UNIVERSITY OF CALIFORNIA SAN DIEGO

Second-Derivative SQP Methods for Large-Scale Nonconvex Optimization

A dissertation submitted in partial satisfaction of the
requirements for the degree
Doctor of Philosophy

in

Mathematics

by

Jeb H. Runnoe

Committee in charge:

Professor Philip E. Gill, Chair
Professor Robert R. Bitmead
Professor Michael J. Holst
Professor Melvin Leok
Professor Wenxin Zhou

2024

The Dissertation of Jeb H. Runnoe is approved, and it is acceptable in quality and form for publication on microfilm and electronically.

University of California San Diego

2024

DEDICATION

To my family.

TABLE OF CONTENTS

Dissertation Approval Page . . . iii

Dedication . . . iv

Table of Contents . . . v

List of Figures . . . vii

List of Tables . . . ix

Acknowledgements . . . x

Vita . . . xi

Abstract of the Dissertation . . . xii

Chapter 1 Introduction . . . 1
1.1 Notation . . . 2
1.2 Background . . . 4
1.3 Overview . . . 7
1.4 Contributions of This Dissertation . . . 13

Chapter 2 Sequential Quadratic Programming . . . 16
2.1 Local Properties of SQP Methods . . . 16
2.1.1 Equality constraints . . . 19
2.1.2 Inequality constraints . . . 34
2.2 Methods for Quadratic Programming . . . 41
2.2.1 Primal active-set methods . . . 43

Chapter 3 Stabilized and Primal-Dual SQP Methods . . . 55
3.1 A Regularized Primal-Dual Line-Search SQP Algorithm . . . 56
3.2 Definition of the Primal-Dual Search Direction . . . 57
3.2.1 Definition of the new iterate . . . 63
3.2.2 Updating the multiplier estimate . . . 65
3.2.3 Updating the penalty parameters . . . 66
3.3 Solution of the Bound-Constrained Subproblem . . . 67
3.3.1 Convexification of the bound-constrained subproblem . . . 69

Chapter 4 Modifying Matrix Factorizations . . . 72
4.1 Tiling . . . 73
4.1.1 Two-stage factorization . . . 77
4.2 First-Stage Strategy . . . 85
4.2.1 Submatrix search . . . 86
4.3 Two-Stage Symmetric Indefinite factorization with Partial Cholesky Decomposition . . . 91

4.3.1 Utilizing the partial Cholesky factors . . . 94
4.4 Full Diagonal Modification of K . . . 97

Chapter 5 Dynamic Convexification . . . 101
5.1 Dynamic Convexification of a QP in Standard Form . . . 101
5.1.1 Non-binding active-set methods in standard form . . . 102
5.1.2 Pre-convexification . . . 106
5.1.3 Concurrent convexification . . . 112
5.1.4 Post-convexification for constraints in standard form . . . 115
5.2 Dynamic Convexification of Stabilized SQP Methods . . . 121
5.2.1 The stabilized subproblem – standard form . . . 122
5.2.2 Pre-convexification and regularization . . . 123
5.2.3 Concurrent convexification of a stabilized QP subproblem . . . 126
5.2.4 Stabilized post-convexification . . . 129
5.3 Primal-Dual SQP methods with Dynamic Convexification . . . 137
5.3.1 Pre-convexification of the bound-constrained subproblem . . . 138
5.3.2 Concurrent convexification of the bound-constrained QP . . . 138
5.3.3 Post-convexification of the bound-constrained QP . . . 143

Chapter 6 A Dynamically-Convexified Primal-Dual SQP Algorithm . . . 153
6.1 Formal Algorithm Statement . . . 153
6.2 Convergence . . . 156
6.3 Numerical Results . . . 162
6.3.1 Implementation . . . 162
6.3.2 Performance profiles . . . 164

Bibliography . . . 176

LIST OF FIGURES

Figure 6.1: Performance profiles comparing function evaluations, iterations, and factorizations used by the algorithms `dcpdSQP` and `pdSQP` when applied to 999 problems from the combined (ALL) CUTEst test set. The (ALL) set is the union of the (BC), (FP), (HS), (LC), and (NC) test sets. 166

Figure 6.2: Performance profiles comparing function evaluations, iterations, and factorizations used by the algorithms `dcpdSQP` and `pdSQP` when applied to 139 bound constrained (BC) problems from the CUTEst test set. 167

Figure 6.3: Performance profiles comparing function evaluations, iterations, and factorizations used by the algorithms `dcpdSQP` and `pdSQP` when applied to 262 feasible-point (FP) problems *with an artificial objective function* from the CUTEst test set. 168

Figure 6.4: Performance profiles comparing function evaluations, iterations, and factorizations used by the algorithms `dcpdSQP` and `pdSQP` when applied to 262 feasible-point (FP) problems *with no objective function* from the CUTEst test set. 169

Figure 6.5: Performance profiles comparing function evaluations, iterations, and factorizations used by the algorithms `dcpdSQP` and `pdSQP` when applied to 126 Hock-Shittkowski (HS) problems from the CUTEst test set. 169

Figure 6.6: Performance profiles comparing function evaluations, iterations, and factorizations used by the algorithms `dcpdSQP` and `pdSQP` when applied to 212 linearly constrained (LC) problems from the CUTEst test set. 170

Figure 6.7: Performance profiles comparing function evaluations, iterations, and factorizations used by the algorithms `dcpdSQP` and `pdSQP` when applied to 386 nonlinearly constrained (NC) problems from the CUTEst test set. 170

Figure 6.8: Performance profiles comparing pre-convexification methods in `dcpdSQP` with `pdSQP` when applied to 999 problems from the combined (ALL) CUTEst test set. 171

Figure 6.9: Performance profiles comparing pre-convexification methods in `dcpdSQP` with `pdSQP` when applied to 139 bound constrained (BC) problems from the CUTEst test set. 171

Figure 6.10: Performance profiles comparing pre-convexification methods in `dcpdSQP` with `pdSQP` when applied to 262 feasible-point (FP) problems from the CUTEst test set. 172

Figure 6.11: Performance profiles comparing pre-convexification methods in `dcpdSQP` with `pdSQP` when applied to 126 Hock-Shittkowski (HS) problems from the CUTEst test set. 172

Figure 6.12: Performance profiles comparing pre-convexification methods in `dcpdSQP` with `pdSQP` when applied to 212 linearly constrained (LC) problems from the CUTEst test set. 173

Figure 6.13: Performance profiles comparing pre-convexification methods in `dcpdSQP` with `pdSQP` when applied to 386 nonlinearly constrained (NC) problems from the CUTEst test set. 173

Figure 6.14: Performance profiles comparing pre-convexification methods in `dcpdSQP` with `pdSQP` when applied to 173 unconstrained (UC) problems from the CUTEst test set. 174

LIST OF TABLES

Table 1.1: Common notation. 3

Table 6.1: Control parameters for Algorithms `pdSQP` and `dcpdSQP`. 164
Table 6.2: Problem set (ALL) outcome counts. 175
Table 6.3: Problem set (BC) outcome counts. 175
Table 6.4: Problem set (FP) outcome counts. 175
Table 6.5: Problem set (HS) outcome counts. 175
Table 6.6: Problem set (LC) outcome counts. 175
Table 6.7: Problem set (NC) outcome counts. 175

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to my advisor and mentor, Professor Philip Gill. The dedication and patience with which he has guided and supported me throughout my development cannot be overstated. He has truly invested himself in helping me to succeed, and he has played an integral role in making it possible for me to elevate my life through my education. I feel I am very fortunate to have had the opportunity to work with him, and I will always be grateful for all he has done for me.

I would also like to thank my doctoral committee members Professor Melvin Leok, Professor Michael Holst, Professor Wenxin Zhou, and Professor Robert Bitmead, for their support, feedback, and inspiration. I appreciate the time and effort spent in sharing their expertise and guidance to help me complete this important part of the process.

I also want to acknowledge the colleagues and instructors I got to work with and learn from along the way. Going through the challenging coursework with my classmates not only prepared me with the knowledge needed to do this research, but also taught me how to manage my time, how to respond to frustration, and how deal with uncertainty.

Lastly, I want to thank my family. I am grateful to my parents for their unwavering love and support, and to my lovely wife, for being my tireless advocate and believing in me every step of the way. I especially want to thank my sister, both for her encouragement and for the example she set, which helped me to see that I could succeed if I truly applied myself.

VITA

2020	B. S. in Applied Mathematics, University of California San Diego
2022	M. S. in Applied Mathematics, University of California San Diego
2020-2024	Graduate Research and Teaching Assistant, University of California San Diego
2024	Ph. D. in Mathematics, University of California San Diego

ABSTRACT OF THE DISSERTATION

Second-Derivative SQP Methods for

Large-Scale Nonconvex Optimization

by

Jeb H. Runnoe

Doctor of Philosophy in Mathematics

University of California San Diego, 2024

Professor Philip E. Gill, Chair

The class of stabilized sequential quadratic programming (SQP) methods for nonlinearly constrained optimization solve a sequence of related quadratic programming (QP) subproblems formed from a two-norm penalized quadratic model of the Lagrangian function subject to shifted, linearized constraints. While these methods have been shown to exhibit superlinear local convergence even when the constraint Jacobian is rank deficient at the solution, they generally have no global convergence theory. To address this issue, primal-dual SQP methods (`pdSQP`) employ a certain primal-dual augmented Lagrangian merit function and solve a subproblem that involves the minimization of a quadratic model of the merit function subject to simple bound constraints.

The model of the merit function is constructed so that the resulting primal-dual subproblem is equivalent to the stabilized SQP subproblem. When used in conjunction with a flexible line-search, the merit function guarantees convergence from any starting point, while the connection with the stabilized subproblem allows `pdSQP` to retain the superlinear local convergence that is characteristic of stabilized SQP methods.

A new dynamic convexification framework is developed that is applicable for *nonconvex* general standard form, stabilized, and primal-dual bound-constrained QP subproblems. Dynamic convexification involves three distinct stages: pre-convexification, concurrent convexification and post-convexification. New techniques are derived and analyzed for the implicit modification of symmetric indefinite factorizations and for the imposition of temporary artificial constraints, both of which are suitable for pre-convexification. Concurrent convexification works synchronously with the active-set method used to solve the subproblem, and computes minimal modifications needed to ensure that the QP iterates are uniformly bounded. Finally, post-convexification defines an *implicit* modification that ensures the solution of the subproblem yields a descent direction for the merit function.

A new exact second-derivative primal-dual SQP method (`dcpdSQP`) is formulated for large-scale nonconvex optimization. Convergence analysis is presented that demonstrates guaranteed global convergence. Extensive numerical testing indicates that the performance of the proposed method is comparable or better than conventional full convexification while significantly reducing the number of factorizations required.

Chapter 1

Introduction

The spirit of optimization is something everyone is familiar with in one way or another. It is a fundamentally human endeavor in that most people are constantly searching for and taking actions to increase their health, income, and happiness, while simultaneously reducing time and energy spent. When purchasing a house, for example, one attempts to maximize some, often vaguely-defined, measure of utility that depends on variables such as cost, location, commute, and square footage. Some requirements may be flexible while others are not. Whether or not the individual realizes it, many of the fundamental components that constitute an optimization problem are present in such a task. Moreover, there are numerous problems throughout science, engineering, finance, economics, and medicine, that can be posed as some form of optimization problem. Many areas of optimization research originated from the need to solve problems that arise naturally in disciplines such as these. A prime example is the soldier diet planning problem that, along with other military applications, spurred the development of the simplex method and the study of linear programming. In order to solve such problems, a mathematical model is constructed that seeks to capture the

problem's defining characteristics while removing needless complexity. When the problem-specific details are abstracted away, what's left is a general form optimization problem. Because many important and interesting problems share characteristics such as smoothness, linearity, and problem function definitions, the solution of the abstract optimization problem is widely applicable.

Mathematical optimization is concerned with the formulation and analysis of methods for solving abstract optimization problems. This amounts to selecting an element x^* from a set $\mathcal{X}$ of possible alternatives such that the value taken by a function of interest $f(x^*)$ is extremal over $\mathcal{X}$. As the maximization of $f(x)$ is mathematically equivalent to the minimization of $-f(x)$, we will focus our attention on minimization. With this in mind, an abstract optimization problem can be written

$$\underset{x \in \mathcal{X}}{\text{minimize}} \;\; f(x). \tag{1.1}$$

The power of mathematical optimization is that the methods and solutions derived by studying (1.1) apply to a diverse set of applications, including the purchase of a house or the formulation of diet plans for the military.

1.1 Notation

The process of modifying a matrix so that the result is positive definite will be referred to as *convexification* or *convexifying* the matrix. Similarly, a function is convexified by modifying its Hessian matrix to be positive definite, and convexification of an optimization problem refers to modification of the Hessian of the objective function.

In general, the notation will be defined as it is introduced. That said, the main notation is summarized here for reference.

Table 1.1: Common notation.

