x_k + s_k), \tag{5.95}$$

might be justified since we obtain (5.94) without any further cost from the first iteration of any of the conjugate–gradient algorithms described above. The updates (5.93), (5.94), and (5.95) satisfy the requirement given by Assumption 5.4 needed to prove global convergence to a point satisfying the first–order necessary optimality conditions.

5.8 Numerical Example

We implemented the TRIP Reduced SQP Algorithms 5.2.1 in `Fortran 77`. This implementation is described in [76]. In this section we report numerical results for the boundary control problem introduced in Section 4.5.1. These results demonstrate the effectiveness of the algorithms. We use the formula (5.90) to compute the quasi–normal component, and Algorithms 5.7.1 and 5.7.2 to calculate the tangential component. The numerical test computations were done on a Sun Sparcstation 10 in double precision.

With the discretization scheme discussed in Section 4.5.1, $C_y(x)$ is a block bidiagonal matrix with tridiagonal blocks. Hence linear systems with $C_y(x)$ and $C_y(x)^T$ can be solved efficiently. In the implementation, the `LINPACK` subroutine `DGTSL` was used to solve the tridiagonal systems. As we pointed out in Section 4.6, the inner products and norms used in the TRIP reduced SQP algorithms are not necessarily the Euclidean ones. In our implementation [76], we call subroutines to calculate the inner products $\langle y^1, y^2 \rangle$ and $\langle u^1, u^2 \rangle$ with $y^1, y^2 \in \mathbb{R}^m$ and $u^1, u^2 \in \mathbb{R}^{n-m}$. The user may supply these subroutines to incorporate a specific scaling. If the inner product $\langle x^1, x^2 \rangle$ is required, then it is calculated as $\langle y^1, y^2 \rangle + \langle u^1, u^2 \rangle$. In this example, we used discretizations of the $L^2(0,T)$ and $L^2(0,T; H^1(0,1))$ norms for the control and the state spaces respectively. This is important for the correct computation of the adjoint and the appropriate scaling of the problem.

In our numerical example we use the functions

$$\tau(y) = q_1 + q_2 y, \quad y \in \mathbb{R}, \quad \kappa(y) = r_1 + r_2 y, \quad y \in \mathbb{R},$$

with parameters $r_1 = q_1 = 4$, $r_2 = -1$, and $q_2 = 1$. The desired and initial temperatures, and the right hand side are given by

$$\begin{aligned} y_d(t) &= 2 - e^{\eta t}, \\ y_0(x) &= 2 + \cos \pi x, \quad \text{and} \\ q(x,t) &= [\eta(q_1 + 2q_2) + \pi^2(r_1 + 2r_2)]e^{\eta t} \cos \pi x \\ &\quad - r_2 \pi^2 e^{2\eta t} + (2 r_2 \pi^2 + \eta q_2) e^{2\eta t} \cos^2 \pi x, \end{aligned}$$

with $\eta = -1$. The final temperature is chosen to be $T = 0.5$ and the scalar $g = 1$ is used in the boundary condition. The functions in this example are those used in [89, Example 4.1]. The size of the problem tested is $n = 2200$, $m = 2100$ corresponding to the values $N_t = 100$, $N_x = 20$.

The scheme used to update the trust radius is the following fairly standard one:

- If $ratio(s_k; \rho_k) < 10^{-4}$, reject s_k and set $\delta_{k+1} = 0.5\, norm(s_k)$;
- If $10^{-4} \le ratio(s_k; \rho_k) < 0.1$, accept s_k and set $\delta_{k+1} = 0.5\, norm(s_k)$;
- If $0.1 \le ratio(s_k; \rho_k) < 0.75$, accept s_k and set $\delta_{k+1} = \delta_k$;
- If $ratio(s_k; \rho_k) \ge 0.75$, accept s_k and set $\delta_{k+1} = \min\{2\delta_k, 10^{10}\}$;

where $ratio(s_k; \rho_k) = \frac{ared(s_k;\rho_k)}{pred(s_k;\rho_k)}$,

$$norm(s_k) = \max\left\{ \|s_k^{\mathsf{q}}\|, \|\bar{D}_k^{-1}(s_k)_u\| \right\}$$

in the decoupled approach, and

$$norm(s_k) = \max \left\{ \|s_k^{\mathsf{q}}\|, \left\| \begin{pmatrix} -C_y(x_k)^{-1} C_u(x_k)(s_k)_u \\ \bar{D}_k^{-1}(s_k)_u \end{pmatrix} \right\| \right\}$$

in the coupled approach. The algorithms are stopped if the trust radius gets below 10^{-8}.

We used $\sigma_k = \sigma = 0.99995$ for all k; $\delta_0 = 1$ as initial trust radius; $\rho_{-1} = 1$ and $\bar{\rho} = 10^{-2}$ in the penalty scheme. The tolerance used in the conjugate–gradient iteration was $\epsilon = 10^{-4}$. The upper and lower bounds were $b_i = 10^{-2}$, $a_i = -1000$, $i = 1, \ldots, n-m$. The starting vector was $x_0 = 0$.

For both the decoupled and the coupled approaches, we did tests using approximations to reduced and to full Hessians. We approximate these matrices with the limited memory BFGS representations given in [15] with a memory size of 5 pairs of vectors. For the reduced Hessian we use a null–space secant update (see [114], [147]). The initial approximation chosen was γI_{n-m} for the reduced Hessian and γI_n for the full Hessian, where γ is the user specified regularization parameter in the objective function (4.26).

In our implementation we use the following form of the diagonal matrix $\bar{D}_k$

$$(\bar{D}_k)_{ii} = \begin{cases} \min\{1, (b - u_k)_i\} & \text{if} \quad (\bar{g}_k)_i < 0, \\ \min\{1, (u_k - a)_i\} & \text{if} \quad (\bar{g}_k)_i \geq 0, \end{cases} \tag{5.96}$$

for $i = 1, \ldots, n-m$. This form of $\bar{D}_k$ gives a better transition between the infinite and finite bound and is less sensitive to the introduction of meaningless bounds. See also Remark 5.5.1.

The algorithms were stopped when

$$\|D_k W_k^T \nabla f_k\| + \|C_k\| < 10^{-8}. \tag{5.97}$$

The results are shown in Tables 5.2 and 5.3 corresponding to the values $\gamma = 10^{-2}$ and $\gamma = 10^{-3}$, respectively. There were no rejected steps. The different alternatives tested performed quite similarly. The decoupled approach with reduced Hessian approximation seems to be the best for this example. Note that in this case the computation of each step costs only three linear system solves with $C_y(x_k)$ and $C_y(x_k)^T$, one to compute the quasi–normal component and two for the computation of the tangential component.

	Decoupled		Coupled	
	Reduced $\tilde{H}_k$	Full H_k	Reduced $\tilde{H}_k$	Full H_k
number of iterations k^*	14	20	17	18
$\Vert C_{k^*}\Vert$	$.5082E-11$	$.1370E-10$	$.7122E-12$	$.8804E-11$
$\Vert D_{k^*}W_{k^*}^T\nabla f_{k^*}\Vert$	$.4033E-08$	$.1389E-08$	$.6365E-10$	$.2641E-08$
$\Vert s_{k^*-1}\Vert$	$.1230E-04$	$.1461E-04$	$.3546E-05$	$.1445E-04$
δ_{k^*-1}	$.1638E+05$	$.1049E+07$	$.1311E+06$	$.2621E+06$
ρ_{k^*-1}	$.1000E+01$	$.1000E+01$	$.1000E+01$	$.1000E+01$

Table 5.2 Numerical results for the boundary control problem. Case $\gamma = 10^{-2}$.

We made an experiment to compare the use of the Coleman–Li affine scaling with the Dikin–Karmarkar affine scaling. When applied to our class of problems, the Coleman–Li affine scaling is given by the matrices D_k and $\bar{D}_k$. A study of the Dikin–Karmarkar affine scaling for steepest descent is given in [128]. For our class of problems, this scaling is given by

$$(K_k)_{ii} = \min\{1,\ (u_k - a)_i,\ (b - u_k)_i\},\ i = 1, \ldots, n - m, \tag{5.98}$$

and has no dual information built in. We ran the TRIP reduced SQP algorithm with the decoupled and reduced Hessian approximation and (5.96) replaced by (5.98). The algorithm took only 11 iterations to reduce $\Vert K_k W_k^T \nabla f_k\Vert + \Vert C_k\Vert$ to 10^{-8}. However, as we can see from the plots of the controls in Figures 5.5 and 5.6, the algorithm did not find the correct solution when it used the Dikin–Karmarkar affine scaling (5.98). Some of the variables are at the wrong bound corresponding to negative multipliers.

	Decoupled		Coupled	
	Reduced $\tilde{H}_k$	Full H_k	Reduced $\tilde{H}_k$	Full H_k
number of iterations k^*	16	18	17	19
$\|C_{k^*}\|$	$.6233E-11$	$.1115E-10$	$.6487E-11$	$.1246E-09$
$\|D_{k^*}W_{k^*}^T\nabla f_{k^*}\|$	$.5161E-08$	$.2539E-08$	$.7282E-09$	$.4696E-08$
$\|s_{k^*-1}\|$	$.1626E-04$	$.1703E-04$	$.1530E-04$	$.4659E-04$
δ_{k^*-1}	$.6554E+05$	$.2621E+06$	$.1311E+06$	$.5243E+06$
ρ_{k^*-1}	$.1000E+01$	$.1000E+01$	$.1000E+01$	$.1000E+01$

Table 5.3 Numerical results for the boundary control problem. Case $\gamma = 10^{-3}$.

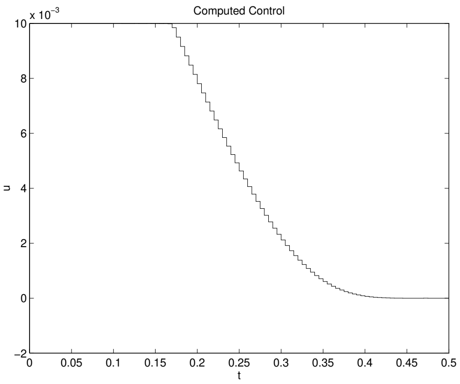

Figure 5.5 Control plot using the Coleman–Li affine scaling.

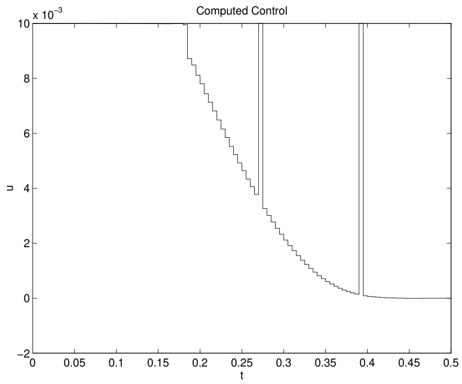

Figure 5.6 Control plot using the Dikin–Karmarkar affine scaling.

Chapter 6

Analysis of Inexact Trust–Region Interior–Point Reduced SQP Algorithms

In Chapter 5, we assumed that exact derivative information for f and C is available, and that linear systems like the linearized state and adjoint equations

$$C_y(x_k)s = \bar{b}_k \quad \text{and} \quad C_y(x_k)^T s = \hat{b}_k \tag{6.1}$$

can be solved exactly for different right hand sides $\bar{b}_k$ and $\hat{b}_k$. In many applications these assumptions are unrealistic. Derivative information may be approximated, for example, by finite differences. Moreover, the linearized state and adjoint equations are often discretizations of partial differential equations and iterative methods are used for their solution. The purpose of this chapter is to extend the TRIP reduced SQP algorithms proposed and analyzed in Chapter 5 to allow inexact calculations in tasks involving first derivatives of C. See also the paper by Heinkenschloss and Vicente [77]. Inexactness in derivatives of the objective function f also can be allowed, but it is not done here to keep the presentation simpler. Since we treat states and controls as independent variables (see Section 4.2), and since the objective functions are often rather simple, e.g. least squares functionals, this does not present a severe restriction. One goal for our analysis is to derive measures of inexactness and forms of controlling the inexactness that are simple to implement.

In the TRIP reduced SQP algorithms, we have to compute quantities of the form $C_u(x_k)d_u$ and $C_u^T(x_k)d_y$, and we have to solve linear systems of the form (6.1). Since these systems are solved inexactly, what is computed are $\bar{s}_k$ and $\hat{s}_k$ such that

$$C_y(x_k)\bar{s}_k = \bar{b}_k + \bar{r}_k \quad \text{and} \quad C_y(x_k)^T \hat{s}_k = \hat{b}_k + \hat{r}_k,$$

where $\bar{r}_k$ and $\hat{r}_k$ are residual vectors. In many iterative methods, like for instance Krylov subspace methods (see the books [73], [81]), the norms $\|\bar{r}_k\|$ and $\|\hat{r}_k\|$ can be computed efficiently with few extra operations. These are some of the quantities that are used to measure inexactness.

We give conditions on the amount of inexactness allowed in the TRIP reduced SQP algorithms that guarantee global convergence to a point satisfying the first–

order necessary optimality conditions. In the case of the linear systems (6.1), these conditions are the following:

$$\|\bar{r}_k\| = \mathcal{O}\Big(\min\{\delta_k, \|C_k\|\}\Big) \quad \text{and} \quad \|\hat{r}_k\| = \mathcal{O}(\|C_k\|), \tag{6.2}$$

where δ_k is the trust radius and $\|C_k\|$ is the norm of the residual of the constraints. Thus as the iterates approach feasibility the accuracy with which the linear systems are solved has to increase. Moreover, the accuracy of the linear systems solves has to increase if the region where the quadratic model is trusted becomes small. This also is reasonable since the trust radius should not be reduced unnecessarily. Similar results can be derived for the inexactness that arises in the computation of directional derivatives of C.

We applied the TRIP reduced SQP algorithms with inexact solutions of linearized state and adjoint equations to the solution of the two optimal control problems described in Section 4.5. The numerical results reported in Section 6.5 confirm our analysis.

It should be pointed out that by inexactness we mean inexact derivative information and inexact solution of linear systems. Trust–region algorithms allow another level of inexactness that is also treated here and in most other papers on trust–region algorithms: the trust–region subproblems do not have to be solved exactly. As we saw for instance in Section 2.3 for unconstrained optimization, it is sufficient to compute steps that predict either a fraction of Cauchy decrease or a fraction of optimal decrease for the trust–region subproblem.

In the context of systems of nonlinear equations, inexact or truncated Newton methods have been proposed and analyzed by many authors. Some of the pioneering work in this area can be found in [32], [135]. More recent references are [9], [10], [43], [44], [45]. Most of the recent papers investigate the use of Krylov subspace methods for the solution of linear systems, like GMRES [127], in inexact Newton methods. These Krylov subspace methods are attractive because they monitor the residual norm of the linear system in an efficient way and only require Jacobian times a vector, not the Jacobian in explicit form. The results for the solution of systems of nonlinear equations have been extended to analyze inexact Newton methods for the solution of unconstrained optimization problems, e.g. [33], [109], [111], inexact Gauss–Newton methods [99], and complementarity problems [117]. In a recent paper [149], the impact of inexactness in reduced–gradient methods for design optimization has been analyzed.

In nonlinear programming, inexactness has been studied by [6], [28], [34], [54], [92], [110], [146] among others. The papers [34], [54], [92], [110] investigating SQP algorithms mostly study the influence of inexactness on the local convergence rate. In [110] conditions on the inexactness are given that guarantee descent in the merit function. In the papers mentioned previously, the inexactness is often measured using the residual of the linearization of the system of nonlinear equations arising from the first–order necessary optimality conditions, or some variation thereof. If globalizations are included in the investigations, then line–search strategies are used. To our knowledge, inexactness for SQP algorithms with trust–region globalizations has not been studied in the literature. Due to the computation of the step in two stages, the computation of the quasi–normal component and of the tangential component, the analysis of inexactness in reduced SQP algorithms with trust–region globalizations requires techniques different from those that can be used for line–search globalizations.