Notation	**Meaning**
x	The n-vector of primal variables.
$f(x)$	The scalar-valued objective function.
$g(x)$	The gradient $\nabla f(x)$ of the objective function.
$c(x)$	The m-vector of general constraint functions $c_i(x)$.
$J(x)$	The $m \times n$ Jacobian of $c(x)$.
y	The m-vector of Lagrange multipliers for the general constraints $c(x) = 0$.
z	The n-vector of Lagrange multipliers for the bound constraints $x \geq 0$.
$L(x, y, z)$	Lagrangian function $L(x, y, z) = f(x) - y^{\mathrm{T}} c(x) - z^{\mathrm{T}} x$.
$H(x, y)$	Hessian of the Lagrangian $\nabla^2 L(x, y, z)$ taken with respect to x.
$[\, u \,]_i$	The i-th component of a vector u.
$[\, u \,]_{\mathcal{S}}$	The components of u with indices in $\mathcal{S}$.
$u \cdot v$	Element-wise vector product $[\, u \cdot v \,]_i = u_i v_i$.
α_k	Step length.
$\| \cdot \|$	Euclidean vector or induced matrix norm.
$i_+(A)$	The number of positive eigenvalues of a symmetric matrix A.
$i_-(A)$	The number of negative eigenvalues of a symmetric matrix A.
$i_0(A)$	The number of zero eigenvalues of a symmetric matrix A.
$\mathrm{In}(A)$	The inertia of symmetric A: $\big(i_+(A),\, i_-(A),\, i_0(A)\big)$.
e	A column vector of ones.
e_i	The i-th column of the identity matrix.
u^+	The positive part of u: $u^+ = \max(u,\, 0) \geq 0$.
u^-	The negative part of u: $u^- = -\min(u,\, 0) \geq 0$.

1.2 Background

In practice, the set of allowed values is defined in terms of constraint functions rather than an abstract set, yielding the form

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \;\; f(x) \quad \text{subject to} \quad \begin{pmatrix} \ell^X \\ \ell^S \end{pmatrix} \leq \begin{pmatrix} x \\ c(x) \end{pmatrix} \leq \begin{pmatrix} u^X \\ u^S \end{pmatrix}, \tag{1.2}$$

where $c : \mathbb{R}^n \mapsto \mathbb{R}^m$, $f : \mathbb{R}^n \mapsto \mathbb{R}$, and (ℓ^X, ℓ^S) and (u^X, u^S) are constant vectors of lower and upper bounds. Throughout, we assume that the number of variables is large, and that the derivatives of f and c are sparse. The constraints involving the functions $c_i(x)$ will be called the *general* constraints; the remaining constraints will be called *bounds*. We assume that the functions f and c are smooth and that their first and second derivatives are available. An *equality* constraint corresponds to the values $\ell_i = u_i$. Similarly, a special "infinite" value for ℓ_i or u_i is used to indicate the absence of one of the bounds. To relate (1.1) with (1.2), note that $\mathcal{X}$ is simply

$$\mathcal{X} = \{x \in \mathbb{R}^n \;:\; \ell^X \leq x \leq u^X \;\text{ and }\; \ell^S \leq c(x) \leq u^S\}.$$

The problem format of (1.2) may be simplified by introducing slack variables and replacing each general constraint of the form $\ell_i \leq \varphi_i(x) \leq u_i$ by the equality constraint $\varphi_i(x) - s_i = 0$ and range constraint $\ell_i \leq s_i \leq u_i$. This gives

$$\underset{x\in\mathbb{R}^n,\, s\in\mathbb{R}^m}{\text{minimize}} \;\; f(x) \quad \text{subject to} \;\; c(x) - s = 0, \;\; \begin{pmatrix} \ell^X \\ \ell^S \end{pmatrix} \leq \begin{pmatrix} x \\ s \end{pmatrix} \leq \begin{pmatrix} u^X \\ u^S \end{pmatrix}, \tag{1.3}$$

where x and the "slack variables" s are treated as independent variables. Without loss of generality, we assume only nonnegativity constraints and include any slack variables in the definition of x and

c. The problem to be solved is then

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \;\; f(x) \quad \text{subject to} \;\; c(x) = 0, \quad x \geq 0, \tag{NP}$$

where f and the m components of the constraint vector c are assumed to be twice continuously differentiable for all $x \in \mathbb{R}^n$. The methods designed to solve (NP) are easily applied to solve the more general problem (1.3). Let $g(x)$ denote $\nabla f(x)$, the gradient of f evaluated at x. Similarly, let $J(x)$ denote the $m \times n$ constraint Jacobian with rows formed from the constraint gradients $\nabla c_i(x)$. Throughout the discussion, the component y_i of the m-vector y will denote the dual variable associated with the constraint $c_i(x) = 0$ or its linearization. Similarly, z_j denotes the dual variable associated with the bound $x_j \geq 0$. The dual variables y and z are also referred to as *Lagrange multipliers*.

A constraint is *active* at x if it is satisfied with equality. For any *feasible* x, i.e., for any x such that $c(x) = 0$ and $x \geq 0$, all m equality constraints $c_i(x) = 0$ are necessarily active. The indices associated with the active nonnegativity constraints comprise the *active set*, denoted by $\mathcal{A}(x)$, i.e., $\mathcal{A}(x) = \big\{ i : x_i = 0 \big\}$. A nonnegativity constraint that is not in the active set is said to be *inactive*. The *inactive set* contains the indices of the inactive constraints, i.e., the so-called "free" variables $\mathcal{F}(x) = \big\{ i : x_i > 0 \big\}$.

Under certain constraint regularity assumptions, an optimal solution of (NP) must satisfy conditions that may be written in terms of the derivatives of the Lagrangian function $L(x, y, z) = f(x) - y^{\mathrm{T}} c(x) - z^{\mathrm{T}} x$. The triple (x^*, y^*, z^*) is said to be a first-order KKT point for problem (NP)

if it satisfies the KKT conditions

$$\begin{aligned} c(x^*) &= 0, \qquad x^* \geq 0, \\ g(x^*) - J(x^*)^{\mathrm{T}} y^* - z^* &= 0, \\ x^* \cdot z^* &= 0, \qquad z^* \geq 0. \end{aligned} \tag{1.4}$$

The property of strict complementarity holds if the vectors x^* and z^* satisfy $x^* \cdot z^* = 0$ with $x^* + z^* > 0$. The vector-triple (x, y, z) is said to constitute a *primal-dual estimate* of the quantities (x^*, y^*, z^*) satisfying (1.4).

The purpose of the constraint regularity assumption is to guarantee that a linearization of the constraints describes the nonlinear constraints with sufficient accuracy that the KKT conditions of (1.4) are necessary for local optimality. One such regularity assumption is the *Mangasarian-Fromovitz constraint qualification* [54, 57], which requires that $J(x^*)$ has rank m, and that there exists a vector p such that $J(x^*)p = 0$ with $p_i > 0$ for all $i \in \mathcal{A}(x^*)$. Another common, but slightly more restrictive, assumption is the *linear independence constraint qualification*, which requires that the matrix of free columns of $J(x^*)$ has full row rank.

Let $H(x, y)$ denote the Hessian of $L(x, y, z)$ with respect to x, i.e.,

$$H(x, y) = \nabla^2_{xx} L(x, y, z) = \nabla^2 f(x) - \sum_{i=1}^{m} y_i \nabla^2 c_i(x).$$

Under the linear independence constraint qualification, the second-order necessary optimality conditions require that the first-order conditions (1.4) hold with the additional condition that

$$p^{\mathrm{T}} H(x^*, y^*) p \geq 0 \text{ for all } p \text{ such that } J(x^*)p = 0, \text{ and } p_i = 0 \text{ for every } i \in \mathcal{A}(x^*).$$

See, e.g., Nocedal and Wright [57, Chapter 12] for more discussion of constraint assumptions and optimality conditions.

For a feasible point x, we will denote by $J_{\mathcal{F}}(x)$ the matrix comprising columns of $J(x)$ corresponding to indices in $\mathcal{F}(x)$. A point x at which $[\,g(x)\,]_{\mathcal{F}} \in \text{range}(J_{\mathcal{F}}(x)^{\mathrm{T}})$ and the linear independence constraint qualification does not hold is said to be *degenerate*. For example, if x is a degenerate vertex, then more than $n - m$ bounds must be active and $J_{\mathcal{F}}(x)$ has more rows than columns. The Mangasarian-Fromovitz constraint qualification may or may not hold at a degenerate point. Practical NLP problems with degenerate points are very common and it is crucial that an algorithm is able to handle $J_{\mathcal{F}}(x)$ with dependent rows. Throughout our discussion of the effects of degeneracy in SQP methods, it will be assumed that the Mangasarian-Fromovitz regularity assumption holds.

1.3 Overview

There are two primary classes of methods available for solving (NP): *interior-point* methods and *sequential quadratic programming* (SQP) methods, defined by two alternative approaches to handling the inequality constraints in (NP). SQP methods find an approximate solution of a sequence of *quadratic programming* (QP) subproblems in which a quadratic model of the objective function is minimized subject to the linearized constraints. Interior methods approximate a continuous path that passes through a solution of (NP). In the simplest case, the path is parameterized by a positive scalar parameter μ that may be interpreted as a perturbation for the optimality conditions (1.4). Both interior methods and SQP methods have an inner/outer iteration structure, with the work for an inner iteration being dominated by the cost of solving a large sparse system of symmetric indefinite linear equations. In the case of SQP methods, these equations involve a subset

of the variables and constraints; for interior methods, the equations involve all the constraints and variables.

SQP methods provide a relatively reliable "certificate of infeasibility" and they have the potential of being able to capitalize on a good initial starting point. Sophisticated matrix factorization updating techniques are used to exploit the fact that the linear equations change by only a single row and column at each inner iteration. These updating techniques are often customized for the particular QP method being used and have the benefit of providing a uniform treatment of ill-conditioning and singularity.

On the negative side, it is difficult to implement SQP methods so that exact second derivatives can be used efficiently and reliably. Some of these difficulties stem from the theoretical properties of the quadratic programming subproblem, which can be nonconvex when second derivatives are used. Nonconvex quadratic programming is NP-hard—even for the calculation of a local minimizer (see Contesse [14] and Forsgren, Gill and Murray [29]. The complexity of the QP subproblem has been a major impediment to the formulation of second-derivative SQP methods (although methods based on indefinite QP have been proposed by Fletcher [23, 24]). Over the years, algorithm developers have avoided this difficulty by eschewing second derivatives and by solving a convex QP subproblem defined with a positive semidefinite quasi-Newton approximate Hessian (see, e.g., Gill, Murray and Saunders [35]). There are other difficulties associated with conventional SQP methods that are not specifically related to the use of second derivatives. An SQP algorithm is often tailored to a particular updating technique, e.g., the matrix factors of the Jacobian in the outer iteration can be chosen to match those of the method for the QP subproblem. Any reliance on customized linear algebra software makes it hard to "modernize" a method to reflect new developments in software technology (e.g., in languages that exploit new advances in computer hardware such as multicore

processors or GPU-based architectures). Another difficulty is that active-set methods may require a substantial number of QP iterations when the outer iterates are far from the solution. The use of a QP subproblem is motivated by the assumption that the QP objective and constraints provide good "models" of the objective and constraints of the NLP (see Section 2.1). This should make it unnecessary (and inefficient) to solve the QP to high accuracy during the preliminary iterations. Unfortunately, the simple expedient of limiting the number of inner iterations may have a detrimental effect upon reliability. Moreover, some of the QP multipliers will have the wrong sign if an active-set method is terminated before a solution is found. This may cause difficulties if the QP multipliers are used to estimate the multipliers for the nonlinear problem. These issues would largely disappear if a primal-dual *interior* method were to be used to solve the QP subproblem. These methods have the benefit of providing a sequence of feasible (i.e., correctly signed) dual iterates. Nevertheless, QP solvers based on conventional interior methods have had limited success within SQP methods because they are difficult to "warm start" from a near-optimal point (see the discussion below). This makes it difficult to capitalize on the property that, as the outer iterates converge, the solution of one QP subproblem is a very good estimate of the solution of the next.

Broadly speaking, the advantages and disadvantages of SQP methods and interior methods complement each other. Interior methods are most efficient when implemented with exact second derivatives. Moreover, they can converge in few inner iterations—even for very large problems. The inner iterates are the iterates of Newton's method for finding an approximate solution of the perturbed optimality conditions for a given μ. As the dimension and zero/nonzero structure of the Newton equations remains *fixed*, these Newton equations may be solved efficiently using either iterative or direct methods available in the form of advanced "off-the-shelf" linear algebra software. In particular, any new software for multicore and parallel architectures is immediately applicable.

Moreover, the perturbation parameter μ plays an auxiliary role as an implicit regularization parameter of the linear equations. This implicit regularization plays a crucial role in the robustness of interior methods on ill-conditioned and ill-posed problems.

On the negative side, although interior methods are very effective for solving "one-off" problems, they are difficult to adapt to solving a sequence of related nonlinear programming problems. This difficulty may be explained in terms of the "path-following" interpretation of interior methods. In the neighborhood of an optimal solution, a step *along* the path $x(\mu)$ of perturbed solutions is well-defined, whereas a step *onto* the path from a neighboring point will be extremely sensitive to perturbations in the problem functions (and hence difficult to compute). Another difficulty with conventional interior methods is that a substantial number of iterations may be needed when the constraints are infeasible.