This chapter is organized as follows. In Section 6.1, we identify the sources of inexactness in the TRIP reduced SQP algorithms and derive a useful representational form. In Section 6.2, we present our inexact analysis showing under what assumptions on the amount of inexactness do the TRIP reduced SQP algorithms remain globally convergent to a point satisfying the first–order necessary optimality conditions. The remainder of the chapter deals with practical issues concerning the step components and multipliers calculations. As we saw in Section 5.2, each step is decomposed in two components: a quasi–normal component and a tangential component. In Section 6.3, we present several techniques to compute quasi–normal components and show how they fit into the theoretical framework given in Section 6.2. In Section 6.4, we discuss conjugate–gradient algorithms to compute the tangential component and analyze the influence of the inexactness. The inexact calculation of the reduced gradient, null–space vectors, and multipliers is covered also in detail in Section 6.4. We present our numerical experiments in Section 6.5.

6.1 Sources and Representation of Inexactness

In this chapter we assume that the linear systems with $C_y(x_k)$ and $C_y(x_k)^T$ are solved inexactly.

The inexact analysis for the quasi–normal component is presented in Section 6.3 and does not interfere with the analysis developed in this section. In fact, we assume that the quasi–normal component s_k^{q}, no matter how is computed, satisfies conditions (5.9), (5.12), and (5.13) given in Section 5.2.1. We see in Section 6.3 that this can be accomplished by a variety of techniques to compute quasi–normal components.

The computation of the tangential component requires the calculation of matrix–vector products of the form $W_k d_u$ and $W_k^T d$. Thus we need to compute quantities like

$$-C_y(x_k)^{-1}C_u(x_k)d_u \quad \text{and} \quad -C_u(x_k)^T C_y(x_k)^{-T} d_y.$$

As we pointed out earlier, often these computations cannot be done exactly. Therefore we have to incorporate errors originating perhaps from finite difference approximations of $C_u(x_k)d_u$ or from the iterative solution of the linear systems $C_y(x_k)d_y = -C_u(x_k)d_u$.

In practice, the computation of the y component z_y of $z = W_k d_u$ is done as follows:

$$\begin{array}{lrcl} \text{Compute} & v_y & = & -C_u(x_k)d_u + e_u. \\ \text{Solve} & C_y(x_k)z_y & = & v_y + e_y. \end{array} \tag{6.3}$$

The u component of $W_k d_u$ is equal to d_u. In (6.3) e_u and e_y are the error terms accounting for the inexactness in the computation of $-C_u(x_k)d_u$ and the inexactness in the solution of the linear system $C_y(x_k)z_y = v_y$. Since the u component of W_k is the identity, we only have an error in the y component z_y of $W_k d_u$ computed via (6.3). It holds that

$$z_y = -C_y(x_k)^{-1}C_u(x_k)d_u + C_y(x_k)^{-1}\left(e_u + e_y\right). \tag{6.4}$$

Of course, the errors e_u and e_y depend in general on d_u.

Similarly, for a given d the matrix–vector product $z = W_k^T d$ is computed successively by the following procedure:

$$\begin{array}{lrcl} \text{Solve} & C_y(x_k)^T v_y & = & -d_y + e_y. \\ \text{Compute} & v_u & = & C_u(x_k)^T v_y + e_u. \\ \text{Compute} & z & = & v_u + d_u. \end{array} \tag{6.5}$$

Again, e_u and e_y are error terms accounting for the inexactness in the computation of $C_u(x_k)^T v_y$ and the inexactness in the solution of the linear system $C_y(x_k)^T v_y = -d_y$. For simplicity we use the same notation, but the error terms in (6.5) are different from those in (6.3). The errors e_u and e_y depend in general on d_y. The computed result can be related to the exact result via the equation

$$z = -C_u(x_k)^T C_y(x_k)^{-T} d_y + d_u + C_u(x_k)^T C_y(x_k)^{-T} e_y + e_u. \tag{6.6}$$

These two sources of inexactness influence the computation of the following important quantities:

$$\bar{g}_k = W_k^T \nabla q_k(s_k^{\mathsf{q}}) = -C_u(x_k)^T C_y(x_k)^{-T} \nabla_y q_k(s_k^{\mathsf{q}}) + \nabla_u q_k(s_k^{\mathsf{q}}), \tag{6.7}$$

and

$$s_k = s_k^{\mathsf{q}} + W_k (s_k)_u = s_k^{\mathsf{q}} + \begin{pmatrix} -C_y(x_k)^{-1} C_u(x_k)(s_k)_u \\ (s_k)_u \end{pmatrix}. \tag{6.8}$$

As we saw in Section 5.2.2, these two calculations are the only ones that appear in the decoupled approach for the computation of the tangential component involving derivatives of C if an approximation $\tilde{H}_k$ to the reduced Hessian $W_k^T \nabla_{xx}^2 \ell_k W_k$ is used. This is not the case in all the other situations (see for instance Table 5.1). If an approximation H_k to the full Hessian $\nabla_{xx}^2 \ell_k$ is used, then we have to account for the inexactness in the calculation of $W_k^T H_k W_k$. Thus, there is no guarantee of monotonicity in the quadratic $\Psi_k(s_u)$ in the conjugate–gradient algorithm, and therefore there is no guarantee that the result expressed in (5.44) for a fraction of Cauchy decrease condition would be satisfied. This raises some interesting problems related to the computation of the tangential component that are addressed in Section 6.4. There we show that, instead of (6.7) and (6.8), the inexact operations with derivatives of C lead to quantities in the form

$$P_k^T \nabla q_k(s_k^{\mathsf{q}}) = -A_k \nabla_y q_k(s_k^{\mathsf{q}}) + \nabla_u q_k(s_k^{\mathsf{q}}), \tag{6.9}$$

and

$$s_k = s_k^{\mathsf{q}} + Q_k (s_k)_u = s_k^{\mathsf{q}} + \begin{pmatrix} -B_k (s_k)_u \\ (s_k)_u \end{pmatrix}, \tag{6.10}$$

where $A_k \simeq C_u(x_k)^T C_y(x_k)^{-T}$, $B_k \simeq C_y(x_k)^{-1} C_u(x_k)$,

$$P_k = \begin{pmatrix} -A_k^T \\ I_{n-m} \end{pmatrix}, \text{ and } Q_k = \begin{pmatrix} -B_k \\ I_{n-m} \end{pmatrix}. \tag{6.11}$$

In these expressions, A_k and B_k represent the inexactness. A detailed derivation and analysis of the linear operators A_k and B_k are given in Section 6.4 together with an extension of Algorithms 5.7.1 and 5.7.2 for the computation of the tangential component.

As a consequence of assuming this inexactness, we no longer have condition (5.44). Instead, we have the following condition:

$$\begin{aligned} q_k(s_k^{\mathsf{q}}) &- q_k(s_k^{\mathsf{q}} + Q_k(s_k)_u) \\ &\geq \varsigma_1 \|\bar{D}_k^{\mathsf{P}} P_k^T \nabla q_k(s_k^{\mathsf{q}})\| \min\left\{\varsigma_2 \|\bar{D}_k^{\mathsf{P}} P_k^T \nabla q_k(s_k^{\mathsf{q}})\|,\ \varsigma_3 \delta_k\right\} - \varsigma_4 \|C_k\|, \end{aligned} \tag{6.12}$$

where $\varsigma_1, \ldots, \varsigma_4$ are positive constants independent from k, and $\bar{D}_k^{\mathsf{P}}$ is a diagonal matrix of order $n-m$ with diagonal elements given by

$$(\bar{D}_k^{\mathsf{P}})_{ii} = \begin{cases} (b-u_k)_i^{\frac{1}{2}} & \text{if } \left(P_k^T \nabla q_k(s_k^{\mathsf{q}})\right)_i < 0 \text{ and } b_i < +\infty, \\ 1 & \text{if } \left(P_k^T \nabla q_k(s_k^{\mathsf{q}})\right)_i < 0 \text{ and } b_i = +\infty, \\ (u_k-a)_i^{\frac{1}{2}} & \text{if } \left(P_k^T \nabla q_k(s_k^{\mathsf{q}})\right)_i \geq 0 \text{ and } a_i > -\infty, \\ 1 & \text{if } \left(P_k^T \nabla q_k(s_k^{\mathsf{q}})\right)_i \geq 0 \text{ and } a_i = -\infty, \end{cases} \tag{6.13}$$

for $i = 1, \ldots, n-m$. The matrix $\bar{D}_k^{\mathsf{P}}$ is the inexact version of $\bar{D}_k$. We show in Section 6.4 how this can be satisfied. Of course we still require the tangential component to be feasible with respect to the trust region and bound constraints. See (5.21), (5.22), and Step 2.2 of Algorithms 5.2.1.

6.2 Inexact Analysis

The assumptions on the inexact calculations required for global convergence to a point satisfying the first–order necessary optimality conditions are the following.

Assumptions 6.1–6.3

6.1 The sequences $\{A_k\}$ and $\{B_k\}$ are bounded.

6.2 $\|\left(-C_y(x_k)B_k + C_u(x_k)\right)(s_k)_u\| \leq \min\left\{\frac{1}{\kappa_3}, \frac{\kappa_2}{2}\right\} \min\{\kappa_3\|C_k\|, \delta_k\}$.

6.3 $\lim_{j\to+\infty} \|(-A_{k_j}^T + C_u(x_{k_j})^T C_y(x_{k_j})^{-T})\nabla_y q_{k_j}(s_{k_j}^{\mathsf{q}})\| = 0$ for all index subsequences $\{k_j\}$ such that $\lim_{j\to+\infty} \|C_{k_j}\| = 0$.

The constants κ_2 and κ_3 are used in (5.13) to define the decrease condition for the quasi–normal component. Assumption 6.2 imposes a bound on the distance of

$Q_k(s_k)_u$ to the null space of the Jacobian J_k. It is obvious that Assumption 6.2 is satisfied when $B_k = C_y(x_k)^{-1}C_u(x_k)$. Assumption 6.3 is only needed to derive Theorem 6.2.1 and restricts the accuracy of the reduced–gradient calculation. We will be more precise later. This assumption is satisfied if $A_k = C_u(x_k)^T C_y(x_k)^{-T}$.

For the rest of this Chapter we suppose that Assumptions 5.1–5.6 given in Section 5.2.5 and Assumptions 6.1–6.3 presented above are always satisfied.

For the global convergence of the inexact TRIP reduced SQP algorithms we still need the step components to satisfy the requirements given in Condition 5.1, Section 5.2.5, but with (5.21) or (5.22) replaced by (6.12). The new condition is given below.

Condition 6.1

6.1 The quasi–normal component s_k^{q} satisfies conditions (5.9), (5.12), and (5.13). The tangential component $(s_k)_u$ satisfies the decrease condition (6.12). The parameter σ_k is chosen in $[\sigma, 1)$, where $\sigma \in (0,1)$ is fixed for all k.

6.2.1 Global Convergence to a First–Order Point

In this section we prove global convergence to a point satisfying the first–order necessary optimality conditions for the TRIP reduced SQP algorithms with inexact solutions of linearized state and adjoint equations of the form (6.1). The proof is virtually the same as the one given in Sections 5.3 and 5.4 for exact solutions of these linear systems. Our job consists of pointing out the few places in the proof where inexactness affects the estimates and how are these situations fixed by using Assumptions 6.1–6.3.

The following lemma states a lower bound on the decrease given by s_k on the linearized residual of the equality constraints. The need for this lemma is that, due to the inexactness assumption, the tangential component $s_k^{\mathsf{t}} = W_k(s_k)_u$ might not lie in the null space of J_k.

Lemma 6.2.1 The step s_k satisfies

$$\|C_k\|^2 - \|J_k s_k + C_k\|^2 \geq \frac{\kappa_2}{2}\|C_k\| \min\{\kappa_3\|C_k\|, \delta_k\}. \tag{6.14}$$

Proof From Assumption 6.2 we get

$$\| \left(-C_y(x_k)B_k + C_u(x_k)\right)(s_k)_u\|^2 \leq \frac{1}{\kappa_3}\kappa_3\|C_k\|\frac{\kappa_2}{2}\min\{\kappa_3\|C_k\|, \delta_k\}.$$

Using this inequality, (5.9), (5.13), $s_k = s_k^{\mathsf{q}} + Q_k(s_k)_u$, and the form (6.11) of Q_k, we have

$$\begin{aligned}\|C_k\|^2 - \|J_k s_k + C_k\|^2 \;&\geq\; \|C_k\|^2 - \|C_y(x_k)(s_k^{\mathsf{q}})_y + C_k\|^2 \\ &\quad -\| \left(-C_y(x_k)B_k + C_u(x_k)\right)(s_k)_u\|^2 \\ &\geq\; \tfrac{\kappa_2}{2}\|C_k\| \min\{\kappa_3\|C_k\|, \delta_k\}.\end{aligned}$$

□

The inequality (6.14) is of the form (5.38). The other estimates given in Section 5.3 and required for global convergence to a point satisfying the first–order necessary optimality conditions remain valid. They consist of inequalities (5.40), (5.41), (5.42), (5.43), (5.50), and (5.56).

The following lemma bounds the predicted decrease in a way similar to Lemma 5.4.1.

Lemma 6.2.2 If $(s_k)_u$ satisfies (6.12), then the predicted decrease in the merit function satisfies

$$\begin{aligned} pred(s_k;\rho) \;&\geq\; \varsigma_1\|\bar{D}_k^{\mathsf{P}} P_k^T \nabla q_k(s_k^{\mathsf{q}})\| \min\left\{\varsigma_2\|\bar{D}_k^{\mathsf{P}} P_k^T \nabla q_k(s_k^{\mathsf{q}})\|, \varsigma_3\delta_k\right\} \\ &\quad -(\kappa_{10}+\varsigma_4)\|C_k\| + \rho\Big(\|C_k\|^2 - \|J_k s_k + C_k\|^2\Big), \end{aligned} \tag{6.15}$$

for any $\rho > 0$.

Proof The inequality (6.15) follows from a direct application of (5.50) and (6.12).

□

Given this result, Lemmas 5.4.2, 5.4.3, and 5.4.4 follow as if the calculations were exact. Thus we are able to state the global convergence result that the TRIP reduced SQP algorithms satisfy when the linear systems (6.1) are solved inexactly. This result is the same as in Theorem 5.4.1 and shows that for a subsequence of the iterates, the first–order necessary optimality conditions given in Proposition 4.4.3 for problem (4.1) are satisfied in the limit.

Theorem 6.2.1 Let $\{x_k\}$ be a sequence of iterates generated by the TRIP Reduced SQP Algorithms 5.2.1 for which the steps satisfy Condition 6.1, and assume Assumptions 5.1–5.6 and 6.1–6.3 hold. Then

$$\liminf_{k\to+\infty} \left(\|D_k W_k^T \nabla f_k\| + \|C_k\| \right) = 0. \tag{6.16}$$

Proof From (5.65) we obtain

$$\liminf_{k\to+\infty} \left(\|\bar{D}_k^{\mathsf{P}} P_k^T \nabla q_k(s_k^{\mathsf{q}})\| + \|C_k\| \right) = 0.$$

Thus there exists an index subsequence $\{k_i\}$ such that

$$\lim_{i\to+\infty} \left(\|\bar{D}_{k_i}^{\mathsf{P}} P_{k_i}^T \nabla q_{k_i}(s_{k_i}^{\mathsf{q}})\| + \|C_{k_i}\| \right) = 0.$$

Now we apply Assumption 6.3 and the forms (4.5) and (6.11) of $W_k = W(x_k)$ and P_k, to obtain

$$\lim_{i\to+\infty} \left(P_{k_i} - W_{k_i} \right)^T \nabla q_{k_i}(s_{k_i}^{\mathsf{q}}) = 0.$$

Using this and the continuity of $D(x)W(x)^T \nabla f(x)$ we get

$$\lim_{i\to+\infty} \left(\|\bar{D}_{k_i} W_{k_i}^T \nabla q_{k_i}(s_{k_i}^{\mathsf{q}})\| + \|C_{k_i}\| \right) = \lim_{i\to+\infty} \left(\|\bar{D}_{k_i} \bar{g}_{k_i}\| + \|C_{k_i}\| \right) = 0.$$

The rest of the proof is given in the last paragraph of the proof of Theorem 5.4.1.