Interior and SQP methods also have several similarities. One of the approaches used by SQP methods for handling degeneracy is constraint regularization, which is the foundation of *stabilized* SQP methods, discussed in detail in Chapters 2 and 3. Similar to interior methods, stabilization can also be viewed as a perturbation of the problem, parameterized by a positive scalar parameter μ. As a result, many the linear equations that arise in stabilized SQP and interior methods have a common structure. Modern stabilized SQP and interior methods use a *merit function* of some kind to ensure convergence from any starting point. A merit function measures the quality of an iterate in terms of feasibility and optimality, and allows for precise quantification of the "improvement" made from one iteration to the next. By using a line search or trust region strategy, these methods are able to ensure a reduction in the merit function that is sufficient to guarantee global convergence is achieved at each step. The direction toward the next iterate is computed from a system of linear equations involving an approximation to the Hessian of the merit

function, which can be expressed in the form

$$H^M = \begin{pmatrix} H + 2J^{\mathrm{T}}D^{-1}J & J^{\mathrm{T}} \\ J & D \end{pmatrix}, \tag{1.5}$$

where D is a positive-definite diagonal matrix that depends on the method being used. Unfortunately, even when H, J and D are sparse and well-conditioned, H^M may be dense and ill-conditioned, so it is not generally advisable to use H^M directly. Any system of equations involving H^M can be solved using the *symmetric KKT matrix*

$$K = \begin{pmatrix} H & J^{\mathrm{T}} \\ J & -D \end{pmatrix} \tag{1.6}$$

instead, because the two matrices are related by the nonsingular transformation

$$UH^MV = K, \qquad \text{where} \qquad U = \begin{pmatrix} I_n & -2J^{\mathrm{T}}D^{-1} \\ 0 & I_m \end{pmatrix} \text{ and } V = \begin{pmatrix} I_n & 0 \\ 0 & -I_m \end{pmatrix}.$$

Interestingly, matrices of the form (1.6) arise independently in both interior and stabilized SQP methods. In interior methods, K appears in the symmetrized Newton equations for computing a root of a vector-valued nonlinear function whose zero is the solution of the perturbation to the optimality conditions (1.4). In stabilized SQP, K appears in the linear equations for the optimality conditions of the stabilized QP subproblem. In both cases, it is advantageous to use K rather than H^M because it inherits the sparsity and conditioning of the original problem.

In order to ensure existence of a step that will yield a sufficient decrease in the merit function, it is required that H^M be positive definite. However, H^M may be indefinite at some iterates when exact second derivatives are used. For this reason, some form of modification to H^M

may be needed to ensure that the computed direction possesses the required descent properties. Of the techniques available for computing the needed modification, the method of Wächter and Biegler [68], is often the only suitable option for large sparse problems. The method exploits the fact that the inertias of H^M and K are related by the equation $\text{In}(H^M) = \text{In}(K) + (m,\ -m,\ 0)$, and proceeds to modify H^M *implicitly* by modifying K so that its inertia is $(n,\ m,\ 0)$, which implies that the resulting H^M is positive definite. A matrix K with inertia $(n,\ m,\ 0)$ is said to be *second-order consistent*. For an increasing sequence $\{\sigma_j\}$, a symmetric indefinite factorization of the matrix

$$K(\sigma_j) = \begin{pmatrix} H + \sigma_j I & J^{\mathrm{T}} \\ J & -D \end{pmatrix} \tag{1.7}$$

is computed, and the inertia of $K(\sigma_j)$ is checked by looking at the block diagonal B_j of the factorization $P_j^{\mathrm{T}} K(\sigma_j) P_j = L_j B_j L_j^{\mathrm{T}}$. If successful, the end result is a diagonal perturbation E and a sparse symmetric indefinite factorization $P^{\mathrm{T}}(K + E)P = LBL^{\mathrm{T}}$ such that $K + E$ is second-order consistent. The fact that the perturbation and the factors are all sparse is a major contributing factor to the efficacy of this method for large sparse problems. Limitations of this approach include that it is difficult to define the sequence $\{\sigma_j\}$ in a way that is favorable in general. The most obvious drawback, however, is the need to compute potentially many factorizations.

The preceding discussion illustrates that the *efficient* and *reliable* use of exact second derivatives is a principal difficulty in SQP methods. Keeping in mind that SQP methods constitute one of the primary classes of methods available for solving (NP), it is clear why there is considerable interest in developing techniques for effectively using second derivatives, particularly in the large-scale and nonconvex settings.

1.4 Contributions of This Dissertation

Both interior and SQP methods require the solution of large, sparse, symmetric indefinite systems of linear equations at each inner iteration. Moreover, dealing with nonconvexity demands modifications be made in order to guarantee well-defined equations and sufficient decrease in the outer iteration. In the case of interior methods, a factorization of an indefinite block 2×2 approximate penalty-barrier merit Hessian must be modified to carry out an approximate modified Newton method. In the case of SQP methods, a symmetric factorization of the block 2×2 indefinite QP Hessian must be modified to guarantee a well-defined, bounded, local solution of the subproblem can be found. In both cases, a factorization of a symmetric indefinite KKT system that inherits sparsity and conditioning of the original problem is used to compute the *implicit* needed modification to the generally dense and ill-conditioned approximate Hessian. In Chapter 4, new techniques are derived and analyzed for the implicit modification of symmetric factorizations of block 2×2 indefinite matrices. A two-stage split factorization is described that requires only two factorizations to guarantee the modified Hessian is positive definite. Both of the factorizations used can be computed using highly optimized "off-the-shelf" software, with no need to constrain the pivot order. The two-stage factorization yields a modification that affects only the first n rows and columns of the KKT system, which is critical to preserve the implicit relationship with the approximate Hessian. The first stage identifies and factors a positive semidefinite submatrix that is then paired, by symmetric permutation, with the $(2,\ 2)$-block of the original matrix to form a leading KKT submatrix. The Schur complement of this submatrix is formed and factored allowing for either a norm-optimal or a diagonal perturbation to be computed that achieves convexity implicitly. Several methods for the first stage are offered, including partial Cholesky factorization as well as novel submatrix search and approximate perturbation algorithms.

The primary focus will be on three variations of SQP methods: conventional, stabilized, and primal-dual. These variations are the result of different approaches to handling degeneracy, described in Section 1.2, and global convergence. Conventional SQP refers to methods that solve a standard quadratic subproblem with a rank-enforcing active-set algorithm, and use a merit function to guarantee convergence. Stabilized SQP avoids degeneracy by solving a perturbed subproblem that results from constraint regularization, but has no global convergence theory. Primal-dual SQP uses a primal-dual merit function to guarantee convergence and a specially formulated subproblem that is equivalent to stabilized SQP, thereby addressing degeneracy.

In Chapter 5, a new *dynamic convexification* technique is formulated for use in the solution of nonconvex conventional, stabilized, and primal-dual quadratic programming subproblems. Dynamic convexification involves three distinct stages: pre-convexification, concurrent convexification, and post-convexification. Among the possible approaches to pre-convexification are two novel techniques: (1) a method based on imposing temporary artificial bound constraints and recursive inertia calculation, and (2) the two-stage factorization method proposed in Chapter 4. In the cases of standard form and stabilized SQP it is shown how post-convexification can be computed so as to preserve optimality of the primal-dual subproblem solution. This is crucial because it means that post-convexification can be applied *implicitly* by shifting the multipliers, without the need to re-solve the QP subproblem or make any matrix modifications. In the primal-dual SQP case, the advantages and limitations of several approaches to post-convexification are considered, including implicit convexification by shifting the multipliers in the merit function.

With respect to practical application, our attention will be restricted to the primal-dual context. In Chapter 6, a new exact second-derivative primal-dual SQP method (`dcpdSQP`) is formulated for large-scale nonconvex optimization. `dcpdSQP` is an extension of a stabilized, primal-dual

method (pdSQP) described in Chapter 3 that utilizes dynamic convexification. An extension of the convergence analysis of Gill, Kungurtsev and Robinson [32, 33] is used to show that dcpdSQP exhibits superlinear local convergence with guaranteed global convergence. Numerical results from a state-of-the-art implementation indicate that the performance of dcpdSQP is comparable or better than pdSQP while significantly reducing the number of factorizations required.

Chapter 2

Sequential Quadratic Programming

2.1 Local Properties of SQP Methods

In many introductory texts, "*the*" SQP method is defined as one in which the quadratic programming subproblem involves the minimization of a quadratic model of the objective function subject to a linearization of the constraints. This description, which broadly defines the original SQP method of Wilson [69] for convex programming, is somewhat over-simplistic for modern SQP methods. Nevertheless, we start by defining a "vanilla" or "plain" SQP method in these terms.

The basic structure of an SQP method involves *inner* and *outer* iterations. Associated with the kth outer iteration is an approximate solution x_k, together with dual variables y_k and z_k for the nonlinear constraints and bounds. Given (x_k, y_k, z_k), new primal-dual estimates are

computed by solving the quadratic programming subproblem

$$\begin{array}{ll} \underset{x\in\mathbb{R}^n}{\text{minimize}} & f(x_k) + g(x_k)^{\mathrm{T}}(x - x_k) + \frac{1}{2}(x - x_k)^{\mathrm{T}} H(x_k, y_k)(x - x_k) \\ \text{subject to} & c(x_k) + J(x_k)(x - x_k) = 0, \quad x \geq 0. \end{array} \tag{2.1}$$

In our plain SQP method, this subproblem is solved by iteration using a quadratic programming method. New estimates y_{k+1} and z_{k+1} of the Lagrange multipliers are the optimal multipliers for the subproblem (2.1). The iterations of the QP method constitute the SQP inner iterations.

The form of the plain QP subproblem (2.1) is motivated by a certain *fixed-point property* that requires the SQP method to terminate in only one (outer) iteration when started at an optimal solution. In particular, the plain QP subproblem is defined in such a way that if $(x_k, y_k, z_k) = (x^*, y^*, z^*)$, then the NLP primal-dual solution (x^*, y^*, z^*) satisfies the QP optimality conditions for (2.1) and thereby constitutes a solution of the subproblem (see Section 2.1.2 below for a statement of the QP optimality conditions). Under certain assumptions on the problem derivatives, this fixed-point property implies that $(x_k, y_k, z_k) \to (x^*, y^*, z^*)$ when the initial point (x_0, y_0, z_0) is sufficiently close to (x^*, y^*, z^*). These assumptions are discussed further below.

Given our earlier statement that SQP methods "minimize a quadratic model of the objective function", readers unfamiliar with SQP methods might wonder why the quadratic term of the quadratic objective of (2.1) involves the Hessian of the Lagrangian function and not the Hessian of the objective function. However, at $(x_k, y_k, z_k) = (x^*, y^*, z^*)$, the objective of the subproblem defines the second-order local variation of f *on the constraint surface* $c(x) = 0$. Suppose that $x(\alpha)$ is a twice-differentiable feasible path starting at x_k, parameterized by a nonnegative scalar α; i.e., $x(0) = x_k$ and $c(x(\alpha)) = 0$. An inspection of the derivatives $f'(x(\alpha))$ and $f''(x(\alpha))$ at $\alpha = 0$

indicates that the function

$$\varphi_k(x) = f(x_k) + g(x_k)^{\mathrm{T}}\big(x - x_k\big) + \tfrac{1}{2}\big(x - x_k\big)^{\mathrm{T}} H(x_k, y_k)\big(x - x_k\big) \tag{2.2}$$

defines a second-order approximation of f for all x lying on $x(\alpha)$, i.e., $\varphi_k(x)$ may be regarded as a *local quadratic model of f that incorporates the curvature of the constraints $c(x) = 0$.*

This constrained variation of the objective is equivalent to the *unconstrained* variation of a function known as the *modified Lagrangian*, which is given by

$$L(x; x_k, y_k) = f(x) - y_k^{\mathrm{T}}(c(x) - \widehat{c}_k(x)), \tag{2.3}$$

where $\widehat{c}_k(x)$ denotes the vector of linearized constraint functions $\widehat{c}_k(x) = c(x_k) + J(x_k)\big(x - x_k\big)$, and $c(x) - \widehat{c}_k(x)$ is known as the *departure from linearity* (see Robinson [63] and Van der Hoek [67]). The first and second derivatives of the modified Lagrangian are given by

$$\begin{aligned}
\nabla L(x; x_k, y_k) &= g(x) - \big(J(x) - J(x_k)\big)^{\mathrm{T}} y_k, \\
\nabla^2 L(x; x_k, y_k) &= \nabla^2 f(x) - \sum_{i=1}^{m} [\, y_k \,]_i \nabla^2 c_i(x).
\end{aligned}$$

The Hessian of the modified Lagrangian is independent of x_k and coincides with the Hessian (with respect to x) of the conventional Lagrangian. Also, $L(x; x_k, y_k)|_{x=x_k} = f(x_k)$, and $\nabla L(x; x_k, y_k)|_{x=x_k} = g(x_k)$, which implies that $\widehat{f}_k(x)$ defines a local quadratic model of $L(x; x_k, y_k)$ at $x = x_k$.

Throughout the remaining discussion, g_k, c_k, J_k and H_k denote $g(x)$, $c(x)$, $J(x)$ and $H(x, y)$ evaluated at x_k and y_k. With this notation, the quadratic objective is $\varphi_k(x) = f_k + g_k^{\mathrm{T}}(x - x_k) + \frac{1}{2}(x - x_k)^{\mathrm{T}} H_k(x - x_k)$, with gradient $\widehat{g}_k(x) = g_k + H_k(x - x_k)$. A "hat" will be used to denote

quantities associated with the QP subproblem.

2.1.1 Equality constraints

We motivate some of the later discussion by first focusing on equality-constrained nonlinear programming and reviewing the SQP methods available for solving the problem

$$\underset{x\in\mathbb{R}^n}{\text{minimize}} \;\; f(x) \quad \text{subject to} \;\; c(x) = 0, \tag{2.4}$$

which is simply (NP) with the nonnegativity constraints omitted.

Newton's method and SQP

We begin by investigating the connection between SQP methods and Newton's method for solving a system of nonlinear equations, which is the basis of *the method of Newton-Lagrange*. In the case of unconstrained optimization, a standard approach to the formulation of algorithms is to use the first-order optimality conditions to define a system of nonlinear equations $\nabla f(x) = 0$ whose solution is a first-order optimal point x^*. In the constrained case, the relevant nonlinear equations involve the gradient of the Lagrangian function $L(x, y)$, which incorporates the first-order feasibility and optimality conditions satisfied by x^* and y^*. If the rows of the constraint Jacobian J at x^* are linearly independent, a primal-dual solution represented by the $n + m$ vector (x^*, y^*) must satisfy the $n + m$ nonlinear equations $F(x, y) = 0$, where

$$F(x, y) \equiv \nabla L(x, y) = \begin{pmatrix} g(x) - J(x)^{\mathrm{T}} y \\ -c(x) \end{pmatrix}. \tag{2.5}$$

These equations may be solved efficiently using Newton's method.

Consider one iteration of Newton's method, starting at estimates x_k and y_k of the primal and dual variables. If v_k denotes the iterate defined by $(n+m)$-vector (x_k, y_k), then the next iterate v_{k+1} is given by

$$v_{k+1} = v_k + d_k, \quad \text{where} \quad F'(v_k)d_k = -F(v_k).$$

Differentiating (2.5) with respect to x and y gives $F'(v) \equiv F'(x, y)$ as

$$F'(x, y) = \begin{pmatrix} H(x, y) & -J(x)^{\mathrm{T}} \\ -J(x) & 0 \end{pmatrix},$$

which implies that the Newton equations may be written as

$$\begin{pmatrix} H_k & -J_k^{\mathrm{T}} \\ -J_k & 0 \end{pmatrix} \begin{pmatrix} p_k \\ q_k \end{pmatrix} = - \begin{pmatrix} g_k - J_k^{\mathrm{T}} y_k \\ -c_k \end{pmatrix},$$

where p_k and q_k denote the Newton steps for the primal and dual variables. If the second block of equations is scaled by -1 we obtain the system

$$\begin{pmatrix} H_k & -J_k^{\mathrm{T}} \\ J_k & 0 \end{pmatrix} \begin{pmatrix} p_k \\ q_k \end{pmatrix} = - \begin{pmatrix} g_k - J_k^{\mathrm{T}} y_k \\ c_k \end{pmatrix}, \tag{2.6}$$

which is an example of a *saddle-point system*. Finally, if the second block of variables is scaled by -1 we obtain an equivalent symmetric system

$$\begin{pmatrix} H_k & J_k^{\mathrm{T}} \\ J_k & 0 \end{pmatrix} \begin{pmatrix} p_k \\ -q_k \end{pmatrix} = - \begin{pmatrix} g_k - J_k^{\mathrm{T}} y_k \\ c_k \end{pmatrix}, \tag{2.7}$$

which is often referred to as the *KKT system*.

It may not be clear immediately how this method is related to an SQP method. The crucial

link follows from the observation that the KKT equations (2.7) represent the first-order optimality conditions for the primal and dual solution (p_k, q_k) of the quadratic program

$$\begin{aligned} \underset{p\in\mathbb{R}^n}{\text{minimize}} \quad & (g_k - J_k^{\mathrm{T}} y_k)^{\mathrm{T}} p + \tfrac{1}{2} p^{\mathrm{T}} H_k p \\ \text{subject to} \quad & c_k + J_k p = 0, \end{aligned}$$

which, under certain conditions on the curvature of the Lagrangian discussed below, defines the step from x_k to the point that minimizes the local quadratic model of the objective function subject to the linearized constraints. It is now a simple matter to include the constant objective term f_k (which does not affect the optimal solution) and write the dual variables in terms of $y_{k+1} = y_k + q_k$ instead of q_k. The equations analogous to (2.7) are then

$$\begin{pmatrix} H_k & J_k^{\mathrm{T}} \\ J_k & 0 \end{pmatrix} \begin{pmatrix} p_k \\ -y_{k+1} \end{pmatrix} = - \begin{pmatrix} g_k \\ c_k \end{pmatrix}, \tag{2.8}$$

which are the first-order optimality conditions for the quadratic program

$$\underset{p\in\mathbb{R}^n}{\text{minimize}} \;\; f_k + g_k^{\mathrm{T}} p + \tfrac{1}{2} p^{\mathrm{T}} H_k p \quad \text{subject to} \;\; c_k + J_k p = 0.$$

When written in terms of the x variables, this quadratic program is

$$\begin{aligned} \underset{x\in\mathbb{R}^n}{\text{minimize}} \quad & f_k + g_k^{\mathrm{T}} (x - x_k) + \tfrac{1}{2} (x - x_k)^{\mathrm{T}} H_k (x - x_k) \\ \text{subject to} \quad & c_k + J_k (x - x_k) = 0. \end{aligned} \tag{2.9}$$

Local convergence

A standard analysis of Newton's method (see, e.g., Moré and Sorensen [56, Theorem 2.8]) shows that if the KKT matrix is nonsingular at a solution (x^*, y^*), and (x_0, y_0) lies in a sufficiently small neighborhood of (x^*, y^*) in which f and c are twice-continuously differentiable, then the SQP iterates (x_k, y_k) will converge to (x^*, y^*) at a Q-superlinear rate. If, in addition, $H(x, y)$ is locally Lipschitz continuous, then the SQP iterates (x_k, y_k) are Q-quadratically convergent. As x is only a subvector of v, with $v = (x, y)$, the convergence rate of x_k does not follow immediately. However, as $\|x_k - x^*\| \leq \|v_k - v^*\|$, a Q-quadratic rate of convergence of (x_k, y_k) implies an R-quadratic rate of convergence of x_k. For more on the rate of convergence of $\{x_k\}$ relative to $\{x_k, y_k\}$, see Ortega and Rheinboldt [58, Chapter 9].

Conditions for the nonsingularity of the KKT matrix may be determined by transforming the KKT system into an equivalent system that reveals the rank. If Q_k is an $n \times n$ nonsingular matrix, then (2.8) is equivalent to the system

$$\begin{pmatrix} Q_k^{\mathrm{T}} H_k Q_k & (J_k Q_k)^{\mathrm{T}} \\ J_k Q_k & 0 \end{pmatrix} \begin{pmatrix} p_Q \\ -y_{k+1} \end{pmatrix} = - \begin{pmatrix} Q_k^{\mathrm{T}} g_k \\ c_k \end{pmatrix}, \quad \text{with } \; p_k = Q_k p_Q. \tag{2.10}$$

Let Q_k be defined so that $J_k Q_k = \begin{pmatrix} 0 & U_k \end{pmatrix}$, where U_k is $m \times m$. The assumption that J_k has rank m implies that U_k is nonsingular. If the n columns of Q_k are partitioned into blocks Z_k and Y_k of dimension $n \times (n - m)$ and $n \times m$, then

$$J_k Q_k = J_k \begin{pmatrix} Z_k & Y_k \end{pmatrix} = \begin{pmatrix} 0 & U_k \end{pmatrix}, \tag{2.11}$$

which shows that $J_k Z_k = 0$ and $J_k Y_k = U_k$. As Z_k and Y_k are sections of the nonsingular matrix Q_k, they must have independent columns, and, in particular, the columns of Z_k must form a basis

for the null-space of J_k. If $Q_k^{\mathrm{T}} H_k Q_k$ and $J_k Q_k$ are partitioned to conform to the Z–Y partition of Q_k, we obtain the block lower-triangular system

$$\begin{pmatrix} U_k & 0 & 0 \\ Z_k^{\mathrm{T}} H_k Y_k & Z_k^{\mathrm{T}} H_k Z_k & 0 \\ Y_k^{\mathrm{T}} H_k Y_k & Y_k^{\mathrm{T}} H_k Z_k & U_k^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} p_Y \\ p_Z \\ -y_{k+1} \end{pmatrix} = - \begin{pmatrix} c_k \\ Z_k^{\mathrm{T}} g_k \\ Y_k^{\mathrm{T}} g_k \end{pmatrix}, \tag{2.12}$$

where the $(n-m)$-vector p_Z and m-vector p_Y are the parts of p_Q that conform to the columns of Z_k and Y_k. It follows immediately from (2.12) that the Jacobian $F'(x_k, y_k)$ is nonsingular if J_k has independent rows and $Z_k^{\mathrm{T}} H_k Z_k$ is nonsingular. In what follows, we use standard terminology and refer to the vector $Z_k^{\mathrm{T}} g_k$ as the *reduced gradient* and the matrix $Z_k^{\mathrm{T}} H_k Z_k$ as the *reduced Hessian*. If $J(x^*)$ has rank m and the columns of the matrix Z^* form a basis for the null-space of $J(x^*)$, then the conditions: (i) $\nabla L(x^*, y^*) = 0$; and (ii) $Z^{*\mathrm{T}} H(x^*, y^*) Z^*$ positive definite, are sufficient for x^* to be an isolated minimizer of the equality constraint problem (2.4).

Properties of the Newton step

The equations (2.12) have a geometrical interpretation that provides some insight into the properties of the Newton direction. From (2.10), the vectors p_Z and p_Y must satisfy

$$p_k = Q_k p_Q = \begin{pmatrix} Z_k & Y_k \end{pmatrix} \begin{pmatrix} p_Z \\ p_Y \end{pmatrix} = Z_k p_Z + Y_k p_Y.$$

Using block substitution on the system (2.12) we obtain the following equations for p_k and y_{k+1}:

$$\begin{aligned} U_k p_Y &= -c_k, & p_N &= Y_k p_Y, \\ Z_k^{\mathrm{T}} H_k Z_k p_Z &= -Z_k^{\mathrm{T}}(g_k + H_k p_N), & p_T &= Z_k p_Z, \\ p_k = p_N + p_T, \qquad U_k^{\mathrm{T}} y_{k+1} &= Y_k^{\mathrm{T}}(g_k + H_k p_k). \end{aligned} \tag{2.13}$$

These equations involve the auxiliary vectors p_N and p_T such that $p_k = p_N + p_T$ and $J_k p_T = 0$. We call p_N and p_T the *normal* and *tangential* steps associated with p_k. Equations (2.13) may be simplified further by introducing the intermediate vector x_F such that $x_F = x_k + p_N$. The definition of the gradient of φ_k implies that $g_k + H_k p_N = \nabla\varphi_k(x_k + p_N) = \widehat{g}_k(x_F)$, which allows us to rewrite (2.13) in the form

$$\begin{aligned} U_k p_Y &= -c_k, & p_N &= Y_k p_Y, \\ x_F = x_k + p_N, \quad Z_k^{\mathrm{T}} H_k Z_k p_Z &= -Z_k^{\mathrm{T}} \widehat{g}_k(x_F), & p_T &= Z_k p_Z, \\ p_k = p_N + p_T, \qquad x_{k+1} &= x_F + p_T, & & \\ U_k^{\mathrm{T}} y_{k+1} &= Y_k^{\mathrm{T}} \widehat{g}_k(x_{k+1}). & & \end{aligned} \tag{2.14}$$

The definition of x_F implies that

$$\widehat{c}_k(x_F) = c_k + J_k(x_F - x_k) = c_k + J_k p_N = c_k + J_k Y_k p_Y = c_k + U_k p_Y = 0,$$

which implies that the normal component p_N satisfies $J_k p_N = -c_k$ and constitutes the Newton step from x_k to the point x_F satisfying the linearized constraints $c_k + J_k(x - x_k) = 0$. On the other hand, the tangential step p_T satisfies $p_T = Z_k p_Z$, where $Z_k^{\mathrm{T}} H_k Z_k p_Z = -Z_k^{\mathrm{T}} \widehat{g}_k(x_F)$. If the reduced Hessian $Z_k^{\mathrm{T}} H_k Z_k$ is positive definite, which will be the case if x_k is sufficiently close to a locally unique (i.e., isolated) minimizer of (2.4), then p_T defines the Newton step from x_F to the *minimizer* of the quadratic model $\varphi_k(x)$ in the subspace orthogonal to the constraint normals (i.e., on the surface of the linearized constraint $\widehat{c}_k(x) = 0$). It follows that the Newton direction is the sum of two steps: a normal step to the linearized constraint and the tangential step on the constraint surface that minimizes the quadratic model. This property reflects the two (usually conflicting)

underlying processes present in all algorithms for optimization—the minimization of the objective and the satisfaction of the constraints.