□

The condition in Assumption 6.3, that

$$\lim_{j\to+\infty} \|(-A_{k_j} + C_u(x_{k_j})^T C_y(x_{k_j})^{-T}) \nabla_y q_{k_j}(s_{k_j}^{\mathsf{q}})\| = 0$$

for all index subsequences $\{k_j\}$ such that $\lim_{j\to+\infty} \|C_{k_j}\| = 0$, is related to the computation of the reduced gradient. If the adjoint update $\lambda_k = -C_y(x_k)^{-T} \nabla_y f_k$, or an inexact version, is used for the multipliers, then this condition can be interpreted as a restriction on how accurate these multipliers have to be computed. We comment on this again in Section 6.4.

6.2.2 Inexact Directional Derivatives

The result proved in Theorem 6.2.1 covers also the inexact calculation of directional derivatives necessary to compute quantities of the form $W_k d_u$ and $W_k^T d$. However these are not the only places in the TRIP Reduced SQP Algorithms 5.2.1 where directional derivatives of C need to be evaluated. In fact, in the computation of the actual and predicted decreases, we need to evaluate $J_k s_k$ after the step s_k is computed. Since we allow the derivatives of C to be approximated, we do not have $J_k s_k$ but rather

$$J_k s_k + e_k, \tag{6.17}$$

where e_k is an error term. The predicted decrease $pred(s_k;\rho_k)$ is affected by this error and has to be redefined as:

$$\begin{aligned} pred(s_k;\rho_k;e_k) &= L(x_k,\lambda_k;\rho_k) \\ &\quad - \Big(q_k(s_k;e_k) + \Delta\lambda_k^T(J_k s_k + e_k + C_k) + \rho_k\|J_k s_k + e_k + C_k\|^2\Big), \end{aligned}$$

where now the quadratic term $q_k(s_k;e_k)$ is given by

$$\begin{aligned} q_k(s_k;e_k) &= \ell_k + \nabla f_k^T s_k + \lambda_k^T(J_k s_k + e_k) + \tfrac{1}{2} s_k^T H_k s_k \\ &= q_k(s_k) + \lambda_k^T e_k. \end{aligned} \tag{6.18}$$

It can be proved that the global convergence result (6.16) given in Theorem 6.2.1 holds if e_k is $\mathcal{O}\Big(\min\{\|C_k\|, \|s_k\|^2\}\Big)$. This extension of Theorem 6.2.1 is not difficult to show. In fact, by imposing this condition on $\|e_k\|$ the actual versus predicted estimate (5.56) is valid for $pred(s_k;\rho_k;e_k)$. Inequalities (5.38) and (5.50) hold also if we replace $J_k s_k$ by $J_k s_k + e_k$.

6.3 Inexact Calculation of the Quasi–Normal Component

The quasi–normal component s_k^{q} is an approximate solution of the trust–region subproblem

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}\|C_y(x_k)(s^{\mathsf{q}})_y + C_k\|^2 \\ \text{subject to} \quad & \|(s^{\mathsf{q}})_y\| \le \delta_k, \end{aligned} \tag{6.19}$$

and it is required to satisfy the conditions (5.9), (5.12), and (5.13). The property (5.12) is a consequence of (5.13) (see Lemma 5.6.2, equation (5.92)). Whether the property (5.13) holds depends on the way in which the quasi–normal component is computed. We show below that (5.13) is satisfied for a variety of techniques to compute s_k^{q}.

We concentrate on methods that are suitable for the large–scale case and do not require the matrix $C_y(x_k)$ in explicit form. The first two groups of methods tackle the trust–region subproblem (6.19) directly. The first group of methods are Krylov subspace methods that require the computation of matrix–vector products $C_y(x_k)d_y$ and $C_y(x_k)^T d_y$, while the second group of methods only require $C_y(x_k)d_y$. The third group of methods compute steps by solving the linear system $C_y(x_k)(s^{\mathsf{q}})_y = -C_k$ approximately. The trust–region constraint is enforced by scaling the solution.

6.3.1 Methods that Use the Transpose

There are various ways to compute the quasi–normal component s_k^{q} for large–scale problems. For example, one can use the Conjugate–Gradient Algorithm 2.3.2, or one can use the Lanczos bidiagonalization as described in [67]. Both methods compute an approximate solution of (6.19) from a subspace that contains the negative gradient $-C_y(x_k)^T C_k$ of the least squares functional $\frac{1}{2}\|C_y(x_k)(s^{\mathsf{q}})_y + C_k\|^2$. Thus, the components s_k^{q} generated by these algorithms satisfy $\|s_k^{\mathsf{q}}\| \le \delta_k$ and

$$\begin{aligned} & \frac{1}{2}\|C_y(x_k)(s_k^{\mathsf{q}})_y + C_k\|^2 \\ \le \quad & \min\left\{\frac{1}{2}\|C_y(x_k)s + C_k\|^2 \ : \ \|s\| \le \delta_k\,,\ s \in span\{-C_y(x_k)^T C_k\}\right\}. \quad (6.20) \end{aligned}$$

We can appeal to Lemma 2.3.1 to show that

$$\|C_k\|^2 - \|C_y(x_k)(s_k^{\mathsf{q}})_y + C_k\|^2 \ge \frac{1}{2}\|C_y(x_k)^T C_k\| \min\left\{\frac{\|C_y(x_k)^T C_k\|}{\|C_y(x_k)^T C_y(x_k)\|}, \delta_k\right\}.$$

Now one can use the fact that $\{C_y(x_k)^T C_y(x_k)\}$ and $\{C_y(x_k)^{-T}\}$ are bounded sequences (see Assumptions 5.3–5.5 in Section 5.2.5) to prove the following lemma.

Lemma 6.3.1 If $(s_k^{\mathsf{q}})_y$ satisfies (6.20), then there exist positive constants κ_2 and κ_3, independent of k, such that

$$\|C_k\|^2 - \|C_y(x_k)(s_k^{\mathsf{q}})_y + C_k\|^2 \ge \kappa_2\|C_k\| \min\{\kappa_3\|C_k\|, \delta_k\}.$$

Another family of methods to solve large–scale trust–region subproblems is proposed and analyzed in [129], [133]. We described briefly these algorithms in Section 2.3.1 and mentioned that they compute steps satisfying a fraction of optimal decrease condition of the type (2.10). Hence, when applied to the trust–region subproblem (6.19), they produce quasi–normal components that verify (6.20) and Lemma 6.3.1 can be applied to obtain (5.13). In Theorem 3.8.1, Section 3.8, we pointed out the numerical difficulties that these trust–region subproblems are likely to offer to algorithms that compute steps satisfying a fraction of optimal decrease condition. The Lanczos bidiagonalization algorithm in [67] is another algorithm that computes steps satisfying this property when applied to the trust–region subproblem (6.19).

6.3.2 Methods that Are Transpose Free

The Conjugate–Gradient Algorithm 2.3.2, the Lanczos bidiagonalization algorithm [67], and the algorithms in [129], [133] require the computation of matrix–vector products of the form $C_y(x_k)d_y$ and $C_y(x_k)^T d_y$ for a given d_y in $\mathbb{R}^m$. For some applications, the evaluation of $C_y(x_k)^T d_y$ is more expensive than the application of $C_y(x_k)d_y$, and therefore it may be more efficient to use methods that avoid the use of $C_y(x_k)^T d_y$. In this case one can apply nonsymmetric transpose–free Krylov subspace methods based on minimum residual approximations, such as GMRES [127] or TFQMR [55]. In the context of nonlinear system solving the use of such methods is described by Brown and Saad [10], [11]. GMRES and TFQMR generate matrices[||]

$$V_l \in \mathbb{R}^{m\times l}, \quad W_{l+1} \in \mathbb{R}^{m\times(l+1)}, \quad \text{and} \quad H_l \in \mathbb{R}^{(l+1)\times l},$$

such that

$$C_y(x_k)V_l = W_{l+1}H_l \quad \text{and} \quad C_k = \|C_k\|\, W_{l+1}e_1 = \|C_k\|\, V_l e_1. \tag{6.21}$$

The columns of the matrices V_l and W_{l+1} have norm one, and it holds that

$$range(V_l) = \mathcal{K}_l(C_y(x_k), C_k) = span\left\{C_k, C_y(x_k)C_k, \ldots, (C_y(x_k))^{l-1}C_k\right\}, \tag{6.22}$$

i.e. the columns of the matrix V_l form a basis of the Krylov subspace $\mathcal{K}_l(C_y(x_k), C_k)$. If GMRES is used then V_l is orthogonal and $W_{l+1} = V_{l+1}$. Using the identities (6.21), the trust–region subproblem (6.19) can be approximated by

$$\begin{aligned} &\text{minimize} \quad \tfrac{1}{2}\left\|W_{l+1}(H_l z + \|C_k\| e_1)\right\|^2 \\ &\text{subject to} \quad \|V_l z\| \le \delta_k. \end{aligned} \tag{6.23}$$

[||] For the presentation of this approach, we follow the notation used for Krylov subspace methods. Here the matrices W_l and H_l are generated by GMRES or TFQMR and are not the matrices representing the null space of J_k or the approximation to the Hessian of the Lagrangian $\nabla^2_{xx}\ell_k$, respectively.

The quasi–normal component is given by

$$(s_k^{\mathsf{q}})_y = V_l z, \tag{6.24}$$

where $z \in \mathbb{R}^l$ is the solution of (6.23).

If V_l and W_{l+1} are orthogonal, i.e. if GMRES is used, then (6.23) is equivalent to

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\Big\| H_l z + \|C_k\| e_1 \Big\|^2 \\ \text{subject to} & \|z\| \leq \delta_k. \end{array} \tag{6.25}$$

Thus, if V_l and W_{l+1} are orthogonal, then (6.22), (6.24), and (6.25) imply that the quasi–normal component satisfies

$$\frac{1}{2}\|C_y(x_k)(s_k^{\mathsf{q}})_y + C_k\|^2 \leq \min\Big\{\frac{1}{2}\|C_y(x_k)s + C_k\|^2 \ : \ \|s\| \leq \delta_k\,,\ s \in span\{-C_k\}\Big\}. \tag{6.26}$$

In this case we can use slight modifications of Lemma 2.3.1 to establish the following result:

Lemma 6.3.2 Suppose that $W_{l+1} \in \mathbb{R}^{m\times(l+1)}$, $W_l = V_l \in \mathbb{R}^{m\times l}$ are the orthogonal matrices generated by GMRES satisfying (6.21) and (6.22). If $(s_k^{\mathsf{q}})_y$ is given by (6.24) and (6.25) and if

$$\frac{1}{2}C_k^T\left(C_y(x_k)^T + C_y(x_k)\right)C_k \geq \nu_{12}\|C_k\|^2 \tag{6.27}$$

holds with $\nu_{12} > 0$, then

$$\|C_k\|^2 - \|C_y(x_k)(s_k^{\mathsf{q}})_y + C_k\|^2 \geq \kappa_2\|C_k\|\min\{\kappa_3\|C_k\|, \delta_k\},$$

where κ_2 and κ_3 are positive constants that do not depend on k.

Proof We consider the function

$$\psi(t) = -\nu_{12}t\,\|C_k\| + \frac{t^2}{\|C_k\|^2}C_k^T C_y(x_k)^T C_y(x_k) C_k$$

on the interval $[0, \delta_k]$. Using the arguments given in the proof of Lemma 2.3.1, we can show that

$$\psi(t) \leq -\frac{\nu_{12}}{2}\|C_k\|\min\left\{\frac{\nu_{12}}{2}\frac{\|C_k\|}{\|C_y(x_k)^T C_y(x_k)\|^2}, \delta_k\right\}.$$

With the estimate (6.26) and assumption (6.27) this inequality implies

$$\begin{aligned}
&\|C_y(x_k)(s^{\mathsf{q}})_y + C_k\|^2 - \|C_k\|^2 \\
\leq\ & \left\| -t\, C_y(x_k)\frac{C_k}{\|C_k\|} + C_k \right\|^2 - \|C_k\|^2 \\
=\ & -\frac{t}{\|C_k\|} C_k^T C_y(x_k)^T C_k + \frac{t^2}{\|C_k\|^2} C_k^T C_y(x_k)^T C_y(x_k) C_k \\
=\ & -\frac{t}{2\|C_k\|} C_k^T \Big(C_y(x_k)^T + C_y(x_k) \Big) C_k + \frac{t^2}{\|C_k\|^2} C_k^T C_y(x_k)^T C_y(x_k) C_k \\
\leq\ & \psi(t).
\end{aligned}$$

Using the boundedness of $\{C_y(x_k)^T C_y(x_k)\}$, from Assumption 5.3 in Section 5.2.5, this gives the desired result. $\square$

The condition (6.27) is implied by the positive definiteness of the symmetric part of $C_y(x_k)$, a condition also important for the convergence of nonsymmetric Krylov subspace methods [73].

If V_l and W_{l+1} are not orthogonal, e.g. if TFQMR [55] is used, then (6.25) is not equivalent to (6.23). However, as in the context of linear system solving, one can solve (6.25) for z and use (6.24) as a quasi–normal component. Due to the nonorthogonality of V_l and W_{l+1}, one cannot guarantee that (6.26) holds anymore.

6.3.3 Scaled Approximate Solutions

An alternative to the previous procedures is to compute a solution of the linear system $C_y(x_k)s = -C_k$ and to scale this component back into the trust region. The resulting quasi–normal component is given in (5.90).

In this section, we assume that the computed solution $(s_k^{\mathsf{q}})_y$ of the linear system $C_y(x_k)s = -C_k$ satisfies

$$C_y(x_k)(s_k^{\mathsf{q}})_y = -C_k + e_k,$$

where the error e_k can be bounded as

$$\|e_k\| \leq \epsilon \|C_k\|. \tag{6.28}$$

This gives

$$\|(s_k^{\mathsf{q}})_y\| \leq (1+\epsilon)\|C_y(x_k)^{-1}\|\,\|C_k\|. \tag{6.29}$$

Lemma 6.3.3 If the approximate solution $(s_k^{\mathsf{q}})_y$ of the linear system $C_y(x_k)s = -C_k$ satisfies $\|C_y(x_k)(s_k^{\mathsf{q}})_y + C_k\| \le \epsilon\|C_k\|$ with $\epsilon < 1$, then the quasi–normal component (5.90) using this inexact solution is such that:

$$\|C_k\|^2 - \|C_y(x_k)(s_k^{\mathsf{q}})_y + C_k\|^2 \ge \kappa_2\|C_k\| \min\{\kappa_3\|C_k\|, \delta_k\},$$

where κ_2 and κ_3 are positive constants independent of k.

Proof A simple manipulation shows that

$$\begin{aligned} \|C_k\|^2 \quad - & \quad \|C_y(x_k)(s_k^{\mathsf{q}})_y + C_k\|^2 \\ \ge & \quad \|C_k\|^2 - \|\xi_k C_y(x_k)(s_k^{\mathsf{q}})_y + C_k\|^2 \\ \ge & \quad \|C_k\|^2 - \Big((1-\xi_k)\|C_k\| + \xi_k\|C_y(x_k)(s_k^{\mathsf{q}})_y + C_k\|\Big)^2. \end{aligned}$$

Now we use $\|C_y(x_k)(s_k^{\mathsf{q}})_y + C_k\| \le \epsilon\|C_k\|$ and $\xi_k \le 1$, to obtain

$$\begin{aligned} \|C_k\|^2 - \|C_y(x_k)(s_k^{\mathsf{q}})_y + C_k\|^2 & \ge \|C_k\|^2 - \Big((1-\xi_k)\|C_k\| + \epsilon\xi_k\|C_k\|\Big)^2 \\ & = \xi_k\Big(2(1-\epsilon) - (1-\epsilon)^2\xi_k\Big)\|C_k\|^2 \\ & \ge \xi_k\Big(2(1-\epsilon) + (1-\epsilon)^2\Big)\|C_k\|^2 \\ & \ge (1-\epsilon^2)\xi_k\|C_k\|^2. \end{aligned}$$

We need to consider two cases. If $\xi_k = 1$ then

$$\|C_k\|^2 - \|C_y(x_k)(s_k^{\mathsf{q}})_y + C_k\|^2 \ge (1-\epsilon^2)\|C_k\| \min\{\|C_k\|, \delta_k\}.$$

Otherwise, it follows from (6.29) that

$$\xi_k = \frac{\delta_k}{\|(s_k^{\mathsf{q}})_y\|} \ge \frac{\delta_k}{(1+\epsilon)\|C_y(x_k)^{-1}\|\,\|C_k\|} \ge \frac{\delta_k}{(1+\epsilon)\nu_6\,\|C_k\|}.$$

In this case we get

$$\begin{aligned} \|C_k\|^2 - \|C_y(x_k)(s_k^{\mathsf{q}})_y + C_k\|^2 & \ge \frac{1-\epsilon^2}{(1+\epsilon)\nu_6}\|C_k\|\delta_k \\ & \ge \frac{1-\epsilon^2}{(1+\epsilon)\nu_6}\|C_k\| \min\{\|C_k\|, \delta_k\}. \end{aligned}$$

Thus the result holds with $\kappa_2 = (1-\epsilon^2)\min\{1, \frac{1}{(1+\epsilon)\nu_6}\}$ and $\kappa_3 = 1$. $\square$

6.4 Inexact Calculation of the Tangential Component

Ideally, the tangential component minimizes the quadratic model $\Psi_k(s_u)$ in the null space $\mathcal{N}(J_k)$ subject to the trust region and the bound constraints. Since the null space of J_k is characterized by W_k, the exact tangential component has the form $s_k^{\mathsf{t}} = W_k(s_k)_u$. If the u component of the tangential component is computed by a conjugate–gradient algorithm, its computation requires the calculation of matrix–vector products $W_k d_u$ and $W_k^T d$. We assume that these calculations are inexact.