In the discussion above, the normal step p_N is interpreted as a Newton direction for the equations $\widehat{c}_k(x) = 0$ at $x = x_k$. However, in some situations, p_N may also be interpreted as the solution of a minimization problem. The Newton direction p_k is unique, but the decomposition $p_k = p_T + p_N$ depends on the choice of the matrix Q_k associated with the Jacobian factorization (2.11). If Q_k is orthogonal, i.e., if $Q_k^{\mathrm{T}} Q_k = I$, then $Z_k^{\mathrm{T}} Y_k = 0$ and the columns of Y_k form a basis for the range space of J_k^{T}. In this case, p_N and p_T define the unique range-space and null-space decomposition of p_k, and p_N is the unique solution with least two-norm of the least-squares problem

$$\min_p \ \|\widehat{c}_k(x_k) + J_k p\|_2, \quad \text{or, equivalently,} \quad \min_p \ \|c_k + J_k p\|_2.$$

This interpretation is useful in the formulation of variants of Newton's method that do not require (x_k, y_k) to lie in a small neighborhood of (x^*, y^*). In particular, it suggests a way of computing the normal step when the equations $J_k p = -c_k$ are not compatible.

For consistency with the inequality constrained case below, the primal-dual solution of the kth QP subproblem is denoted by $(\widehat{x}_k, \widehat{y}_k)$. With this notation, the first-order optimality conditions for the QP subproblem (2.9) are given by

$$\begin{aligned} J_k(\widehat{x}_k - x_k) + c_k &= 0, \\ g_k + H_k(\widehat{x}_k - x_k) - J_k^{\mathrm{T}} \widehat{y}_k &= 0. \end{aligned} \tag{2.15}$$

Similarly, the Newton iterates are given by $x_{k+1} = \widehat{x}_k = x_k + p_k$ and $y_{k+1} = \widehat{y}_k = y_k + q_k$.

Calculation of the Newton step

There are two broad approaches for solving the Newton equations (either in saddle-point form (2.6) or symmetric form (2.7). The first involves solving the full $n+m$ set of KKT equations, the second decomposes the KKT equations into the three systems associated with the block lower-triangular equations (2.12).

In the full-matrix approach, the matrix K may be represented by its *symmetric indefinite factorization* (see, e.g., Bunch and Parlett [7], and Bunch and Kaufman [6]):

$$PKP^{\mathrm{T}} = LDL^{\mathrm{T}}, \tag{2.16}$$

where P is a permutation matrix, L is lower triangular and D is block diagonal, with 1×1 or 2×2 blocks. (The latter are required to retain numerical stability.) Some prominent software packages include MA27 (Duff and Reid [19]), MA57 (Duff [18]), MUMPS (Amestoy et al. [1]), PARDISO (Schenk and Gärtner [65]), and SPOOLES (Ashcraft and Grimes [3]).

The decomposition approach is based on using an explicit or implicit representation of the null-space basis matrix Z_k. When J_k is dense, Z_k is usually computed directly from a QR factorization of J_k (see, e.g., Coleman and Sorensen [10], and Gill et al. [36]). When J_k is sparse, however, known techniques for obtaining an orthogonal *sparse* Z may be expensive in time and storage, although some effective algorithms have been proposed (see, e.g., Coleman and Pothen [9]; Gilbert and Heath [31]).

The representation of Z_k most commonly used in sparse problems is called the *variable-reduction* form of Z_k, and is obtained as follows. The columns of J_k are partitioned so as to identify explicitly an $m \times m$ nonsingular matrix B (the *basis matrix*). Assuming that the columns are

permuted so that B is at the "left" of J_k, we have

$$J_k = \begin{pmatrix} B & S \end{pmatrix}.$$

(In practice, the columns of B may occur anywhere.) When J_k has this form, a basis for the null space of J_k is given by the columns of the (non-orthogonal) matrix Q_k defined as

$$Q_k = \begin{pmatrix} Z_k & Y_k \end{pmatrix}, \text{ with } Z_k = \begin{pmatrix} -B^{-1}S \\ \\ I_{n-m} \end{pmatrix} \text{ and } Y_k = \begin{pmatrix} I_m \\ \\ 0 \end{pmatrix}.$$

This definition of Q_k means that matrix-vector products $Z_k^{\mathrm{T}}v$ or $Z_k v$ can be computed using a factorization of B (typically, a sparse LU factorization; see Gill, Murray, Saunders and Wright [37]). The matrix Z_k is not stored explicitly.

For large sparse problems, the reduced Hessian $Z_k^{\mathrm{T}} H_k Z_k$ associated with the solution of (2.14) will generally be much more dense than H_k and B. However, in many cases, $n - m$ is small enough to allow the storage of a dense Cholesky factor of $Z_k^{\mathrm{T}} H_k Z_k$.

Merit function line-search methods

The convergence result discussed in Section 2.1.1 requires that (x_0, y_0) lies in a sufficiently small neighborhood of (x^*, y^*) because the method of Newton-Lagrange alone is not able to guarantee convergence to a local minimizer from any starting point. As was alluded to in Section 1.3, one notable strategy for forcing convergence is to designate a *merit function* M whose value measures the distance to a local minimizer. This function may be used in conjunction with a line search model

function to force convergence. Popular choices for M include the ℓ_1 and ℓ_∞ penalty functions

$$P_1(x\,;\mu) = f(x) + \frac{1}{\mu}\|c(x)\|_1 \qquad \text{and} \qquad P_\infty(x\,;\mu) = f(x) + \frac{1}{\mu}\|c(x)\|_\infty. \tag{2.17}$$

Although these penalty functions are nonsmooth, they have the benefit of being *exact* in the sense that for μ sufficiently small, x^* is an unconstrained local minimizer of $P_1(x\,;\mu)$ and $P_\infty(x\,;\mu)$.

The search direction $p_k = \widehat{x} - x_k$ is computed from the solution $\widehat{x}$ of the subproblem

$$\begin{aligned} \underset{x\in\mathbb{R}^n}{\text{minimize}} \quad & f_k + g_k^{\mathrm{T}}(x - x_k) + \tfrac{1}{2}(x - x_k)^{\mathrm{T}}\widehat{H}_k(x - x_k) \\ \text{subject to} \quad & c_k + J_k(x - x_k) = 0, \end{aligned} \tag{2.18}$$

where $\widehat{H}_k$ is a positive-definite approximation of the Hessian of the Lagrangian H_k, and the variables are then updated as $x_{k+1} = x_k + \alpha_k p_k$ where α_k is a positive scalar. A line search is performed to choose a step α_k such that the reduction in the merit function is at least a factor η_s of the reduction predicted by a model function m_k of the merit function. In the case of the ℓ_1 penalty function, the line search computes a step α_k such that

$$P(x_k\,;\mu) - P(x_k + \alpha p_k\,;\mu) \geq \eta_s(m_k(x_k\,;\mu) - m_k(x_k + \alpha p_k\,;\mu)) \tag{2.19}$$

with $\alpha = \alpha_k$. An appropriate model of the P_1 merit function at x_k is

$$m_k(x\,;\mu) = f_k + g_k^{\mathrm{T}}(x - x_k) + \frac{1}{\mu}\|c_k + J_k(x - x_k)\|_1,$$

which is simply $P_1(x\,;\mu)$ composed with *affine* models of $f(x)$ and $c(x)$. A critical property of this choice of merit and model function is summarized in the following result.

Result 2.1.1. *Let p_k denote the solution and $\widehat{y}_k$ the Lagrange multipliers of the convex QP subproblem* (2.18)*, where J_k has rank m and $\widehat{H}_k$ is positive definite. Let η_s be a scalar such that $0 < \eta_s < \frac{1}{2}$. If p_k is nonzero and $\mu \leq 1/\|\widehat{y}_k\|_\infty$, there exists $\bar{\alpha} > 0$ such that* (2.19) *holds for all $\alpha \in (0, \bar{\alpha})$.*

For details and a similar result for the ℓ_∞ merit function $P_\infty(x\,;\mu)$, see Gill and Wright [44].

Methods for determining the positive-definite approximate Lagrangian Hessian $\widehat{H}_k$ typically fall into one of two categories, depending on the order of derivatives available. If second derivatives can be computed, there are several ways to compute a positive-definite matrix $\widehat{H}_k \approx H_k$, such as the modified Cholesky method or the method of Wächter and Biegler [68]. If second derivatives are not available then a quasi-Newton approximation can be used. This approach maintains an approximation to the Hessian of the Lagrangian function via the update formula

$$\widehat{H}_{k+1} = \widehat{H}_k - \frac{1}{d_k^{\mathrm{T}} \widehat{H}_k d_k} \widehat{H}_k d_k d_k^{\mathrm{T}} \widehat{H}_k + \frac{1}{w_k^{\mathrm{T}} d_k} w_k w_k^{\mathrm{T}},$$

where $d_k = x_{k+1} - x_k$ and $w_k = \nabla L(x_{k+1}, \widehat{y}) - \nabla L(x_k, \widehat{y})$. Besides the disadvantage of not using second derivatives, the update formula does not preserve sparsity and so has limited utility for large sparse problems.

The merit functions $P_1(x\,;\mu)$ and $P_\infty(x\,;\mu)$ proposed so far in this section suffer from *the Maratos effect*, which refers to situation in which the unit step is not accepted by the line search. This has an adverse effect on the rate of convergence because Newton's method requires $\alpha_k = 1$ for a superlinear rate of convergence. However, this issue does not apply to most smooth merit functions. Some smooth merit functions that have been proposed include the Fletcher augmented

Lagrangian

$$M(x\,;\mu) = f(x) - y(x)^{\mathrm{T}}c(x) + \frac{1}{2\mu}\|c(x)\|^2, \;\; \text{where} \;\; y(x) = (J(x)J(x)^{\mathrm{T}})^{-1}J(x)g(x),$$

i.e., $y(x) = \operatorname{argmin} \|g(x) - J(x)^{\mathrm{T}}y\|$. Other choices include the quadratic penalty function

$$P_2(x\,;\mu) = f(x) + \frac{1}{2\mu}\|c(x)\|^2,$$

and the primal-dual augmented Lagrangian

$$M(x, y\,;\mu) = f(x) - y^{\mathrm{T}}c(x) + \frac{1}{2\mu}\|c(x)\|^2.$$

In the case of the primal-dual merit function, a line search determines the step for both primal and dual variables $x_{k+1} = x_k + \alpha_k p_k$ and $y_{k+1} = y_k + \alpha_k q_k$, where $q_k = \widehat{y}_k - y_k$.

Sequential unconstrained methods

The methods described next are different in the sense that the search direction is not computed from the QP subproblem (2.18). Nonetheless, these methods still involve a sequence of subproblems that can be expressed as a quadratic program. Sequential unconstrained methods minimize a penalty function for a sequence of decreasing parameters $\mu > 0$. The penalty functions $P_1(x\,;\mu)$, $P_2(x\,;\mu)$, and $P_\infty(x\,;\mu)$ can all be minimized by solving a sequence of unconstrained subproblems. We will focus on the ℓ_1 penalty function, which can be approximated by the local

quadratic model $m_k(x\,;\mu) \approx P_1(x\,;\mu)$ with

$$m_k(x\,;\mu) = g_k^{\mathrm{T}}(x - x_k) + \frac{1}{2}(x - x_k)^{\mathrm{T}} H_k(x - x_k) + \frac{1}{\mu}\|c_k + J_k(x - x_k)\|_1.$$

The corresponding unconstrained subproblem is therefore

$$\underset{p\in\mathbb{R}^n}{\text{minimize}} \quad g_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} H_k p + \frac{1}{\mu}\|c_k + J_k p\|_1. \tag{2.20}$$

A crucial property of (2.20) is that it is equivalent to the smooth constrained subproblem

$$\begin{aligned} \underset{p,u,v}{\text{minimize}} \quad & g_k^{\mathrm{T}} p + \tfrac{1}{2} p^{\mathrm{T}} H_k p + \tfrac{1}{\mu} e^{\mathrm{T}}(u + v) \\ \text{subject to} \quad & c_k + J_k p - u + v = 0, \quad u \geq 0, \quad v \geq 0, \end{aligned} \tag{2.21}$$

which is derived by writing $c_k + J_k p$ as the difference of two positive functions $c_k + J_k p = u - v$ where $u = (c_k + J_k p)^+$ and $v = (c_k + J_k p)^-$, so that $e^{\mathrm{T}}(u + v) = \|c_k + J_k p\|_1$. To ensure global convergence, $P_1(x\,;\mu)$ can be minimized using a line-search or trust-region strategy. Using a line search requires obtaining a positive-definite approximation $\widehat{H}_k \approx H_k$ with which to form (2.21). If a trust-region is used then the exact Lagrangian Hessian requires no modification. Instead, a limit on the norm of p is imposed on the subproblems (2.21), giving the additional constraint $\|p\|_\infty \leq \delta_k$, where δ_k is the *trust-region radius*. The resulting QP subproblem is then

$$\begin{aligned} \underset{p,u,v}{\text{minimize}} \quad & g_k^{\mathrm{T}} p + \tfrac{1}{2} p^{\mathrm{T}} H_k p + \tfrac{1}{\mu} e^{\mathrm{T}}(u + v) \\ \text{subject to} \quad & c_k + J_k p - u + v = 0, \\ & u \geq 0, \quad v \geq 0, \quad -\delta_k e \leq p \leq \delta_k e. \end{aligned} \tag{2.22}$$

Similar to the line-search method, the reduction achieved in $P_1(x\,;\mu)$ is compared with that predicted by the model $m_k(x\,;\mu)$, and the comparison is used to update the trust-region radius δ_k.

Constraint regularization

If the rows of J_k are linearly dependent then the first-order optimality condition equations (2.8) for the equality constrained QP (2.9) are singular. One way to address this is to perturb the problem functions $f(x)$ and $c(x)$ so that the nonsingularity of the resulting equations does not depend on the rank of J_k. In order to do this, it is necessary to allow the functions to depend on both primal and dual variables so that the rows of the perturbed constraint Jacobian are guaranteed to be independent.

Explicitly, consider the shifted constraints $\widetilde{c}(x,y) = c(x) + \mu(y - y_k)$, where μ is a small positive parameter. The resulting Jacobian is $\widetilde{J}(x,y) = \begin{pmatrix} J(x) & \mu I \end{pmatrix}$, which has linearly independent rows regardless of the rank of $J(x)$. To limit the magnitude of the perturbation it is necessary to limit the norm of y. This may be done by augmenting the objective function by a two-norm penalty term, giving $\widetilde{f}(x,y) = f(x) + \frac{1}{2}\mu\|y\|^2$. Observe that the quadratic objective based on $\widetilde{f}$ and $\widetilde{c}$ has the form

$$\widetilde{\varphi}_k(v) = \widetilde{f}(v_k) + \nabla\widetilde{f}(v_k)^{\mathrm{T}}(v - v_k) + \tfrac{1}{2}\mu(v - v_k)^{\mathrm{T}}\widetilde{H}(v - v_k),$$

which, after some simplification, can be written $\widetilde{\varphi}_k(x,y) = \varphi_k(x) + \frac{1}{2}\mu\|y\|^2$. Moreover, the linearization of the shifted constraints is $\widetilde{c}(v_k) + \widetilde{J}(v_k)(v - v_k)$ or simply $c_k + J_k(x - x_k) + \mu(y - y_k)$. Thus, if an equality constrained quadratic program analogous to (2.9) is formed with the regularized

functions it can be expressed as

$$\begin{aligned} \underset{x\in\mathbb{R}^n, y\in\mathbb{R}^m}{\text{minimize}} \quad & f_k + g_k^{\mathrm{T}}(x - x_k) + \tfrac{1}{2}(x - x_k)^{\mathrm{T}} H_k (x - x_k) + \tfrac{1}{2}\mu\|y\|^2 \\ \text{subject to} \quad & c_k + J_k(x - x_k) + \mu(y - y_k) = 0. \end{aligned} \tag{2.23}$$

The first-order KKT conditions for a solution $(\widehat{x},\ \widehat{y},\ \widehat{\pi})$ of the regularized equality-constrained problem (2.23) are given by

$$\begin{aligned} c_k + J_k^{\mathrm{T}}(\widehat{x} - x_k) + \mu(\widehat{y} - y_k) &= 0, \\ g_k + H_k(\widehat{x} - x_k) - J_k^{\mathrm{T}}\widehat{\pi} &= 0, \qquad \mu\widehat{y} = \mu\widehat{\pi}. \end{aligned} \tag{2.24}$$

Using the identity $\widehat{y} = \widehat{\pi}$ to eliminate $\widehat{\pi}$ allows the conditions (2.24) to be expressed as a linear system of equations

$$\begin{pmatrix} H_k & J_k^{\mathrm{T}} \\ J_k & -\mu I \end{pmatrix} \begin{pmatrix} \widehat{x} - x_k \\ -\widehat{y} \end{pmatrix} = - \begin{pmatrix} g_k \\ c_k \end{pmatrix}. \tag{2.25}$$

This system is analogous to (2.8) except the nonsingular perturbation appearing in the $(2,2)$-block of the KKT matrix.

Let $\begin{pmatrix} U & V \end{pmatrix}$ be an orthonormal matrix such that the columns of U form a basis for $\text{null}(J_k^{\mathrm{T}})$ and the columns of V form a basis for $\text{range}(J_k)$. The unique expansion $\widehat{y}_k = Uy_U + Vy_V$ allows us to rewrite (2.25) as

$$\begin{pmatrix} H_k & J_k^{\mathrm{T}}V & \\ V^{\mathrm{T}}J_k & -\mu I & \\ & & -\mu I \end{pmatrix} \begin{pmatrix} p_k \\ -y_V \\ -y_U \end{pmatrix} = - \begin{pmatrix} g_k \\ V^{\mathrm{T}}c_k \\ 0 \end{pmatrix}, \tag{2.26}$$

where $J_k^{\mathrm{T}}U = 0$ from the definition of U, and $U^{\mathrm{T}}c_k = 0$ because $c_k \in \text{range}(J_k)$. The following simple argument shows that the equations (2.26) are nonsingular, regardless of the rank of J_k.

First, observe that $V^{\mathrm{T}}J_k$ has full row rank. Otherwise, if $v^{\mathrm{T}}V^{\mathrm{T}}J_k = 0$, it must be the case that $Vv \in \text{null}(J_k^{\mathrm{T}})$. But as $Vv \in \text{range}(V)$ and $\text{range}(V)$ is orthogonal to $\text{null}(J_k^{\mathrm{T}})$, we conclude that $Vv = 0$, and the linearly independence of the columns of V gives $v = 0$.

Moreover, equations (2.26) imply that $y_U = 0$ and $\widehat{y}_k \in \text{range}(J_k)$. If $g_{k+1} = g_k + Hp_k$, then

$$J_k^{\mathrm{T}}\widehat{y}_k = g_{k+1} \quad \text{and} \quad \widehat{y}_k \in \text{range}(J_k).$$

These are the necessary and sufficient conditions for $\widehat{y}_k$ to be the unique least-length solution of the compatible equations $J_k^{\mathrm{T}}y = g_{k+1}$. This implies that the regularization gives a unique vector of multipliers.

2.1.2 Inequality constraints

Given an approximate primal-dual solution (x_k, y_k) with $x_k \geq 0$, an outer iteration of a typical SQP method involves solving the QP subproblem (2.1), repeated here for convenience:

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad & f_k + g_k^{\mathrm{T}}(x - x_k) + \tfrac{1}{2}(x - x_k)^{\mathrm{T}}H_k(x - x_k) \\ \text{subject to} \quad & J_k(x - x_k) = -c_k, \quad x \geq 0. \end{aligned} \tag{2.27}$$

Assume for the moment that this QP subproblem is feasible, with primal-dual solution $(\widehat{x}_k, \widehat{y}_k, \widehat{z}_k)$. The next plain SQP iterate is $x_{k+1} = \widehat{x}_k$, $y_{k+1} = \widehat{y}_k$ and $z_{k+1} = \widehat{z}_k$. The QP first-order optimality conditions are

$$\begin{aligned} J_k(\widehat{x}_k - x_k) + c_k &= 0, \qquad \widehat{x}_k \geq 0; \\ g_k + H_k(\widehat{x}_k - x_k) - J_k^{\mathrm{T}}\widehat{y}_k - \widehat{z}_k &= 0, \\ \widehat{x}_k \cdot \widehat{z}_k &= 0, \qquad \widehat{z}_k \geq 0. \end{aligned} \tag{2.28}$$

Let $p_k = \widehat{x}_k - x_k$ and let $p_{\mathcal{F}} = [\, p_k \,]_{\mathcal{F}}$ denote the vector of free components of p_k, i.e., the components with indices in $\mathcal{F}(\widehat{x}_k)$. Similarly, let $z_{\mathcal{F}}$ denote the free components of $\widehat{z}_k$. The complementarity conditions imply that $z_{\mathcal{F}} = 0$ and we may combine the first two sets of equalities in (2.28) to give

$$\begin{pmatrix} H_{\mathcal{F}} & J_{\mathcal{F}}^{\mathrm{T}} \\ J_{\mathcal{F}} & 0 \end{pmatrix} \begin{pmatrix} p_{\mathcal{F}} \\ -\widehat{y}_k \end{pmatrix} = - \begin{pmatrix} [\, g_k + H_k \eta_k \,]_{\mathcal{F}} \\ c_k + J_k \eta_k \end{pmatrix}, \tag{2.29}$$

where $J_{\mathcal{F}}$ is the matrix of free columns of J_k, and η_k is the vector

$$[\, \eta_k \,]_i = \begin{cases} [\, \widehat{x}_k - x_k \,]_i & \text{if } i \in \mathcal{A}(\widehat{x}_k); \\ \\ 0 & \text{if } i \in \mathcal{F}(\widehat{x}_k). \end{cases}$$

If the active sets at $\widehat{x}_k$ and x_k are the same, i.e., $\mathcal{A}(\widehat{x}_k) = \mathcal{A}(x_k)$, then $\eta_k = 0$. If $\widehat{x}_k$ lies in a sufficiently small neighborhood of a *nondegenerate* solution x^*, then $\mathcal{A}(\widehat{x}_k) = \mathcal{A}(x^*)$ and hence $J_{\mathcal{F}}$ has full row rank (see Robinson [64]). In this case we say that the QP *identifies the correct active set* at x^*. If, in addition, (x^*, y^*) satisfies the second-order sufficient conditions for optimality, then KKT system (2.29) is nonsingular and the plain SQP method is equivalent to Newton's method applied to the equality-constraint subproblem defined by fixing the variables in the active set at their bounds.

However, at a *degenerate* QP solution, the rows of $J_{\mathcal{F}}$ are linearly dependent and the KKT equations (2.29) are compatible but singular. Broadly speaking, there are two approaches to dealing with the degenerate case, where each approach is linked to the method used to solve the QP subproblem. The first approach employs a QP method that not only finds the QP solution $\widehat{x}_k$, but also identifies a "working set" of variables that defines a Jacobian matrix with *linearly independent* rows. The second approach solves a *regularized* or *perturbed* QP subproblem that

provides a perturbed version of the KKT system (2.29) that is nonsingular for any J_F.

Identifying independent constraints

The first approach is based on using a QP algorithm that provides a primal-dual QP solution that satisfies a *nonsingular* KKT system analogous to (2.29). A class of quadratic programming methods with this property are primal-feasible active-set methods, which form the basis of the software packages NPSOL and SNOPT. Primal-feasible QP methods have two phases: in phase 1, a feasible point is found by minimizing the sum of infeasibilities; in phase 2, the quadratic objective function is minimized while feasibility is maintained. In each iteration, the variables are labeled as being "basic" or "nonbasic", where the nonbasic variables are temporarily fixed at their current value. The indices of the basic and nonbasic variables are denoted by $\mathcal{B}$ and $\mathcal{N}$ respectively. A defining property of the $\mathcal{B}$–$\mathcal{N}$ partition is that the rows of the Jacobian appearing in the KKT matrix are always linearly independent. Once an initial basic set is identified, all subsequent KKT equations have a constraint block with independent rows. (See Section 2.2.1 for more details on primal-feasible active-set methods.)

Let $p_k = \widehat{x}_k - x_k$, where $(\widehat{x}_k, \widehat{y}_k)$ is the QP solution found by a primal-feasible active-set method. Let p_B denote the vector of components of p_k in the final basic set $\mathcal{B}$, with J_B the corresponding columns of J_k. The vector $(p_B, \widehat{y}_k)$ satisfies the *nonsingular* KKT equations

$$\begin{pmatrix} H_B & J_B^{\mathrm{T}} \\ J_B & 0 \end{pmatrix} \begin{pmatrix} p_B \\ -\widehat{y}_k \end{pmatrix} = - \begin{pmatrix} [\, g_k + H_k \eta_k \,]_B \\ c_k + J_k \eta_k \end{pmatrix}, \tag{2.30}$$

where η_k is now defined in terms of the final QP nonbasic set, i.e.,

$$[\,\eta_k\,]_i = \begin{cases} [\,\widehat{x}_k - x_k\,]_i & \text{if } i \in \mathcal{N}; \\ \\ 0 & \text{if } i \notin \mathcal{N}. \end{cases} \tag{2.31}$$

As in (2.29), if the basic-nonbasic partition is not changed during the solution of the subproblem, then $\eta_k = 0$. If this final QP nonbasic set is used to define the initial nonbasic set for the next QP subproblem, it is typical for the later QP subproblems to reach optimality in a *single iteration* because the solution of the first QP KKT system satisfies the QP optimality conditions immediately. In this case, the phase-1 procedure simply performs a feasibility check that would be required in any case.