6.4.1 Reduced Gradient

For the computation of the tangential component, we first have to compute the reduced gradient $W_k^T \nabla q_k(s_k^{\mathsf{q}})$ of the quadratic model $\Psi_k(s_u)$. If this is done using (6.5), then we have an approximation to $W_k^T \nabla q_k(s_k^{\mathsf{q}})$ of the form

$$W_k^T \nabla q_k(s_k^{\mathsf{q}}) + e_A, \tag{6.30}$$

where the error term e_A depends on $W_k^T \nabla q_k(s_k^{\mathsf{q}})$. By bounding the error term in (6.6), we find that

$$\|e_A\| \le \|C_u(x_k)^T C_y(x_k)^{-T}\| \, \|(e_A)_y\| + \|(e_A)_u\|. \tag{6.31}$$

We can interpret the inexact computation of $W_k^T \nabla q_k(s_k^{\mathsf{q}})$ as the exact solution of a perturbed equation. If we set

$$E_A = \frac{1}{\|\nabla_y q_k(s_k^{\mathsf{q}})\|^2} \, e_A \, (\nabla_y q_k(s_k^{\mathsf{q}}))^T,$$

then

$$\Big(- C_u(x_k)^T C_y(x_k)^{-T} + E_A \Big) \nabla_y q_k(s_k^{\mathsf{q}}) = -C_u(x_k)^T C_y(x_k)^{-T} \nabla_y q_k(s_k^{\mathsf{q}}) + e_A.$$

Thus we can define $A_k = C_y(x_k)^{-1} C_u(x_k) - E_A^T$ and

$$P_k = \begin{pmatrix} -A_k^T \\ I_{n-m} \end{pmatrix} = \begin{pmatrix} -C_y(x_k)^{-1} C_u(x_k) + E_A^T \\ I_{n-m} \end{pmatrix}. \tag{6.32}$$

With this definition we can write

$$W_k^T \nabla q_k(s_k^{\mathsf{q}}) + e_A = P_k^T \nabla q_k(s_k^{\mathsf{q}}).$$

The linear operator A_k satisfies

$$\begin{aligned} \| -A_k \ + \ C_u(x_k)^T C_y(x_k)^{-T} \| &= \|E_A^T\| \ \le \ \|e_A\| / \|\nabla_y q_k(s_k^{\mathsf{q}})\| \\ &\le \ \Big(\|C_u(x_k)^T C_y(x_k)^{-T}\| \, \|(e_A)_y\| + \|(e_A)_u\| \Big) / \|\nabla_y q_k(s_k^{\mathsf{q}})\| \end{aligned} \tag{6.33}$$

and

$$\begin{aligned} \Big\| (-A_k \ + \ C_u(x_k)^T C_y(x_k)^{-T}) \nabla_y q_k(s_k^{\mathsf{q}}) \Big\| \ &= \ \|E_A^T \nabla_y q_k(s_k^{\mathsf{q}})\| \ = \ \|e_A\| \\ &\le \ \Big(\|C_u(x_k)^T C_y(x_k)^{-T}\| \ \|(e_A)_y\| + \|(e_A)_u\| \Big). \end{aligned} \tag{6.34}$$

If for a given $\nabla_y q_k(s_k^{\mathsf{q}})$ the error terms in the computation of the reduced gradient via (6.5) obey

$$\max\Big\{ \|(e_A)_y\|, \ \|(e_A)_u\| \Big\} \ \le \ \epsilon \, \|C_k\|, \tag{6.35}$$

with $\epsilon > 0$, then (6.34) and Assumptions 5.3–5.5 in Section 5.2.5 imply Assumption 6.3. Moreover, if

$$\max\Big\{ \|(e_A)_y\|, \ \|(e_A)_u\| \Big\} \ \le \ \epsilon \|\nabla_y q_k(s_k^{\mathsf{q}})\|, \tag{6.36}$$

then (6.33) and Assumptions 5.3–5.5 in Section 5.2.5 imply the boundedness of $\{A_k\}$. This gives the first part of Assumption 6.1.

6.4.2 Use of Conjugate Gradients to Compute the Tangential Component

In the following, we formulate extensions of the Conjugate–Gradient Algorithms 5.7.1 and 5.7.2 for the computation of the tangential component. To keep the presentation simple, we continue to use the notation W_k and W_k^T. However, whenever matrix–vector products with W_k or W_k^T are computed, we assume that this is done using (6.3), or (6.5). The degree of inexactness, i.e. the size of the error terms e_y and e_u, is specified later. The reduced gradient $W_k^T \nabla q_k(s_k^{\mathsf{q}})$ of the quadratic model $\Psi_k(s_u)$ is assumed to be computed by (6.30) with errors $(e_A)_y$ and $(e_A)_u$ satisfying (6.35) and (6.36).

In the case where an approximation $\tilde{H}_k$ to the reduced Hessian $W_k^T \nabla^2_{xx} \ell_k W_k$ is used, the quadratic

$$-\Big(P_k^T \nabla q_k(s_k^{\mathsf{q}}) \Big)^T s_u - \frac{1}{2} s_u^T \tilde{H}_k s_u - \frac{1}{2} s_u^T E_k (\bar{D}_k^{\mathsf{P}})^{-2} s_u$$

is reduced at every iteration of the conjugate–gradient algorithm. If we use an approximation H_k to the full Hessian $\nabla^2_{xx} \ell_k$ we have to compute matrix–vector multiplications with $W_k^T H_k W_k$. One of the consequences of the inexactness is that the quadratic evaluated at the iterates of the conjugate–gradient algorithms is not guaranteed to decrease. For instance, the inexact application of W_k and W_k^T may cause $W_k^T H_k W_k$ to be nonsymmetric. Hence we need to measure the Cauchy decrease after the final iteration of the conjugate–gradient algorithm.

The extension of the Conjugate–Gradient Algorithm 5.7.1 is given below.

Algorithm 6.4.1 *(Inexact Computation of $s_k = s_k^{\mathsf{q}} + W_k(s_k)_u$ (Decoupled Case))*

1 Set $s_u^0 = 0$, $r^0 = -P_k^T \nabla q_k(s_k^{\mathsf{q}})$, $q^0 = (\bar{D}_k^{\mathsf{P}})^2 r^0$, $d^0 = q^0$, and $\epsilon > 0$.

2 For $i = 0, 1, 2, \ldots$ do

2.1 Compute

$$\gamma^i = \begin{cases} \frac{(r^i)^T(q^i)}{(d^i)^T(\tilde{H}_k + E_k(\bar{D}_k^{\mathsf{P}})^{-2})(d^i)} & \text{(reduced Hessian)}, \\ \frac{(r^i)^T(q^i)}{(d^i)^T(W_k^T H_k W_k + E_k(\bar{D}_k^{\mathsf{P}})^{-2})(d^i)} & \text{(full Hessian)}. \end{cases}$$

2.2 Compute

$$\tau^i = \max\Big\{\tau > 0 \quad : \quad \|(\bar{D}_k^{\mathsf{P}})^{-1}(s_u^i + \tau d^i)\| \le \delta_k, \\ \sigma_k(a - u_k) \le s_u^i + \tau d^i \le \sigma_k(b - u_k)\Big\}.$$

2.3 If $\gamma^i \le 0$, or if $\gamma^i > \tau^i$, then set $s_u^* = s_u^i + \tau^i d^i$, where τ^i is given as in 2.2 and go to 3; otherwise set $s_u^{i+1} = s_u^i + \gamma^i d^i$.

2.4 Update the residuals: $r^{i+1} =$

$$\begin{cases} r^i - \gamma^i(\tilde{H}_k + E_k(\bar{D}_k^{\mathsf{P}})^{-2})d^i & \text{(reduced Hessian)}, \\ r^i - \gamma^i(W_k^T H_k W_k + E_k(\bar{D}_k^{\mathsf{P}})^{-2})d^i & \text{(full Hessian)}, \end{cases}$$

and $q^{i+1} = (\bar{D}_k^{\mathsf{P}})^2 r^{i+1}$.

2.5 Check truncation criteria: if $\sqrt{\frac{(r^{i+1})^T(q^{i+1})}{(r^0)^T(q^0)}} \le \epsilon$, set $s_u^* = s_u^{i+1}$ and go to 3.

2.6 Compute $\alpha^i = \frac{(r^{i+1})^T(q^{i+1})}{(r^i)^T(q^i)}$ and set $d^{i+1} = q^{i+1} + \alpha^i d^i$.

3 Compute $W_k s_u^*$.

If a reduced Hessian approximation is used, set $(s_k)_u = s_u^*$ and $s_k = s_k^{\mathsf{q}} + W_k s_u^*$.

If a full Hessian approximation is used and if

$$\begin{aligned} -\left(P_k^T \nabla q_k(s_k^{\mathsf{q}})\right)^T s_u^* &- \tfrac{1}{2}(W_k s_u^*)^T H_k (W_k s_u^*) \\ &< -\left(W_k^T \nabla q_k(s_k^{\mathsf{q}})\right)^T s_u^1 - \tfrac{1}{2} {s_u^1}^T W_k^T H_k W_k s_u^1, \end{aligned}$$

then set $(s_k)_u = s_u^1$ and $s_k = s_k^{\mathsf{q}} + W_k s_u^1$. Otherwise $(s_k)_u = s_u^*$ and $s_k = s_k^{\mathsf{q}} + W_k s_u^*$.

The extension for the coupled approach is analogous and is omitted.

6.4.3 Distance to the Null Space of the Linearized Constraints

Let $(s_k^{\mathsf{t}})_y$ and $(s_k^{\mathsf{t}})_u = (s_k)_u$ be the quantities computed by Algorithm 6.4.1. Since $W_k(s_k)_u$ is not computed exactly in Step 3, it holds that

$$\begin{aligned} (s_k^{\mathsf{t}})_y &= -C_y(x_k)^{-1} C_u(x_k)(s_k)_u + C_y(x_k)^{-1}\left((e_B)_u + (e_B)_y\right) \\ &= -C_y(x_k)^{-1} C_u(x_k)(s_k)_u + e_B, \end{aligned}$$

where the error term e_B depends on $(s_k)_u$ and satisfies

$$\|e_B\| \le \|C_y(x_k)^{-1}\| \Big(\|(e_B)_u\| + \|(e_B)_y\| \Big), \tag{6.37}$$

cf. (6.4). As before, we can interpret the inexact computation $(s_k^{\mathsf{t}})_y$ of $s_k^{\mathsf{t}} = W_k(s_k)_u$ as the exact solution of a perturbed equation. If

$$E_B = \frac{1}{\|(s_k)_u\|^2}\, e_B\, (s_k)_u^T,$$

then

$$\Big(- C_y(x_k)^{-1} C_u(x_k) + E_B \Big)(s_k)_u = -C_y(x_k)^{-1} C_u(x_k)(s_k)_u + e_B = (s_k^{\mathsf{t}})_y.$$

We define $B_k = C_y(x_k)^{-1} C_u(x_k) - E_B$ and

$$Q_k = \begin{pmatrix} -B_k \\ I_{n-m} \end{pmatrix} = \begin{pmatrix} -C_y(x_k)^{-1} C_u(x_k) + E_B \\ I_{n-m} \end{pmatrix}. \tag{6.38}$$

With this definition, we can write

$$s_k^{\mathsf{t}} = Q_k (s_k)_u.$$

The linear operator B_k satisfies

$$\begin{aligned} \| - B_k + C_y(x_k)^{-1}C_u(x_k)\| &= \|E_B\| \le \|e_B\|/\|(s_k)_u\| \\ &\le \Big(\|C_y(x_k)^{-1}\| \left(\|(e_B)_u\| + \|(e_B)_y\|\right)\Big)/\|(s_k)_u\| \end{aligned} \tag{6.39}$$

and

$$\begin{aligned} \Big\|(- C_y(x_k)B_k + C_u(x_k))(s_k)_u\Big\| &= \|C_y(x_k)E_B(s_k)_u\| = \|C_y(x_k)e_B\| \\ &\le \|(e_B)_u\| + \|(e_B)_y\|. \end{aligned} \tag{6.40}$$

If the error terms in the computation of $(s_k^{\mathsf{t}})_y$ using (6.3) obey

$$\max\Big\{\|(e_B)_y\|,\ \|(e_B)_u\|\Big\} \le \frac{1}{2}\min\left\{\frac{1}{\kappa_3}, \frac{\kappa_2}{2}\right\} \min\left\{\kappa_3\|C_k\|, \delta_k\right\}, \tag{6.41}$$

where κ_2 and κ_3 are defined in (5.13), then one can see from (6.40) that B_k satisfies Assumption 6.2. Moreover, if

$$\max\Big\{\|e_y\|,\ \|e_u\|\Big\} \le \epsilon\|(s_k)_u\|, \tag{6.42}$$

then (6.39) and the boundedness of $\{C_y(x_k)^{-1}\}$ assured by Assumption 5.5 in Section 5.2.5, imply the boundedness of $\{B_k\}$ required in Assumption 6.1.

6.4.4 Fraction of Cauchy Decrease Condition

Now we establish the decrease condition (6.12). We analyze reduced and full Hessians approximations separately.

Reduced Hessian Approximation

In this case an approximation $\tilde{H}_k$ for $W_k^T\nabla^2_{xx}\ell_k W_k$ is used and all the calculations of Step 2 of Algorithm 6.4.1 are performed exactly. Hence $(s_k)_u$ satisfies the following condition

$$\begin{aligned} -\left(P_k^T\nabla q_k(s_k^{\mathsf{q}})\right)^T (s_k)_u &- \tfrac{1}{2}(s_k)_u^T\tilde{H}_k(s_k)_u \\ &\ge \kappa_6\|\bar{D}_k^{\mathsf{P}}P_k^T\nabla q_k(s_k^{\mathsf{q}})\|\min\Big\{\kappa_7\|\bar{D}_k^{\mathsf{P}}P_k^T\nabla q_k(s_k^{\mathsf{q}})\|,\ \kappa_8\delta_k\Big\}, \end{aligned} \tag{6.43}$$

for some positive constants κ_6, κ_7, and κ_8 independent of k. This is just an application of Lemma 5.3.3.

Now recall that we need to establish (6.12), where the left hand side is given by

$$-\left(Q_k^T \nabla q_k(s_k^{\mathsf{q}})\right)^T (s_k)_u - \frac{1}{2}(s_k)_u^T Q_k^T H_k Q_k (s_k)_u.$$

However, in (6.43) the left hand side is

$$-\left(P_k^T \nabla q_k(s_k^{\mathsf{q}})\right)^T (s_k)_u - \frac{1}{2}(s_k)_u^T \tilde{H}_k (s_k)_u.$$

First we use the expression (5.32) for H_k and the form (6.38) of Q_k to write

$$\frac{1}{2}(s_k)_u^T \tilde{H}_k (s_k)_u = \frac{1}{2}(s_k)_u^T Q_k^T H_k Q_k (s_k)_u.$$

Then we relate the inexactness represented by P_k and Q_k with the constraint residual $\|C_k\|$. First,

$$\begin{aligned} -\left(P_k^T \nabla q_k(s_k^{\mathsf{q}})\right)^T (s_k)_u &= -\nabla q_k(s_k^{\mathsf{q}})^T Q_k (s_k)_u - \nabla q_k(s_k^{\mathsf{q}})^T \begin{pmatrix} E_A^T \\ 0 \end{pmatrix} (s_k)_u \\ &\quad + \nabla q_k(s_k^{\mathsf{q}})^T \begin{pmatrix} E_B \\ 0 \end{pmatrix} (s_k)_u \\ &= -\left(Q_k^T \nabla q_k(s_k^{\mathsf{q}})\right)^T (s_k)_u - e_A^T (s_k)_u + e_B^T \nabla_y q_k(s_k^{\mathsf{q}}). \end{aligned}$$

The error bounds (6.31), (6.35), (6.37), (6.41), and Assumptions 5.3–5.5 in Section 5.2.5 give

$$e_A^T (s_k)_u - e_B^T \nabla_y q_k(s_k^{\mathsf{q}}) \;\leq\; \|e_A\| \, \|(s_k)_u\| + \|e_B\| \, \|\nabla_y q_k(s_k^{\mathsf{q}})\| \;\leq\; \varsigma_4' \|C_k\|, \qquad (6.44)$$

where ς_4' is a positive constant independent of k. Hence we proved (6.12) with $\varsigma_4 = \varsigma_4'$.

Full Hessian Approximation

The Cauchy step s_u^1 computed in the first iteration of Algorithm 6.4.1 satisfies

$$\begin{aligned} -\left(P_k^T \nabla q_k(s_k^{\mathsf{q}})\right)^T s_u^1 \;&-\; \tfrac{1}{2} {s_u^1}^T \widetilde{W}_k^T H_k \widehat{W}_k s_u^1 \\ &\geq\; \kappa_6 \|\bar{D}_k^{\mathsf{P}} P_k^T \nabla q_k(s_k^{\mathsf{q}})\| \min\left\{\kappa_7 \|\bar{D}_k^{\mathsf{P}} P_k^T \nabla q_k(s_k^{\mathsf{q}})\|, \; \kappa_8 \delta_k\right\}, \end{aligned} \qquad (6.45)$$

where the operators $\widehat{W}_k$ and $\widetilde{W}_k$ represent the inexact calculation $\widetilde{W}_k^T H_k \widehat{W}_k d^0$ of $W_k^T H_k W_k d^0$. Again, this is just an application of Lemma 5.3.3.

Let us assume first that $(s_k)_u = s_u^1$. We deal with $-(s_k)_u^T \widetilde{W}_k^T H_k \widehat{W}_k (s_k)_u$ using arguments similar to those used to obtain (6.44). We can show that

$$\begin{aligned}
\frac{1}{2}(s_k)_u^T Q_k^T H_k Q_k (s_k)_u \quad - \quad & \frac{1}{2}(s_k)_u^T \widetilde{W}_k^T H_k \widehat{W}_k (s_k)_u \\
= \quad & \frac{1}{2}(s_k)_u^T \left(W_k + \begin{pmatrix} E_B^T \\ 0 \end{pmatrix} \right)^T H_k \left(W_k + \begin{pmatrix} E_B \\ 0 \end{pmatrix} \right) (s_k)_u \\
& - \frac{1}{2}(s_k)_u^T \left(W_k + \begin{pmatrix} E_{\widetilde{A}}^T \\ 0 \end{pmatrix} \right)^T H_k \left(W_k + \begin{pmatrix} E_{\widehat{B}} \\ 0 \end{pmatrix} \right) (s_k)_u \\
\geq \quad & -\varsigma_4'' \, \|C_k\|. \qquad (6.46)
\end{aligned}$$

An explanation is in order. $E_{\widetilde{A}}$ and $E_{\widehat{B}}$ are constructed as E_A and E_B, respectively. The operator $E_{\widehat{B}}$ is the error matrix that is involved in computing $W_k d^0$. The operator $E_{\widetilde{A}}$ accounts for the error in computing $W_k^T \left(H_k \widehat{W}_k d^0 \right)$. We can force the residuals of these computations to depend on $\|C_k\|$ as in (6.35) and (6.41). From this and Assumptions 5.3–5.5 in Section 5.2.5, we get (6.46) with ς_4'' positive and independent of k. So, in the case $(s_k)_u = s_u^1$, we combine (6.44) and (6.46) to obtain (6.12) with $\varsigma_4 = \varsigma_4' + \varsigma_4''$.

If $(s_k)_u \neq s_u^1$, then from Step 3 of Algorithm 6.4.1 we see that $(s_k)_u$ satisfies

$$\begin{aligned}
-\left(P_k^T \nabla q_k(s_k^{\mathsf{q}}) \right)^T (s_k)_u \quad - \quad & \tfrac{1}{2}(\overline{W}_k (s_k)_u)^T H_k (\overline{W}_k (s_k)_u) \\
\geq \quad & -\left(P_k^T \nabla q_k(s_k^{\mathsf{q}}) \right)^T s_u^1 - \tfrac{1}{2} {s_u^1}^T \widetilde{W}_k^T H_k \widehat{W}_k s_u^1 \\
\geq \quad & \kappa_6 \| \bar{D}_k^{\mathsf{P}} P_k^T \nabla q_k(s_k^{\mathsf{q}}) \| \min \left\{ \kappa_7 \| \bar{D}_k^{\mathsf{P}} P_k^T \nabla q_k(s_k^{\mathsf{q}}) \|, \ \kappa_8 \delta_k \right\}.
\end{aligned}$$

Now we follow the same arguments used to establish (6.44) and (6.46). If the residual in $\overline{W}_k (s_k)_u$ is bounded by $\|C_k\|$, we obtain

$$\frac{1}{2}(s_k)_u^T Q_k^T H_k Q_k (s_k)_u - \frac{1}{2}(\overline{W}_k (s_k)_u)^T H_k (\overline{W}_k (s_k)_u) \geq -\varsigma_4''' \, \|C_k\|,$$

with ς_4''' a positive constant independent of k. Finally, if we use this and (6.44), we obtain (6.12) with $\varsigma_4 = \varsigma_4' + \varsigma_4'''$.

6.4.5 Inexact Calculation of Lagrange Multipliers

Note that the only assumption on λ_k required to prove the global convergence result (6.2.1) is the boundedness of the sequence $\{\lambda_k\}$ (see Assumption 5.4 in Section 5.2.5).

A choice of λ_k that is available from the reduced–gradient calculation of $q_k(s)$ is

$$\lambda_k = -C_y(x_k)^{-T}\nabla_y q_k(s_k^{\mathsf{q}}). \tag{6.47}$$

Due to inexactness λ_k actually satisfies

$$-C_y(x_k)^T\lambda_k = \nabla_y q_k(s_k^{\mathsf{q}}) + e_k,$$

where e_k is the corresponding residual vector. From Assumptions 5.3–5.5 in Section 5.2.5, if $\{e_k\}$ is bounded then $\{\lambda_k\}$ is also bounded.

Another choice for λ_k is

$$\lambda_k = -C_y(x_k)^{-T}\nabla_y f_k. \tag{6.48}$$

See Section 5.7.2 for a discussion on the choices (6.47) and (6.48) of λ_k.

6.5 Numerical Experiments

We tested the TRIP Reduced SQP Algorithms 5.2.1 with inexact solutions of linearized state and adjoint equations. The implementation is described in [76]. The numerical test computations were done on a Sun Sparcstation 10 in double precision `Fortran 77`. We solved the two examples described in Section 4.5 with a regularization parameter $\gamma = 10^{-3}$. The numerical results are satisfactory and revealed interesting properties of these algorithms.

We used the formula (5.90) to compute the quasi–normal component, and conjugate gradients with a tolerance $\epsilon = 10^{-4}$ to calculate the tangential component. In both cases all the linear systems of the form (6.1) are solved inexactly with the tolerances given below. The Hessian and reduced Hessian approximations, the scheme used to update the trust radius and the penalty parameter, and the inexact form D_k^{P} and $\bar{D}_k^{\mathsf{P}}$ of the affine scaling matrices D_k and $\bar{D}_k$ are the same as in Section 5.8. We used also $\sigma_k = \sigma = 0.99995$ for all k. The stopping criterion we used is (5.97), where $W_k^T\nabla f_k$ is calculated inexactly with the tolerance (6.50) given below.

The tolerance for inexact solvers with $C_y(x_k)$ was set to

$$\min\left\{10^{-2},\, 10^{-2}\min\{\|C_k\|,\, \delta_k\}\right\}, \tag{6.49}$$

and for inexact solvers with $C_y(x_k)^T$ to

$$\min\left\{10^{-2},\, 10^{-2}\|C_k\|\right\}. \tag{6.50}$$

This scheme for setting the tolerances satisfies our theoretical requirement (6.2).

6.5.1 Boundary Control Problem

The matrix $C_y(x)$ for the boundary nonlinear parabolic control problem with the discretization scheme mentioned in Section 4.5.1 is a block bidiagonal matrix with nonsymmetric tridiagonal blocks. In the exact (except for round off errors) implementation, we used the `LINPACK` subroutine `DGTSL` to solve the tridiagonal systems. These calculations are reported in Section 5.8 and summarized in the first line of Table 6.1 containing number of iterations. We introduce inexactness into this problem by solving these tridiagonal systems inexactly. For this purpose we tested several iterative methods like GMRES, QMR, and BiCGSTAB from the library [3]. The results are quite similar and we report here those obtained with GMRES(10)**. Since we have to solve a nonsymmetric tridiagonal system at each time step, we require the residual norms for these systems to be smaller than the tolerances given in (6.49) and (6.50) divided by the number of time steps N_t. The size, the functions, the starting vector, and the lower and upper bounds for this example are described in Section 5.8.

We ran the exact and inexact TRIP reduced SQP algorithms using decoupled and coupled approaches and reduced and full Hessians. The total number of iterations for each case is given in Table 6.1. There were no rejected steps. The objective function $f(x)$, the norm of the constraint residual $\|C(x)\|$, and the norm of the scaled reduced gradient $\|D(x)W(x)^T\nabla f(x)\|$ are plotted in Figure 6.1. In all the cases the algorithms took less then fifty iterations to attain the stopping criterion. The coupled approach did not perform as well as the decoupled approach. This is explained by the accumulation of errors due to inexactness. In fact, if the decoupled approach is used,

Optimal control problem governed by	Decoupled		Coupled	
	Reduced $\tilde{H}_k$	Full H_k	Reduced $\tilde{H}_k$	Full H_k
heat equation (exact solvers)	16	18	17	19
heat equation (inexact solvers)	16	18	29	48
semi-linear elliptic equation	18	20	27	36(39)

Table 6.1 Number of iterations to solve the optimal control problems.

**In GMRES(p), the number p denotes the dimension of the Krylov basis $\mathcal{K}_p$. See Section 6.3.2.

the y part of the tangential component $s_k^{\mathsf{t}} = W_k(s_k)_u$ is computed only in Step 3 of Algorithm 6.4.1, and although this computation is inexact, there is no accumulation of errors. In the coupled approach, the y part of the tangential component s_k^{t} of the step is updated at every conjugate–gradient iteration through an inexact linearized state solver. This destroys the symmetry of the subproblem and the conjugate–gradient algorithm requires more iterations. As the number of conjugate–gradient iterations increases, this error propagates, and the steps that are computed are farther away from the null space of the Jacobian J_k.

We illustrate this situation in Figure 6.2, where we show how far $\|J_k s_k^{\mathsf{q}}\|$ and $\|J_k(s_k^{\mathsf{q}} + s_k^{\mathsf{t}})\|$ are from each other. The dotted line shows the size of the residual of the linearized state equation after the computation of the quasi–normal component. If the tangential component is in the null space of the Jacobian, then this would be the size of the residual of the linearized state equation for the whole step. In other words, we would have

$$\|J_k s_k^{\mathsf{q}}\| = \|J_k(s_k^{\mathsf{q}} + s_k^{\mathsf{t}})\|.$$

However, due to the inexactness in the application of W_k and W_k^T, the size of the residual of the linearized state equation for the whole step is larger and is given by the solid line. It can be seen that the difference grows as W_k and W_k^T are applied more often in the computation of the tangential component. In particular, the difference is larger if the coupled approach is used.

6.5.2 Distributed Control Problem

For the distributed semi–linear control problem given in Section 4.5.2, we used $\Omega = (0,1)^2$, $d = 0$, $g(y) = e^y$, and $y_d = \sin(2\pi x_1)\sin(2\pi x_2)$. In this case the state equation (4.28) for $u = 0$ is the Bratu problem (4.30) with $\lambda = -1$. We used the discretization of this problem implemented by M. Heinkenschloss (ICAM, Virginia Polytechnic Institute and State University) with piecewise linear finite elements on a uniform triangulation obtained by first subdividing the x and the y subinterval into a sample of subintervals and then cutting each resulting subsquare into two triangles (see for instance [63]). The same discretization was used for the states and the controls.

The norms used for the states and controls are the discretizations of the $H^1(\Omega)$ and $L^2(\Omega)$ norms. The linearized state and adjoint equations are solved using GMRES(20) preconditioned from the left with the inverse Laplacian. To apply this preconditioner,

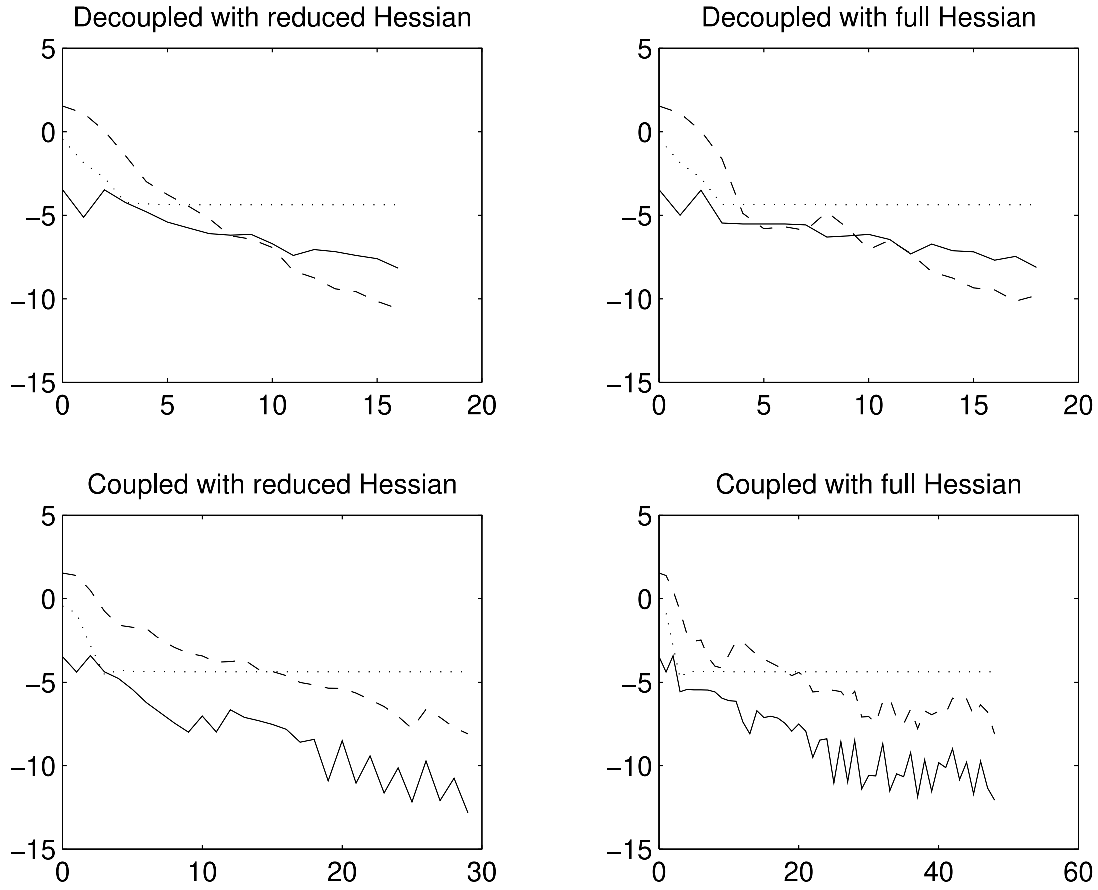

Figure 6.1 Performance of the inexact TRIP reduced SQP algorithms applied to the boundary control problem. Here $\ln_{10} f_k$ (dotted line), $\ln_{10} \|C_k\|$ (dashed line), and $\ln_{10} \|D_k W_k^T \nabla f_k\|$ (solid line) are plotted as a function of k.

one has to compute the solution of the discrete Laplace equation with different right hand sides. This was done using multilevel preconditioned conjugate gradients [151]. Note that for $g(y) = e^y$, the problem is self–adjoint. Therefore a conjugate–gradient algorithm could have been used instead of GMRES. However, the implementation was done for the more general problem with state equation $-\Delta y + g(y, \nabla y) = u$, which in general is not self–adjoint.