Constraint regularization

Analogous to the equality-constrained case, constraint regularization can be used to define KKT equations that are nonsingular regardless of the rank of $J_{\scriptscriptstyle F}$. Consider the perturbed version of equations (2.29) such that

$$\begin{pmatrix} H_{\scriptscriptstyle F} & J_{\scriptscriptstyle F}^{\mathrm{T}} \\ J_{\scriptscriptstyle F} & -\mu I \end{pmatrix} \begin{pmatrix} p_{\scriptscriptstyle F} \\ -\widehat{y}_k \end{pmatrix} = - \begin{pmatrix} [\,g_k + H_k\eta_k\,]_{\scriptscriptstyle F} \\ c_k + J_k\eta_k \end{pmatrix}, \tag{2.32}$$

where μ is a small positive constant. In addition, assume that $Z_{\scriptscriptstyle F}^{\mathrm{T}} H_{\scriptscriptstyle F} Z_{\scriptscriptstyle F}$ is positive definite, where the columns of $Z_{\scriptscriptstyle F}$ form a basis for the null space of $J_{\scriptscriptstyle F}$. With this assumption, the regularized system (2.32) is nonsingular regardless of the rank of $J_{\scriptscriptstyle F}$. In contrast, the unperturbed KKT equations (2.29) are singular if and only if $J_{\scriptscriptstyle F}$ has linearly dependent rows.

Note also that the QP subproblem analogous to (2.27) but formed with perturbed problem

functions $\widetilde{f}(x,y)$ and $\widetilde{c}(x,y)$ has the form

$$\begin{array}{ll} \underset{x\in\mathbb{R}^n, y\in\mathbb{R}^m}{\text{minimize}} & f_k + g_k^{\mathrm{T}}(x-x_k) + \frac{1}{2}(x-x_k)^{\mathrm{T}} H_k (x-x_k) + \frac{1}{2}\mu\|y\|^2 \\ \text{subject to} & c_k + J_k(x-x_k) + \mu(y-y_k) = 0, \quad x \geq 0. \end{array} \tag{2.33}$$

When formulated in terms of the free variables, the optimality conditions analogous to (2.28) for the regularized QP yield a system identical to (2.32).

Wright [70, 71, 72] and Hager [49] show that an SQP method using the regularized equations (2.32) will converge at a superlinear rate, even in the degenerate case. In Chapter 3, QP methods are discussed that give equations of the form (2.32) at every outer iteration, not just in the neighborhood of the solution. These methods implicitly shift the constraints by an amount of order μ and give QP multipliers that converge to an $O(\mu)$ estimate of the least-length multipliers.

A related regularization scheme has been proposed and analyzed by Fischer [21], who solves a second QP to obtain the multiplier estimates. Anitescu [2] regularizes the problem by imposing a trust-region constraint on the plain SQP subproblem (2.1) and solving the resulting subproblem by a semidefinite programming method.

Merit function methods

Much of the theory for equality-constrained SQP can be extended to the inequality constrained case. In order to be consistent with the notation of Section 2.1.1 the constraints will be written in the form $c(x) \geq 0$, while keeping in mind what follows also applies to (NP). As in the equality-constrained case, convergence can be forced by using a merit function and a local line-search model function. The primary difference is that the constraint violation is now given by $c^-(x) = -\min(c(x),\ 0)$ rather than $c(x)$. The ℓ_1 norm, infinity norm, and quadratic

penalty functions, as well as the primal-dual augmented Lagrangian can each be adapted to serve as a merit function in this setting. For example, the ℓ_1 penalty function, which now is given by $P_1(x\,;\mu) = f(x) + \frac{1}{\mu}\|c^-(x)\|_1$, can be used in conjunction with an appropriate line-search model given by

$$m_k(x_k\,;\mu) = f_k + g_k^{\mathrm{T}}(x - x_k) + \frac{1}{\mu}\|\big(c_k + J_k(x - x_k)\big)^-\|_1$$

to ensure global convergence. The following result is analogous to Result 2.1.1 and is the foundation of guaranteed convergence.

Result 2.1.2. *Let p_k denote the solution and $\widehat{y}$ the optimal Lagrange multipliers of the QP subproblem*

$$\begin{aligned} &\underset{p\in\mathbb{R}^n}{\text{minimize}} && f_k + g_k^{\mathrm{T}}p + \tfrac{1}{2}p^{\mathrm{T}}\widehat{H}_k p \\ &\text{subject to} && c_k + J_k p \geq 0, \end{aligned} \tag{2.34}$$

where $\widehat{H}_k$ is positive definite. Let η_S be a scalar such that $0 < \eta_S < \frac{1}{2}$. If p_k is nonzero, then for all $\mu \leq 1/\|\widehat{y}\|_\infty$ there exists $\bar{\alpha} > 0$ such that

$$P_1(x_k\,;\mu) - P_1(x_k + \alpha p_k\,;\mu) \geq \eta_s(m_k(x_k\,;\mu) - m_k(x_k + \alpha p\,;\mu))$$

for all $\alpha \in (0, \bar{\alpha})$.

See Gill and Wright [44] for details. As in the equality-constraint case, Hessian convexification or a quasi-Newton approximation can be used to define $\widehat{H}_k$.

As before, sequential unconstrained methods can also be used for minimizing the ℓ_1 penalty function $P_1(x\,;\mu) = f(x) + \frac{1}{\mu}\|c^-(x)\|_1$, which is approximated by the quadratic model

$$m_k(x\,;\mu) = f_k + g_k^{\mathrm{T}}(x - x_k) + \frac{1}{2}(x - x_k)^{\mathrm{T}}H_k(x - x_k) + \frac{1}{\mu}\|(c_k + J_k(x - x_k))^-\|_1.$$

The nonsmooth unconstrained problem of minimizing $m_k(x\,;\mu)$ can be replaced by the equivalent smooth constrained problem

$$\begin{array}{ll} \underset{p\in\mathbb{R}^n, v\in\mathbb{R}^m}{\text{minimize}} & g_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} H_k p + \frac{1}{\mu} e^{\mathrm{T}} v \\ \text{subject to} & c_k + J_k p + v \geq 0, \quad v \geq 0. \end{array} \tag{2.35}$$

This method can be globalized by using a line-search or trust-region strategy. In the line-search case a positive-definite matrix $\widehat{H}_k$ is required to approximate H_k, and a line-search computes α_k such that the actual reduction in $P_1(x\,;\mu)$ is at least fixed fraction of the reduction predicted by the line-search model. In the trust-region case, the exact Hessian H_k can be used, but a trust-region constraint $\|p\|_\infty \leq \delta_k$ must be imposed, with the trust-region radius δ_k chosen so that $P_1(x_k + p_k\,;\mu) < P_1(x_k\,;\mu)$. The resulting QP subproblem is then given by

$$\begin{array}{ll} \underset{p\in\mathbb{R}^n, v\in\mathbb{R}^m}{\text{minimize}} & g_k^{\mathrm{T}} p + \frac{1}{2} p^{\mathrm{T}} H_k p + \frac{1}{\mu} e^{\mathrm{T}} v \\ \text{subject to} & c_k + J_k p + v \geq 0, \quad v \geq 0, \quad -\delta_k e \leq p \leq \delta_k e. \end{array} \tag{2.36}$$

There are alternatives to using a merit functions, though their presentation is outside the scope of this discussion. The class of *filter methods* employ a fundamentally different approach to guarantee global convergence. For more information see, e.g., [44], [25], and [26].

2.2 Methods for Quadratic Programming

We consider methods for the quadratic program

$$\begin{aligned} \underset{x\in\mathbb{R}^n}{\text{minimize}} \quad & g^{\mathrm{T}}(x-x_I)+\tfrac{1}{2}(x-x_I)^{\mathrm{T}}H(x-x_I) \\ \text{subject to} \quad & Ax = Ax_I - b, \quad x \geq 0, \end{aligned} \tag{2.37}$$

where g, H, b, A and x_I are given constant quantities, with H symmetric. The QP objective is denoted by $\varphi(x)$, with gradient $\widehat{g}(x) = g+H(x-x_I)$. In some situations, the general constraints will be written as $\widehat{c}(x) = 0$, with $\widehat{c}(x) = A(x-x_I)+b$. The QP active set is denoted by $\mathcal{A}(x)$. A primal-dual QP solution is denoted by (x^*, y^*, z^*). In terms of the QP defined at the kth outer iteration of an SQP method, we have $x_I = x_k$, $b = c(x_k)$, $g = g(x_k)$, $A = J(x_k)$ and $H = H(x_k, y_k)$. It is assumed that A has rank m. No assumptions are made about H other than symmetry. Conditions that must hold at an optimal solution of (2.37) are provided by the following result (see, e.g., Borwein [4], Contesse [14] and Majthay [53]).

Result 2.2.1 (QP optimality conditions)**.**

The point x^* is a local minimizer of the quadratic program (2.37) *if and only if*

(a) $\widehat{c}(x^*) = 0$, $x^* \geq 0$, *and there exists at least one pair of vectors y^* and z^* such that* $\widehat{g}(x^*) - A^{\mathrm{T}}y^* - z^* = 0$, *with* $z^* \geq 0$, *and* $z^* \cdot x^* = 0$;

(b) $p^{\mathrm{T}}Hp \geq 0$ *for all nonzero p satisfying* $\widehat{g}(x^*)^{\mathrm{T}}p = 0$, $Ap = 0$, *and* $p_i \geq 0$ *for every* $i \in \mathcal{A}(x^*)$.

Part (a) gives the first-order KKT conditions (2.28) for the QP (2.37). If H is positive semidefinite, the first-order KKT conditions are both necessary and sufficient for (x^*, y^*, z^*) to be a local primal-dual solution of (2.37).

Suppose that (x^*, y^*, z^*) satisfies condition (a) with $z_i^* = 0$ and $x_i^* = 0$ for some i. If H is positive semidefinite, then x^* is a *weak minimizer* of (2.37). In this case, x^* is a global minimizer with a unique global minimum $\varphi(x^*)$. If H has at least one negative eigenvalue, then x^* is known as a *dead point*. Verifying condition (b) at a dead point requires finding the global minimizer of an indefinite quadratic form over a cone, which is an NP-hard problem (see, e.g., Cottle, Habetler and Lemke [15], Pardalos and Schnitger [59], and Pardalos and Vavasis [60]). This implies that the optimality of a candidate solution of a general quadratic program can be verified only if more restrictive (but computationally tractable) sufficient conditions are satisfied. A dead point is a point at which the sufficient conditions are not satisfied, but certain necessary conditions hold. Computationally tractable necessary conditions are based on the following result.

Result 2.2.2 (Necessary conditions for optimality)**.**

The point x^* is a local minimizer of the QP (2.37) *only if*

(a) $\widehat{c}(x^*) = 0$, $x^* \geq 0$, *and there exists at least one pair of vectors y^* and z^* such that* $\widehat{g}(x^*) - A^{\mathrm{T}}y^* - z^* = 0$, *with* $z^* \geq 0$, *and* $z^* \cdot x^* = 0$;

(b) $p^{\mathrm{T}}Hp \geq 0$ *for all nonzero p satisfying* $Ap = 0$, *and* $p_i = 0$ *for every* $i \in \mathcal{A}(x^*)$.

Suitable sufficient conditions for optimality are given by (a)–(b) with (b) replaced by the condition that $p^{\mathrm{T}}Hp \geq \omega\|p\|^2$ for some $\omega > 0$ and all p such that $Ap = 0$, and $p_i = 0$ for every $i \in \mathcal{A}_+(x)$, where $\mathcal{A}_+(x)$ is the index set $\mathcal{A}_+(x) = \{i \in \mathcal{A}(x) : z_i > 0\}$.

Typically, software for general quadratic programming is designed to terminate at a dead point. Nevertheless, it is possible to define procedures that check for optimality at a dead point, but the chance of success in a reasonable amount of computation time depends on the dimension of the problem (see Forsgren, Gill and Murray [29]).

2.2.1 Primal active-set methods

We start by reviewing the properties of *primal-feasible active-set methods* for quadratic programming. An important feature of these methods is that once a feasible iterate is found, all subsequent iterates are feasible. The methods have two phases. In the first phase (called the *feasibility phase* or *phase one*), a feasible point is found by minimizing the sum of infeasibilities. In the second phase (the *optimality phase* or *phase two*), the quadratic objective function is minimized while feasibility is maintained. Each phase generates a sequence of inner iterates $\{x_j\}$ such that $x_j \geq 0$. The new iterate x_{j+1} is defined as $x_{j+1} = x_j + \alpha_j p_j$, where the *step length* α_j is a nonnegative scalar, and p_j is the *QP search direction*. For efficiency, it is beneficial if the computations in both phases are performed by the same underlying method. The two-phase nature of the algorithm is reflected by changing the function being minimized from a function that reflects the degree of infeasibility to the quadratic objective function. For this reason, it is helpful to consider methods for the optimality phase first.

At the jth step of the optimality phase, $\widehat{c}(x_j) = A(x_j - x_I) + b = 0$ and $x_j \geq 0$. The vector p_j is chosen to satisfy certain properties with respect to the objective and constraints. First, p_j must be a *direction of decrease* for φ at x_j, i.e., there must exist a positive $\bar{\alpha}$ such that

$$\varphi(x_j + \alpha p_j) < \varphi(x_j) \quad \text{for all } \alpha \in (0, \bar{\alpha}].$$

In addition, $x_j + p_j$ must be feasible with respect to the general constraints, and feasible with respect to the bounds associated with a certain "working set" of variables that serves as an estimate of the optimal active set of the QP. Using the terminology of linear programming, we call this working set of variables the *nonbasic set*, denoted by $\mathcal{N} = \{\nu_1, \nu_2, \ldots, \nu_{n_N}\}$. Similarly, we define the set

$\mathcal{B}$ of indices that are not in $\mathcal{N}$ as the *basic set*, with $\mathcal{B} = \{\beta_1, \beta_2, \ldots, \beta_{n_B}\}$, where $n_B = n - n_N$. Although $\mathcal{B}$ and $\mathcal{N}$ are strictly index sets, we will follow common practice and refer to variables x_{β_r} and x_{ν_s} as being "in $\mathcal{B}$" and "in $\mathcal{N}$" respectively.