In this example, the number of control variables that comprises the components of u is equal to the number of state variables represented by components of y. In the computations reported below we use $m = n - m = 289$ which corresponds to a

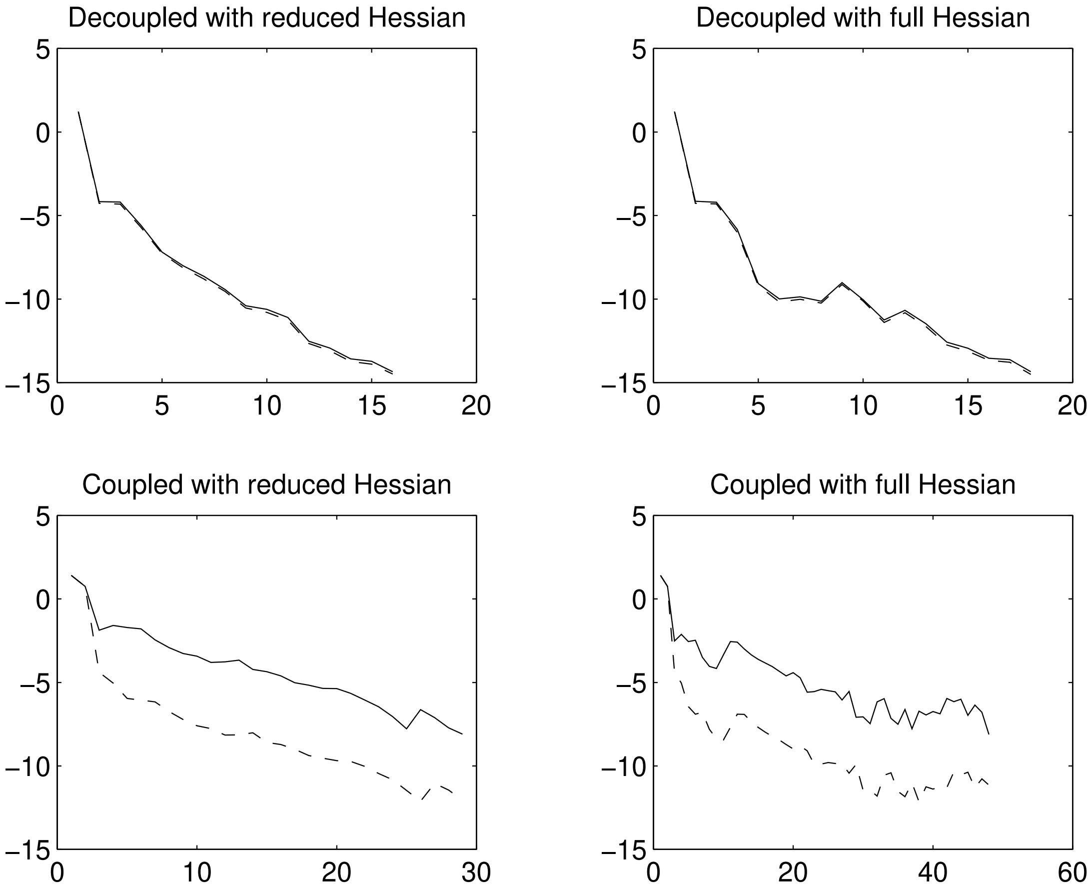

Figure 6.2 Illustration of the performance of the inexact TRIP reduced SQP algorithms applied to the boundary control problem. These plots show the residuals $\ln_{10}\|J_k s_k^{\mathsf{q}}\|$ in dashed line and $\ln_{10}\|J_k(s_k^{\mathsf{q}} + s_k^{\mathsf{t}})\|$ in solid line.

uniform triangulation with 512 triangles. The upper and lower bounds were $b_i = 5$, $a_i = -1000$, $i = 1, \ldots, n-m$. The starting vector was $x_0 = 0$.

The total numbers of iterations needed by the inexact TRIP reduced SQP algorithms to solve this problem are presented in Table 6.1. In all situations but one, all the steps were accepted. (The situation we refer to is the coupled approach with full Hessian approximation where there were 36 accepted steps among the 39 computed.) The objective function $f(x)$, the norm of the constraint residual $\|C(x)\|$, and the norm of the scaled reduced gradient $\|D(x)W(x)^T\nabla f(x)\|$ are plotted in Figure 6.3. The convergence behavior of the inexact TRIP reduced SQP algorithms is similar to the convergence behavior for the other example. Again the decoupled approach performs better than the coupled one due to the fact that less errors are accumulated. See Figure 6.4.

The last experiment that we report consisted of applying the inexact version of the TRIP Reduced SQP Algorithms 5.2.1 to solve large instances of the distributed semi–linear control problem. In this experiment we used the decoupled approach with a limited memory BFGS update to approximate the reduced Hessian matrix as described in Section 5.8. The number of iterations corresponding to four instances of this control problem are given in Table 6.2. These instances were generated by decreasing the mesh size, i.e. by increasing the number of triangles in the discretization. In this table we include the number of linearized state and adjoint equations of the form (6.1) solved by the algorithms.

We point out that in this example the control is distributed in Ω and the number of components in u is $\frac{n}{2}$. For the values $b_i = 5$, $a_i = -1000$, $i = 1, \ldots, n-m$ of the upper and lower bounds that we chose, the number of control variables u active at the solution is roughly equal to $\frac{n}{10}$. These observations are important for the conclusions we draw in the next paragraph.

It is well known that in many interior–point algorithms for linear and convex programming problems the number of iterations is a polynomial function of the size of the problem. On the other hand, most active set methods have a exponential worst–case complexity. In interior–point algorithms, as we increase the dimension of the problem we should observe at most a polynomial increase in the number of the iterations. We can see from Table 6.2 that this is clearly the case for the TRIP reduced SQP algorithms. These results once more show the effectiveness of these algorithms for optimal control problems with bound constraints on the controls. (If there are rejected steps, then the number of iterations in brackets corresponds to all the accepted and rejected iterations.)

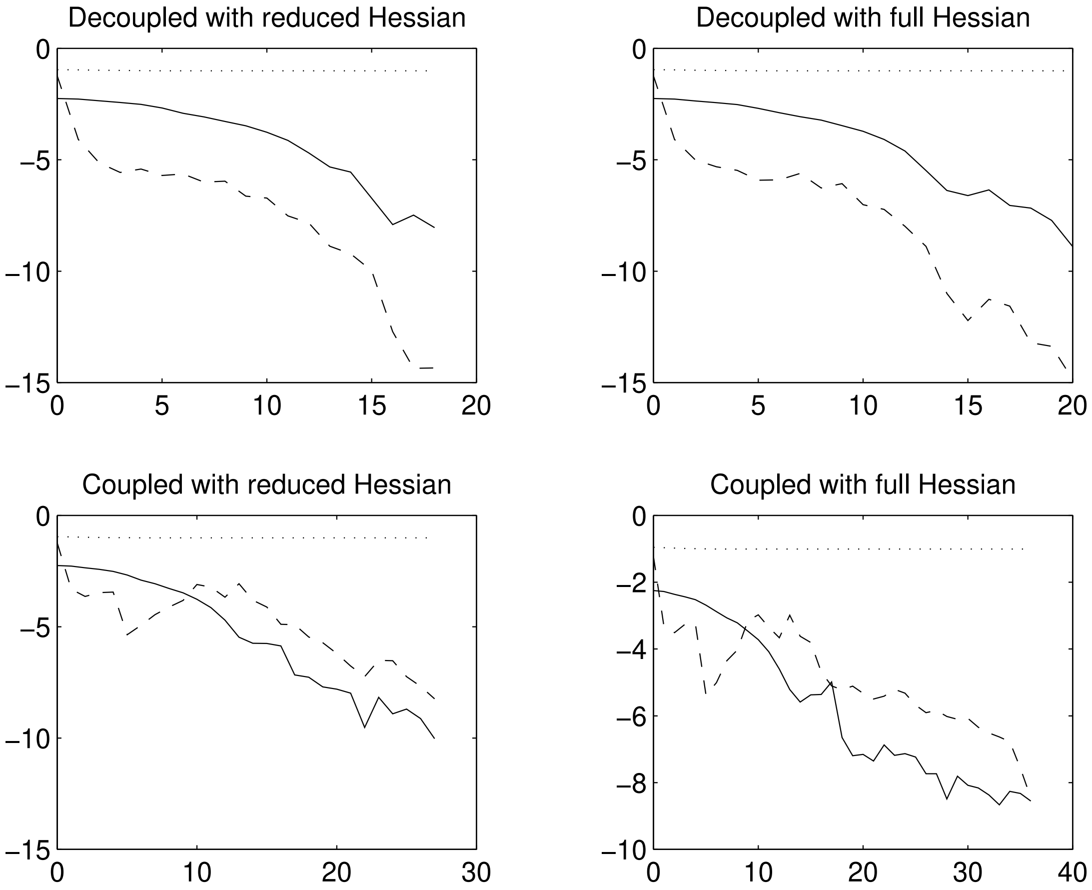

Figure 6.3 Performance of the inexact TRIP reduced SQP algorithms applied to the distributed control problem. Here $\ln_{10} f_k$ (dotted line), $\ln_{10} \|C_k\|$ (dashed line), and $\ln_{10} \|D_k W_k^T \nabla f_k\|$ (solid line) are plotted as a function of k.

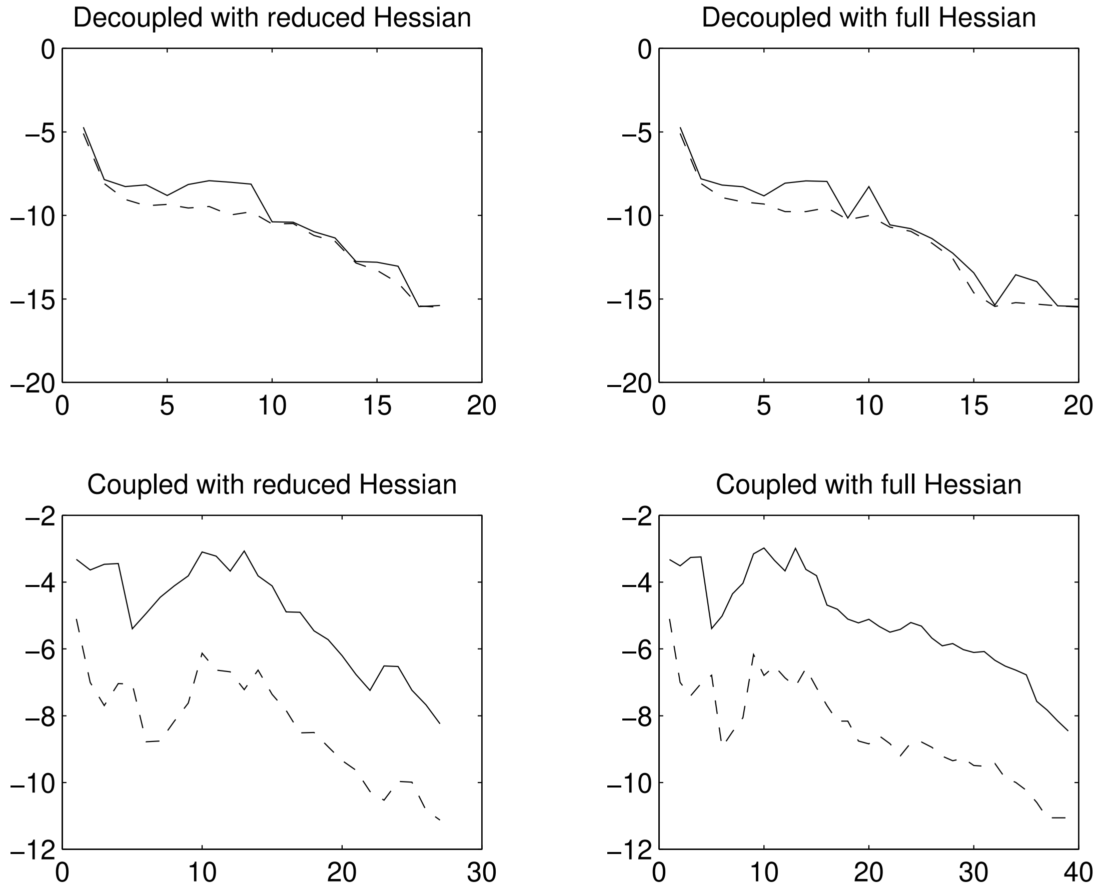

Figure 6.4 Illustration of the performance of the inexact TRIP reduced SQP algorithms applied to the distributed control problem. These plots show the residuals $\ln_{10} \|J_k s_k^{\mathsf{q}}\|$ in dashed line and $\ln_{10} \|J_k(s_k^{\mathsf{q}} + s_k^{\mathsf{t}})\|$ in solid line.

variables (n)	constraints (m)	iterations	$C_y(x_k)$ solvers	$C_y(x_k)^T$ solvers
578	289	18	54	37
2178	1089	22	66	45
8450	4225	26 (31)	83	58
33282	16641	49	147	99

Table 6.2 Number of iterations to solve large distributed semi–linear control problems.

Chapter 7

Conclusions and Open Questions

7.1 Conclusions

In this dissertation we introduced and analyzed trust–region interior–point (TRIP) reduced sequential quadratic programming (SQP) algorithms for an important class of nonlinear programming problems which appear in many engineering applications. These problems appear from the discretization of many optimal control, parameter identification, and inverse problems and consequently their equality constraints are often large discretized nonlinear partial differential equations. In Chapter 4 of this thesis, we described this class of problems in a great detail, analyzed the structure, and derived the optimality conditions.

The TRIP reduced SQP algorithms use the structure of the problem, and they combine trust–region techniques for equality–constrained optimization with a primal–dual affine scaling interior–point approach for simple bounds. We proved in Chapter 5 global and local convergence results for these algorithms that include as special cases both the results established for equality constraints [35], [42] and those for simple bounds [24]. (See Figures 1.1 and 1.2.) We are satisfied with the sharpness of the results. In Sections 5.4 and 5.5, we proved global convergence to a point satisfying the first and second order necessary optimality conditions for these algorithms by using assumptions that reduce to the weakest assumptions used to establish similar results in unconstrained, equality–constrained, and box–constrained optimization. Section 5.6 showed that the TRIP reduced SQP algorithms behave properly close to a point satisfying the second–order sufficient optimality conditions: the trust radius is uniformly bounded away from zero and the penalty parameter is uniformly bounded. This and the fact that the algorithms are Newton related allowed us to show a q–quadratic rate of convergence.

Chapter 6 investigated the theoretical behavior of this class of TRIP reduced SQP algorithms under the presence of inexactness in the solution of linear systems, such as the linearized state and adjoint equations, and in the computation of directional derivatives. The most important conclusion that we can derive from this analysis is that global convergence to a point satisfying the first–order necessary optimality conditions can be guaranteed if the absolute error in the solution of linear systems with $C_y(x_k)$ (linearized state equations) is $\mathcal{O}\left(\min\{\delta_k, \|C_k\|\}\right)$ and with $C_y(x_k)^T$ (adjoint equations) is $\mathcal{O}\left(\|C_k\|\right)$. We recall that δ_k is the trust radius and $\|C_k\|$ is the residual of the equality constraints, and that these quantities are known at the beginning of each iteration k.

We implemented the TRIP reduced SQP algorithms and included here results on two discretized nonlinear optimal control problems. The implementation covers several step computations and second–order approximations. The numerical results reported in Sections 5.8 and 6.5 were quite satisfactory and confirmed most of our theoretical findings. The software that produced these results currently is being beta–tested with the intent of electronic distribution [76].

Chapter 3 demonstrates global convergence to a point satisfying the second–order necessary optimality conditions for a family of trust–region algorithms for equality–constrained optimization and presents a detailed analysis of the trust–region subproblem for the linearized constraints. The important feature of this family of algorithms is that they do not require the computation of normal components for the step and an orthogonal basis for the null space of the Jacobian of the equality constraints.

7.2 Open Questions

The extension of the TRIP reduced SQP algorithms to handle bounds on the state variables y is probably the most important question that this dissertation leaves open. If lower and upper bounds of the form $c \le y \le d$, with c, $d \in \mathbb{R}^m$, are imposed in problem (4.1), then the condition (4.23) in the first–order necessary optimality conditions becomes

$$\left(\begin{array}{c} \nabla_y f(x_*) \\ \nabla_u f(x_*) \end{array}\right) + \left(\begin{array}{c} C_y(x_*)^T\lambda_* \\ C_u(x_*)^T\lambda_* \end{array}\right) - \left(\begin{array}{c} \mu_*^c \\ \mu_*^a \end{array}\right) + \left(\begin{array}{c} \mu_*^d \\ \mu_*^b \end{array}\right) = 0,$$

for some nonnegative multipliers $\mu_*^c, \mu_*^d \in \mathbb{R}^m$ satisfying the complementarity condition

$$\left((y_*)_i - c_i\right)(\mu_*^c)_i = \left(d_i - (y_*)_i\right)(\mu_*^d)_i = 0, \quad i = 1, \ldots, m.$$

(See Proposition 4.4.1.) A key point here is that now

$$\lambda_* = -C_y(x_*)^{-T}(\nabla_y f(x_*) - \mu_*^c + \mu_*^d)$$

and this dependence of λ_* on the unknown multipliers μ_*^c and μ_*^d is problematic. Of course this affects the extension of the primal–dual affine scaling strategy to cover the bound constraints on y. We have investigated this topic further, but up to this moment we have not reached any satisfactory answer. It is not at all clear for us that affine scaling strategies are the appropriate interior–point techniques to handle problems of the form (4.1) with bounds on controls u and states y. As a possible alternative for the affine scaling interior–point strategy, we have in mind the use of primal–dual interior–point algorithms. For general nonlinear programs of the form (4.20) these algorithms have been studied in [50], [98], [148] where they are referred also as Newton or quasi–Newton interior–point methods.

The formulation and analysis of the TRIP reduced SQP algorithms in an infinite dimensional framework is another research topic that deserves to be investigated.

Our implementation of the TRIP reduced SQP algorithms will be subject to many improvements. We have in mind for instance the computation of the quasi–normal and tangential components by adapting to our context the algorithms proposed in [129], [133]. Testing the effectiveness of the coupled approach for ill–conditioned problems is part of our future plans.

The conditions on the inexactness described in Chapter 6, and summarized in Section 6.2, are sufficient to guarantee global convergence to a point satisfying the first–order necessary optimality conditions. However, as it is the case for systems of nonlinear equations, the practical implementation of such conditions greatly influences the performance of the algorithms. Issues like oversolving and forcing faster rates of local convergence are of importance and should be the subject of future investigations. Since the quasi–normal component of the step can be viewed as one step of Newton's method (with a trust–region globalization) towards feasibility of $C(y, u) = 0$ for fixed u, there is a close relationship with the studies of inexact Newton methods for systems of nonlinear equations [44], [45]. The computation of the tangential component using the coupled approach is another issue that needs further investigation. In particular the loss of symmetry due to the inexactness and the use of nonsymmetric iterative methods for the solution of these subproblems deserves attention (see [101]).

Bibliography

[1] N. Alexandrov, *Multilevel Algorithms for Nonlinear Equations and Equality Constrained Optimization*, PhD thesis, Department of Computational and Applied Mathematics, Rice University, Houston, Texas 77251, USA, 1993. Tech. Rep. TR93–20.

[2] L. Armijo, *Minimization of functions having Lipschitz–continuous first partial derivatives*, Pacific J. Math., 16 (1966), pp. 1–3.

[3] R. Barret, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst, *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*, SIAM, Philadelphia, Pennsylvania, 1994.

[4] L. T. Biegler, J. Nocedal, and C. Schmid, *A reduced Hessian method for large–scale constrained optimization*, SIAM J. Optim., 5 (1995), pp. 314–347.

[5] P. T. Boggs, *Sequential quadratic programming*, in Acta Numerica 1995, A. Iserles, ed., Cambridge University Press, Cambridge, London, New York, 1995, pp. 1–51.

[6] P. T. Boggs, J. W. Tolle, and A. J. Kearsley, *A practical algorithm for general large scale nonlinear optimization problems*, Tech. Rep. NISTIR 5407, Computing and Applied Mathematics Laboratory, National Institute of Standards and Statistics, 1994. To appear in SIAM J. Optim.

[7] J. F. Bonnans and C. Pola, *A trust region interior point algorithm for linearly constrained optimization*, Tech. Rep. 1948, INRIA, 1993.

[8] M. A. Branch, T. F. Coleman, and Y. Li, *A subspace, interior, and conjugate gradient method for large–scale bound–constrained minimization problems*, Tech. Rep. CTC95TR217, Advancing Computing Research Institute, Cornell University, 1995.

[9] P. N. Brown, *A local convergence theory for combined inexact–Newton finite–difference projection methods*, SIAM J. Numer. Anal., 24 (1987), pp. 407–434.

[10] P. N. Brown and Y. Saad, *Hybrid Krylov methods for nonlinear systems of equations*, SIAM J. Sci. Statist. Comput., 11 (1990), pp. 450–481.

[11] ——, *Convergence theory of nonlinear Newton–Krylov algorithms*, SIAM J. Optim., 4 (1994), pp. 297–330.

[12] J. Burger and M. Pogu, *Functional and numerical solution of a control problem originating from heat transfer*, J. Optim. Theory Appl., 68 (1991), pp. 49–73.

[13] J. V. Burke, *A robust trust region method for constrained nonlinear programming problems*, SIAM J. Optim., 2 (1992), pp. 325–347.

[14] J. V. Burke, J. J. More, and G. Toraldo, *Convergence properties of trust region methods for linear and convex constraints*, Math. Programming, 47 (1990), pp. 305–336.

[15] R. H. Byrd, J. Nocedal, and R. B. Schnabel, *Representations of quasi–Newton matrices and their use in limited memory methods*, Math. Programming, 63 (1994), pp. 129–156.

[16] R. H. Byrd and R. B. Schnabel, *Continuity of the null space basis and constrained optimization*, Math. Programming, 35 (1986), pp. 32–41.

[17] R. H. Byrd, R. B. Schnabel, and G. A. Shultz, *A trust region algorithm for nonlinearly constrained optimization*, SIAM J. Numer. Anal., 24 (1987), pp. 1152–1170.

[18] ——, *Approximate solution of the trust region problem by minimization over two–dimensional subspaces*, Math. Programming, 40 (1988), pp. 247–263.

[19] R. G. Carter, *On the global convergence of trust region algorithms using inexact gradient information*, SIAM J. Numer. Anal., 28 (1991), pp. 251–265.

[20] A. Cauchy, *Méthode générale pour la résolution des systèmes d'équations simultanées*, Compte Rendu des Séances de L'Académie des Sciences XXV, (1847), pp. 536–538.

[21] M. Celis, J. E. Dennis, and R. A. Tapia, *A trust region strategy for nonlinear equality constrained optimization*, in Numerical Optimization 1984, SIAM, Philadelphia, Pennsylvania, 1985, pp. 71–82.

[22] E. M. Cliff, M. Heinkenschloss, and A. Shenoy, *An optimal control problem for flows with discontinuities*, J. Optim. Theory Appl., (1997), pp. (to appear, August).

[23] T. F. Coleman and Y. Li, *On the convergence of interior–reflective Newton methods for nonlinear minimization subject to bounds*, Math. Programming, 67 (1994), pp. 189–224.

[24] ——, *An interior trust region approach for nonlinear minimization subject to bounds*, SIAM J. Optim., 6 (1996), pp. 418–445.

[25] T. F. Coleman and J. Liu, *An interior Newton method for quadratic programming*, Tech. Rep. TR93–1388, Department of Computer Science, Cornell University, 1993.

[26] T. F. Coleman and D. C. Sorensen, *A note on the computation of an orthonormal basis for the null space of a matrix*, Math. Programming, 29 (1984), pp. 234–242.

[27] T. F. Coleman and W. Yuan, *A new trust region algorithm for equality constrained optimization*, Tech. Rep. TR95–1477, Department of Computer Science, Cornell University, 1995.

[28] A. R. Conn, N. I. M. Gould, A. Sartenaer, and P. L. Toint, *Global convergence of a class of trust region algorithms for optimization using inexact projections onto convex constraints*, SIAM J. Optim., 3 (1993), pp. 164–221.

[29] A. R. Conn, N. I. M. Gould, and P. L. Toint, *Global convergence of a class of trust region algorithms for optimization problems with simple bounds*, SIAM J. Numer. Anal., 25 (1988), pp. 433–460.

[30] ——, *A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds*, SIAM J. Numer. Anal., 28 (1991), pp. 545–572.

[31] E. J. Cramer, J. E. Dennis, P. D. Frank, R. M. Lewis, and G. R. Shubin, *Problem formulation for multidisciplinary optimization*, SIAM J. Optim., 4 (1994), pp. 754–776.

[32] R. S. Dembo, S. C. Eisenstat, and T. Steihaug, *Inexact Newton methods*, SIAM J. Numer. Anal., 19 (1982), pp. 400–408.

[33] R. S. Dembo and T. Steihaug, *Truncated–Newton algorithms for large–scale unconstrained optimization*, Math. Programming, 19 (1983), pp. 190–212.

[34] R. S. Dembo and U. Tulowitzki, *Sequential truncated quadratic programming*, in Numerical Optimization 1984, P. T. Boggs, R. H. Byrd, and R. B. Schnabel, eds., SIAM, Philadelphia, 1985, pp. 83–101.

[35] J. E. Dennis, M. El-Alem, and M. C. Maciel, *A global convergence theory for general trust–region–based algorithms for equality constrained optimization*, SIAM J. Optim., 7 (1997), pp. 177–207.

[36] J. E. Dennis, M. Heinkenschloss, and L. N. Vicente, *Trust–region interior–point SQP algorithms for a class of nonlinear programming problems*, Tech. Rep. TR94–45, Department of Computational and Applied Mathematics, Rice University, 1994. Revised November 1995. Appeared also as Tech. Rep. 94–12–01, Interdisciplinary Center for Applied Mathematics, Virginia Polytechnic Institute and State University.

[37] J. E. Dennis and H. H. W. Mei, *Two new unconstrained optimization algorithms which use function and gradient values*, J. Optim. Theory Appl., 28 (1979), pp. 453–482.

[38] J. E. Dennis and J. J. Moré, *Quasi–Newton methods, motivation and theory*, SIAM Rev., 19 (1977), pp. 46–89.

[39] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice–Hall, Englewood Cliffs, New Jersey, 1983.

[40] ——, *A view of unconstrained optimization*, in Handbooks in Operations Research and Management Science, G. L. Nemhauser, A. H. G. R. Kan, and M. J. Todd, eds., North Holland, Amsterdam, 1988. (Vol. 1, Optimization).

[41] J. E. Dennis and L. N. Vicente, *Trust–region interior–point algorithms for minimization problems with simple bounds*, in Applied Mathematics and Parallel Computing, Festschrift for Klaus Ritter, H. Fisher, B. Riedmüller, and S. Schäffler, eds., Physica–Verlag, Springer–Verlag, 1996, pp. 97–107.

[42] ——, *On the convergence theory of general trust–region–based algorithms for equality–constrained optimization*, SIAM J. Optim., (To appear).

[43] P. Deuflhard, *Global inexact Newton methods for very large scale nonlinear problems*, Impact of Computing in Science and Engineering, 4 (1991), pp. 366–393.

[44] S. C. Eisenstat and H. F. Walker, *Globally convergent inexact Newton methods*, SIAM J. Optim., 4 (1994), pp. 393–422.

[45] ——, *Choosing the forcing terms in an inexact Newton method*, SIAM J. Sci. Statist. Comput., 17 (1996), pp. 16–32.

[46] M. El-Alem, *A Global Gonvergence Theory for a Class of Trust Region Algorithms for Constrained Optimization*, PhD thesis, Department of Computational and Applied Mathematics, Rice University, Houston, Texas 77251, USA, 1988. Tech. Rep. TR88–5.

[47] ——, *A global convergence theory for the Celis–Dennis–Tapia trust–region algorithm for constrained optimization*, SIAM J. Numer. Anal., 28 (1991), pp. 266–290.

[48] ——, *Convergence to a second–order point for a trust–region algorithm with a nonmonotonic penalty parameter for constrained optimization*, Tech. Rep. TR95–28, Department of Computational and Applied Mathematics, Rice University, 1995.

[49] ——, *A robust trust–region algorithm with a non–monotonic penalty parameter scheme for constrained optimization*, SIAM J. Optim., 5 (1995), pp. 348–378.

[50] A. S. El-Bakry, R. A. Tapia, T. Tsuchiya, and Y. Zhang, *On the formulation and theory of the Newton interior–point method for nonlinear programming*, J. Optim. Theory Appl., 89 (1996), pp. 507–541.

[51] M. El-Hallabi, *A global convergence theory for arbitrary norm trust–region algorithms for equality constrained optimization*, Tech. Rep. TR93–60, Department of Computational and Applied Mathematics, Rice University, 1993. Revised May 1995.

[52] R. Fletcher, *An ℓ_1 penalty method for nonlinear constraints*, in Numerical Optimization 1984, P. T. Boggs, R. H. Byrd, and R. B. Schnabel, eds., SIAM, Philadelphia, 1985, pp. 26–40.

[53] ——, *Practical Methods of Optimization*, John Wiley & Sons, Chichester, second ed., 1987.

[54] R. Fontecilla, *On inexact quasi–Newton methods for constrained optimization*, in Numerical Optimization 1984, P. T. Boggs, R. H. Byrd, and R. B. Schnabel, eds., SIAM, Philadelphia, 1985, pp. 102–118.

[55] R. Freund, *A transpose–free quasi–minimal residual algorithm for non–Hermitian linear systems*, SIAM J. Sci. Statist. Comput., 14 (1993), pp. 470–482.

[56] D. M. Gay, *Computing optimal locally constrained steps*, SIAM J. Sci. Statist. Comput., 2 (1981), pp. 186–197.

[57] I. M. Gel'fand, *Some problems in the theory of quasilinear equations*, Amer. Math. Soc. Transl., 29 (1963), pp. 295–381.

[58] P. E. Gill, W. Murray, M. Saunders, G. W. Stewart, and M. H. Wright, *Properties of a representation of a basis for the null space*, Math. Programming, 33 (1985), pp. 172–186.

[59] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *User's guide for NPSOL (version 4.0): A FORTRAN package for nonlinear programming*, Technical Report SOL 86–2, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, CA 94305–4022, 1986.

[60] P. E. Gill, W. Murray, and M. H. Wright, *Practical Optimization*, Academic Press, INC., San Diego, 1981.

[61] ——, *Some theoretical properties of an augmented Lagrangian merit function*, Technical Report SOL 86–6, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, CA 94305–4022, 1986.

[62] R. Glowinski, *Numerical Methods for Nonlinear Variational Problems*, Springer–Verlag, Berlin, Heidelberg, New York, Tokyo, 1984.

[63] R. Glowinski, H. B. Keller, and L. Reinhart, *Continuation–conjugate gradient methods for the least–squares solution of nonlinear boundary value problems*, SIAM J. Sci. Statist. Comput., 6 (1985), pp. 793–832.

[64] S. Goldfeld, R. Quandt, and H. Trotter, *Maximization by quadratic hill climbing*, Econometrica, 34 (1966), pp. 541–551.

[65] A. A. Goldstein, *On steepest descent*, SIAM J. Control Optim., 3 (1965), pp. 147–151.

[66] G. H. Golub and C. F. Van Loan, *Matrix Computations*, The John Hopkins University Press, Baltimore and London, second ed., 1989.

[67] G. H. Golub and U. von Matt, *Quadratically constrained least squares and quadratic problems*, Numer. Math., 59 (1991), pp. 561–580.

[68] J. Goodman, *Newton's method for constrained optimization*, Math. Programming, 33 (1985), pp. 162–171.

[69] C. W. Groetsch, *Generalized Inverses of Linear Operators*, Marcel Dekker, Inc., New York, Basel, 1977.

[70] W. A. Gruver and E. W. Sachs, *Algorithmic Methods In Optimal Control*, Pitman, London, 1980.

[71] M. D. Hebden, *An algorithm for minimization using exact second order derivatives*, Tech. Rep. T.P. 515, Atomic Energy Research Establishment, Harwell, England, 1973.

[72] M. Heinkenschloss, *SQP methods for the solution of optimal control problems governed by the Navier Stokes equations.* In preparation.

[73] ——, *Krylov subspace methods for the solution of linear systems and linear least squares problems.* Lecture Notes, 1994.

[74] ——, *On the solution of a two ball trust region subproblem*, Math. Programming, 64 (1994), pp. 249–276.

[75] ——, *Projected sequential quadratic programming methods*, SIAM J. Optim., 6 (1996), pp. 373–417.

[76] M. Heinkenschloss and L. N. Vicente, *TRICE: A Package of Trust–Region Interior–Point SQP Algorithms for the Solution of Optimal Control and Enginnering Design Problems – User's Guide –*. In preparation.

[77] ——, *Analysis of inexact trust–region interior–point SQP algorithms*, Tech. Rep. TR95–18, Department of Computational and Applied Mathematics, Rice University, 1995. Revised April 1996. Appeared also as Tech. Rep. 95–06–01, Interdisciplinary Center for Applied Mathematics, Virginia Polytechnic Institute and State University.

[78] M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Standards, 49 (1952), pp. 409–436.

[79] K. Ito and K. Kunisch, *The augmented Lagrangian method for parameter estimation in elliptic systems*, SIAM J. Control Optim., 28 (1990), pp. 113–136.

[80] W. Karush, *Minima of Functions of Several Variables with Inequalities as Side Constraints*, Master's thesis, Department of Mathematics, University of Chicago, 1939.

[81] C. T. Kelley, *Iterative Methods for Linear and Nonlinear Equations*, SIAM, Philadelphia, Pennsylvania, 1995.

[82] C. T. Kelley and E. W. Sachs, *Solution of optimal control problems by a pointwise projected Newton method*, SIAM J. Control Optim., 33 (1995), pp. 1731–1757.

[83] C. T. Kelley and S. J. Wright, *Sequential quadratic programming for certain parameter identification problems*, Math. Programming, 51 (1991), pp. 281–305.

[84] H. W. Kuhn and A. W. Tucker, *Nonlinear programming*, in Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability, J. Neyman, ed., University of California Press, 1951.

[85] K. Kunisch and G. Peichl, *Estimation of a temporally and spatially varying diffusion coefficient in a parabolic system by an augmented Lagrangian technique*, Numer. Math., 59 (1991), pp. 473–509.

[86] K. Kunisch and E. Sachs, *Reduced SQP methods for parameter identification problems*, SIAM J. Numer. Anal., 29 (1992), pp. 1793–1820.

[87] F.-S. Kupfer, *Reduced Successive Quadratic Programming in Hilbert Space with Applications to Optimal Control*, PhD thesis, Universität Trier, Fb–IV, Mathematik, D–54286 Trier, Germany, 1992.

[88] F.-S. Kupfer and E. W. Sachs, *A prospective look at SQP methods for semilinear parabolic control problems*, in Optimal Control of Partial Differential Equations, Irsee 1990, K.-H. Hoffmann and W. Krabs, eds., vol. 149, Springer Lect. Notes in Control and Information Sciences, 1991, pp. 143–157.

[89] ——, *Numerical solution of a nonlinear parabolic control problem by a reduced SQP method*, Comput. Optim. and Appl., 1 (1992), pp. 113–135.

[90] J. L. Lagrange, *Oeuvres de Lagrange, Volumes XI and XII*, Gauthier–Villars, Paris, 1888–1889.

[91] M. Lalee, J. Nocedal, and T. Plantenga, *On the implementation of an algorithm for large–scale equality constrained optimization.* Submitted for publication, 1994.

[92] F. Leibfritz and E. W. Sachs, *Numerical solution of parabolic state constrained control problems using SQP– and interior–point–methods*, in Large Scale Optimization: State of the Art, W. W. Hager, D. Hearn, and P. Pardalos, eds., Kluwer, 1994, pp. 251–264.

[93] K. Levenberg, *A method for the solution of certain nonlinear problems in least squares*, Quart. Appl. Math., 2 (1944), pp. 164–168.

[94] Y. Li, *On global convergence of a trust region and affine scaling method for nonlinearly constrained minimization*, Tech. Rep. CTC94TR197, Advanced Computing Research Institute, Cornell University, 1994.

[95] ——, *A trust region and affine scaling method for nonlinearly constrained minimization*, Tech. Rep. CTC94TR198, Advanced Computing Research Institute, Cornell University, 1994.

[96] D. G. Luenberger, *Linear and Nonlinear Programming*, Addison-Wesley Publishing Company, Massachusetts, 1989.

[97] D. W. Marquardt, *An algorithm for least squares estimation of nonlinear parameters*, SIAM J. Math. Anal., 11 (1963), pp. 431–441.

[98] H. J. Martinez, Z. Parada, and R. A. Tapia, *On the characterization of q–superlinear convergence of quasi–Newton interior–point methods for nonlinear programming*, Boletin de la Sociedad Matematica Mexicana, 1 (1995), pp. 1–12.

[99] J. M. Martinez, *An algorithm for solving sparse nonlinear least squares problems*, Computing, 39 (1987), pp. 307–325.

[100] J. M. Martinez and S. A. Santos, *A trust–region strategy for minimization on arbitrary domains*, Math. Programming, 68 (1995), pp. 267–301.

[101] J. C. Meza and W. W. Symes, *Conjugate residual methods for almost symmetric linear systems*, J. Optim. Theory Appl., 72 (1992), pp. 415–440.

[102] J. J. Moré, *The Levenberg–Marquardt algorithm: implementation and theory*, in Proceedings of the Dundee Conference on Numerical Analysis, G. A. Watson, ed., Springer Verlag, New York, 1978.

[103] ——, *Recent developments in algorithms and software for trust regions methods*, in Mathematical programming. The state of art, A. Bachem, M. Grotschel, and B. Korte, eds., Springer Verlag, New York, 1983, pp. 258–287.

[104] ——, *A collection of nonlinear model problems*, in Computational Solution of Nonlinear Systems of Equations, E. L. Allgower and K. Georg, eds., Americal Mathematical Society, Providence , Rhode Island, 1990, pp. 723–762. Lectures in Applied Mathematics Vol. 26.

[105] ——, *Generalizations of the trust region problem*, Optimization Methods and Software, 2 (1993), pp. 189–209.

[106] J. J. Moré and D. C. Sorensen, *Computing a trust region step*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 553–572.

[107] J. J. Moré and D. Thuente, *Line search algorithms with guaranteed sufficient decrease*, ACM Trans. Math. Software, 20 (1994), pp. 286–307.

[108] W. Murray and F. J. Prieto, *A sequential quadratic programming algorithm using an incomplete solution of the subproblem*, SIAM J. Optim., 5 (1995), pp. 590–640.

[109] S. G. Nash, *Newton–like minimization via the Lanczos method*, SIAM J. Numer. Anal., 21 (1984), pp. 770–788.

[110] ——, *Solving nonlinear programming problems using truncated–Newton techniques*, in Numerical Optimization 1984, P. T. Boggs, R. H. Byrd, and R. B. Schnabel, eds., SIAM, Philadelphia, 1985, pp. 119–136.

[111] S. G. Nash and J. Nocedal, *A numerical study of the limited memory BFGS method and the truncated–Newton method for large scale optimization*, SIAM J. Optim., 1 (1991), pp. 358–372.

[112] S. G. Nash and A. Sofer, *Linear and Nonlinear Programming*, McGraw–Hill, New York, 1996.

[113] J. Nocedal, *Theory of algorithms for unconstrained optimization*, Acta Numerica, (1992), pp. 199–242.

[114] J. Nocedal and M. L. Overton, *Projected Hessian updating algorithms for nonlinearly constrained optimization*, SIAM J. Numer. Anal., 22 (1985), pp. 821–850.

[115] E. O. Omojokon, *Trust Region Algorithms for Optimization with Nonlinear Equality and Inequality Constraints*, PhD thesis, University of Colorado, 1989.

[116] J. M. Ortega and W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, New York, 1970.

[117] J. S. PANG, *Inexact Newton methods for the nonlinear complementarity problem*, Math. Programming, 36 (1986), pp. 54–71.

[118] T. PLANTENGA, *Large–Scale Nonlinear Constrained Optimization using Trust Regions*, PhD thesis, Northwestern University, Evanston, Illinois, 1994.

[119] E. POLAK, *Computational Methods in Optimization. A Unified Approach*, Academic Press, New York, London, Paris, San Diego, San Francisco, 1971.

[120] M. J. D. POWELL, *A new algorithm for unconstrained optimization*, in Nonlinear Programming, J. B. Rosen, O. L. Mangasarian, and K. Ritter, eds., Academic Press, New York, 1970.

[121] ——, *Convergence properties of a class of minimization algorithms*, in Nonlinear Programming 2, O. L. Mangasarian, R. R. Meyer, and S. M. Robinson, eds., Academic Press, New York, 1975, pp. 1–27.

[122] ——, *On the global convergence of trust region algorithms for unconstrained minimization*, Math. Programming, 29 (1984), pp. 297–303.

[123] M. J. D. POWELL AND Y. YUAN, *A trust region algorithm for equality constrained optimization*, Math. Programming, 49 (1991), pp. 189–211.

[124] J. RAPHSON, *Analysis Aequationum Universalis Seu Ad Aequationes Algebraicas Resolvendas Methodus Generalis, et Expedita, Ex nova Infinitarum Serierum Doctrina, Deducta Ac Demonstrata*, London, 1690. Original in British Library, London.

[125] C. H. REINSCH, *Smoothing by spline functions II*, Numer. Math., 16 (1971), pp. 451–454.

[126] F. RENDL AND H. WOLKOWICZ, *A semidefinite framework for trust region subproblems with applications to large scale minimization*, Tech. Rep. 94–32, CORR, 1994.

[127] Y. SAAD AND M. H. SCHULTZ, *GMRES a generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.

[128] C. M. Samuelson, *The Dikin–Karmarkar Principle for Steepest Descent*, PhD thesis, Department of Computational and Applied Mathematics, Rice University, Houston, Texas 77251, USA, 1992. Tech. Rep. TR92–29.

[129] S. A. Santos and D. C. Sorensen, *A new matrix–free algorithm for the large–scale trust–region subproblem*, Tech. Rep. TR95–20, Department of Computational and Applied Mathematics, Rice University, 1994.

[130] G. A. Shultz, R. B. Schnabel, and R. H. Byrd, *A family of trust–region–based algorithms for unconstrained minimization with strong global convergence properties*, SIAM J. Numer. Anal., 22 (1985), pp. 47–67.

[131] T. Simpson, *Essays on several Curious and Useful Subjects, In Speculative and Mix'd Mathematicks, Ilustrated by a Variety of Examples*, London, 1740.

[132] D. C. Sorensen, *Newton's method with a model trust region modification*, SIAM J. Numer. Anal., 19 (1982), pp. 409–426.

[133] ——, *Minimization of a large scale quadratic function subject to an spherical constraint*, SIAM J. Optim., (To appear).

[134] T. Steihaug, *The conjugate gradient method and trust regions in large scale optimization*, SIAM J. Numer. Anal., 20 (1983), pp. 626–637.

[135] ——, *Local and superlinear convergence for truncated iterated projections methods*, Math. Programming, 27 (1983), pp. 176–190.

[136] R. Stern and H. Wolkowitz, *Indefinite trust region subproblems and nonsymmetric eigenvalue perturbations.* To appear in SIAM J. Optim., 1995.

[137] S. W. Thomas, *Sequential Estimation Techniques for Quasi-Newton Algorithms*, PhD thesis, Cornell University, Ithaca, New York, 1975.

[138] A. N. Tichonoff, *Methods for the regularization of optimal control problems*, Dokl. Akad. Nauk., Soviet Maths., 162 (1965), pp. 761–763.

[139] P. L. Toint, *Towards an efficient sparsity exploiting Newton method for minimization*, in Sparse Matrices and Their Uses, I. S. Duff, ed., Academic Press, New York, 1981, pp. 57–87.

[140] ——, *Global convergence of a class of trust–region methods for nonconvex minimization in Hilbert space*, IMA J. Numer. Anal., 8 (1988), pp. 231–252.

[141] A. Vardi, *A trust region algorithm for equality constrained minimization: convergence properties and implementation*, SIAM J. Numer. Anal., 22 (1985), pp. 575–591.

[142] L. N. Vicente, *What happens when we trust a region that is a line*, Tech. Rep. TR95–10, Department of Computational and Applied Mathematics, Rice University, 1995.

[143] D. T. Whiteside, ed., *The Mathematical Papers of Issac Newton (Volumes I–VII)*, Cambridge University Press, Cambridge, 1967–1976.

[144] P. Wolfe, *Convergent conditions for ascent methods*, SIAM Rev., 11 (1969), pp. 226–235.

[145] ——, *Convergent conditions for ascent methods. II: Some corrections*, SIAM Rev., 13 (1971), pp. 185–188.

[146] S. J. Wright, *Interior point methods for optimal control of discrete–time systems*, J. Optim. Theory Appl., 77 (1993), pp. 161–187.

[147] Y. Xie, *Reduced Hessian Algorithms for Solving Large–Scale Equality Constrained Optimization Problems*, PhD thesis, Dept. of Computer Science, University of Colorado, 1991.

[148] H. Yamashita, *A globally convergent primal–dual interior–point method for constrained optimization*, tech. rep., Mathematical Systems Institute, Japan, 1992.

[149] D. P. Young, W. P. Huffman, R. G. Melvin, M. B. Bieterman, C. L. Hilmes, and F. T. Johnson, *Inexactness and global convergence in design optimization*, in 5th AIAA/NASA/USAF/ISSMO Symposium on Multidisciplinary Analysis and Optimization, September 1994.

[150] T. Ypma, *Historical development of the Newton–Raphson method*, SIAM Rev., 37 (1995), pp. 531–551.

[151] H. Yserentant, *On the multi–level splitting of finite element spaces*, Numer. Math., 49 (1986), pp. 379–412.

[152] Y. Yuan, *On a subproblem of trust region algorithms for constrained optimization*, Math. Programming, 47 (1990), pp. 53–63.

[153] ——, *A dual algorithm for minimizing a quadratic function with two quadratic constraints*, J. Comput. Math., 9 (1991), pp. 348–359.

[154] ——, *On the convergence of a new trust region algorithm*, Numer. Math., 70 (1995), pp. 515–539.

[155] J. Zhang, N. Kim, and L. Lasdon, *An improved successive linear programming algorithm*, Management Sci., 31 (1985), pp. 1312–1331.

[156] J. Z. Zhang and D. T. Zhu, *Projected quasi–Newton algorithm with trust region for constrained optimization*, J. Optim. Theory Appl., 67 (1990), pp. 369–393.

[157] Y. Zhang, *Computing a Celis–Dennis–Tapia trust–region step for equality constrained optimization*, Math. Programming, 55 (1992), pp. 109–124.

[158] G. Zoutendijk, *Nonlinear Programming, Computational Methods*, in Integer and Nonlinear Programming, J. Abadie, ed., North–Holland, Amsterdam, 1970, pp. 37–86.