With these definitions, we define the columns of A indexed by $\mathcal{N}$ and $\mathcal{B}$, the *nonbasic* and *basic* columns of A, as A_N and A_B, respectively. We refrain from referring to the nonbasic and basic sets as the "fixed" and "free" variables because some active-set methods allow some nonbasic variables to move (the simplex method for linear programming being one prominent example). An important attribute of the nonbasic set is that A_B *has rank* m, i.e., the rows of A_B are linearly independent. This implies that the cardinality of the nonbasic set must satisfy $0 \leq n_N \leq n - m$. It must be emphasized that our definition of $\mathcal{N}$ does not require a nonbasic variable to be active (i.e., at its lower bound). Also, whereas the active set is defined uniquely at each point, there are many choices for $\mathcal{N}$ (including the empty set). Given any n-vector y, the vector of *basic components of* y, denoted by y_B, is the n_B-vector whose jth component is component β_j of y. Similarly, y_N, the vector *nonbasic components of* y, is the n_N-vector whose jth component is component ν_j of y.

Given a basic-nonbasic partition of the variables, we introduce the definitions of stationarity and optimality with respect to a basic set.

Definition 2.2.1 (Subspace stationary point)**.** *Let $\mathcal{B}$ be a basic set defined at an $\widehat{x}$ such that $\widehat{c}(\widehat{x}) = 0$. Then $\widehat{x}$ is a* subspace stationary point with respect to $\mathcal{B}$ (or, equivalently, with respect to A_B) *if there exists a vector y such that $\widehat{g}_B(\widehat{x}) = A_B^{\mathrm{T}} y$. Equivalently, $\widehat{x}$ is a subspace stationary point with respect to $\mathcal{B}$ if the reduced gradient $Z_B^{\mathrm{T}} \widehat{g}_B(\widehat{x})$ is zero, where the columns of Z_B form a basis for the null-space of A_B.*

If $\widehat{x}$ is a subspace stationary point, φ is stationary on the subspace $\{x : A(x - \widehat{x}) = 0,\ x_N = \widehat{x}_N\}$. At a subspace stationary point, it holds that $g(\widehat{x}) = A^{\mathrm{T}} y + z$, where $z_i = 0$ for $i \in \mathcal{B}$—i.e.,

$z_B = 0$. Subspace stationary points may be classified based on the curvature of φ on the nonbasic set.

Definition 2.2.2 (Subspace minimizer)**.** *Let $\widehat{x}$ be a subspace stationary point with respect to $\mathcal{B}$. Let the columns of Z_B form a basis for the null-space of A_B. Then $\widehat{x}$ is a* subspace minimizer with respect to $\mathcal{B}$ *if the reduced Hessian $Z_B^{\mathrm{T}} H Z_B$ is positive definite.*

It should be noted here that sometimes subspace stationary points and minimizers are defined with respect to the working set rather than the basic set. In this case, Z_N is defined to be a matrix whose columns form a basis for the null space of G_N, where

$$G = \begin{pmatrix} A \\ I \end{pmatrix}, \qquad G = \begin{pmatrix} A_B & A_N \\ P_B & P_N \end{pmatrix}, \qquad \text{and} \qquad G_N = \begin{pmatrix} A_B & A_N \\ 0 & I_{n_N} \end{pmatrix},$$

where we assumed the basic columns precede the nonbasic ones, and P_B and P_N consist of columns of identity indexed by the corresponding set. It follows Z_N must have the form

$$Z_N = \begin{pmatrix} Z_B \\ 0 \end{pmatrix},$$

where Z_B is defined as in Definitions 2.2.1 and 2.2.2. In this alternative definition, a subspace stationary point satisfies $Z_N^{\mathrm{T}} \widehat{g}(\widehat{x}) = 0$ and a subspace minimizer gives $Z_N^{\mathrm{T}} H Z_N$ positive definite. Since $Z_N^{\mathrm{T}} \widehat{g}(\widehat{x}) = Z_B^{\mathrm{T}} \widehat{g}_B(\widehat{x})$ and $Z_N^{\mathrm{T}} H Z_N = Z_B^{\mathrm{T}} H Z_B$, it follows these definitions are equivalent.

If the nonbasic variables are active at $\widehat{x}$, then $\widehat{x}$ is called a *standard* subspace minimizer. At a standard subspace minimizer, if $z_N \geq 0$ then $\widehat{x}$ satisfies the necessary conditions for optimality. Otherwise, there exists an index $\nu_s \in \mathcal{N}$ such that $z_{\nu_s} < 0$. If some nonbasic variables are not active at $\widehat{x}$, then $\widehat{x}$ is called a *nonstandard* subspace minimizer.

It is convenient sometimes to be able to characterize the curvature of φ in a form that

does not require the matrix Z_B explicitly. The *inertia* of a symmetric matrix X, denoted by $\mathrm{In}(X)$, is the integer triple (i_+, i_-, i_0), where i_+, i_- and i_0 denote the number of positive, negative and zero eigenvalues of X. Gould [45] shows that if A_B has rank m and $A_B Z_B = 0$, then $Z_B^{\mathrm{T}} H_B Z_B$ is positive definite if and only if

$$\mathrm{In}(K_B) = (n_B, m, 0), \quad \text{where} \quad K_B = \begin{pmatrix} H_B & A_B^{\mathrm{T}} \\ A_B & 0 \end{pmatrix} \tag{2.38}$$

(see Forsgren [27] for a more general discussion, including the case where A_B does not have rank m). Many algorithms for solving symmetric equations that compute an explicit matrix factorization of K_B also provide the inertia as a by-product of the calculation, see, e.g., Bunch [5], and Bunch and Kaufman [6].

Below, we discuss two alternative formulations of an active-set method. Each generates a feasible sequence $\{x_j\}$ such that $x_{j+1} = x_j + \alpha_j p_j$ with $\varphi(x_{j+1}) \leq \varphi(x_j)$. Neither method requires the QP to be convex, i.e., H need not be positive semidefinite. The direction p_j is defined as the solution of an QP subproblem with equality constraints. Broadly speaking, the nonbasic components of p_j are *specified* and the basic components of p_j are adjusted to satisfy the general constraints $A(x_j + p_j) = Ax_I - b$. If p_B and p_N denote the basic and nonbasic components of p_j, then the nonbasic components are fixed by enforcing constraints of the form $p_N = d_N$, where d_N is a constant vector that characterizes the active-set method being used. The restrictions on p_j define constraints $Ap = 0$ and $p_N = d_N$. Any remaining degrees of freedom are used to define p_j as the direction that produces the largest reduction in φ. This gives the equality constrained QP subproblem

$$\underset{p}{\text{minimize}} \;\; \widehat{g}(x_j)^{\mathrm{T}} p + \tfrac{1}{2} p^{\mathrm{T}} H p \quad \text{subject to} \quad Ap = 0, \quad p_N = d_N.$$

In the following sections we define two methods based on alternative definitions of d_N. Both methods exploit the properties of a subspace minimizer (see Definition 2.2.2) in order to simplify the linear systems that must be solved.

Nonbinding-direction methods

We start with a method that defines a change in the basic-nonbasic partition at every iteration. In particular, one of three changes occurs: (i) a variable is moved from the basic set to the nonbasic set; (ii) a variable is moved from the nonbasic set to the basic set; or (ii) a variable in the basic set is swapped with a variable in the nonbasic set. These changes result in a column being added, deleted or swapped in the matrix A_B.

In order to simplify the notation, we drop the subscript j and consider the definition of a single iteration that starts at the primal-dual point (x, y) and defines a new iterate $(\bar{x}, \bar{y})$ such that $\bar{x} = x + \alpha p$ and $\bar{y} = y + \alpha q_y$. A crucial assumption about (x, y) is that it is a subspace minimizer with respect to the basis $\mathcal{B}$. It will be shown that this assumption guarantees that the next iterate $(\bar{x}, \bar{y})$ (and hence each subsequent iterate) is also a subspace minimizer.

Suppose that the reduced cost associated with the sth nonbasic variable is negative, i.e., $z_{\nu_s} < 0$. The direction p is defined so that all the nonbasic components are fixed except for the sth, which undergoes a unit change. This definition implies that a positive step along p increases x_{ν_s} but leaves all the other nonbasics unchanged. The required direction is defined by the equality constrained QP subproblem:

$$\underset{p}{\text{minimize}} \;\; \widehat{g}(x)^{\mathrm{T}} p + \tfrac{1}{2} p^{\mathrm{T}} H p \quad \text{subject to} \quad Ap = 0, \quad p_N = e_s, \tag{2.39}$$

and is said to be *nonbinding* with respect to the nonbasic variables. If the multipliers for the

constraints $Ap = 0$ are defined in terms of an increment q_y to y, then p_B and q_y satisfy the optimality conditions

$$\left(\begin{array}{cc|c} H_B & -A_B^{\mathrm{T}} & H_D \\ A_B & 0 & A_N \\ \hline 0 & 0 & I_N \end{array}\right) \begin{pmatrix} p_B \\ q_y \\ p_N \end{pmatrix} = - \begin{pmatrix} \widehat{g}_B(x) - A_B^{\mathrm{T}} y \\ 0 \\ - e_s \end{pmatrix},$$

where, as above, $\widehat{g}_B(x)$ are the basic components of $\widehat{g}(x)$, and H_B and H_D are the basic rows of the basic and nonbasic columns of H. If x is a subspace minimizer, then $\widehat{g}_B(x) - A_B^{\mathrm{T}} y = 0$, so that this system simplifies to

$$\left(\begin{array}{cc|c} H_B & -A_B^{\mathrm{T}} & H_D \\ A_B & 0 & A_N \\ \hline 0 & 0 & I_N \end{array}\right) \begin{pmatrix} p_B \\ q_y \\ p_N \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ e_s \end{pmatrix}, \tag{2.40}$$

yielding p_B and q_y as the solution of the smaller system

$$\begin{pmatrix} H_B & -A_B^{\mathrm{T}} \\ A_B & 0 \end{pmatrix} \begin{pmatrix} p_B \\ q_y \end{pmatrix} = - \begin{pmatrix} [\, h_{\nu_s} \,]_B \\ a_{\nu_s} \end{pmatrix}. \tag{2.41}$$

The increment q_N for multipliers z_N are computed from p_B, p_N and q_y as $q_N = (Hp - A^{\mathrm{T}} q_y)_{\mathcal{N}}$. Once p_B and q_y are known, a nonnegative step α is computed so that $x + \alpha p$ is feasible and $\varphi(x + \alpha p) \leq \varphi(x)$. The step that minimizes φ as a function of α is given by

$$\alpha_* = \begin{cases} -\widehat{g}(x)^{\mathrm{T}} p / p^{\mathrm{T}} H p & \text{if } p^{\mathrm{T}} H p > 0, \\ +\infty & \text{otherwise.} \end{cases} \tag{2.42}$$

The best feasible step is then $\alpha = \min\{\alpha_*, \alpha_M\}$, where α_M is the maximum feasible step:

$$\alpha_M = \min_{1 \le i \le n_B} \{\gamma_i\}, \quad \text{where } \gamma_i = \begin{cases} \dfrac{[\, x_B \,]_i}{-[\, p_B \,]_i} & \text{if } [\, p_B \,]_i < 0, \\ \\ +\infty & \text{otherwise.} \end{cases} \tag{2.43}$$

(As $p_N = e_s$ and the problem contains only lower bounds, $x + tp$ remains feasible with respect to the nonbasic variables for all $t \ge 0$.) If $\alpha = +\infty$ then φ decreases without limit along p and the problem is unbounded. Otherwise, the new iterate is $(\bar{x}, \bar{y}) = (x + \alpha p, y + \alpha q_y)$.

It is instructive to define the step α_* of (2.42) in terms of the identities

$$\widehat{g}(x)^{\mathrm{T}} p = z_{\nu_s} \quad \text{and} \quad p^{\mathrm{T}} H p = [\, q_N \,]_s, \tag{2.44}$$

which follow from the equations (2.40) that define p_B and p_N. Then, if α_* is bounded, we have $\alpha_* = -z_{\nu_s}/[\, q_N \,]_s$, or, equivalently,

$$z_{\nu_s} + \alpha_* [\, q_N \,]_s = 0.$$

Let $z(t)$ denote the vector of reduced costs at any point on the ray $(x + tp, y + tq_y)$, i.e., $z(t) = \widehat{g}(x + tp) - A^{\mathrm{T}}(y + tq_y)$. It follows from the definition of p and q_y of (2.40) that $z_B(t) = 0$ for all t, which implies that $x + tp$ is a subspace stationary point *for any step* t. (Moreover, $x + tp$ is a subspace minimizer because the KKT matrix K_B is independent of t.) This property, known as the *parallel subspace property of quadratic programming*, implies that $x + tp$ is the solution of an equality-constraint QP in which the bound on the sth nonbasic is *shifted* to pass through $x + tp$. The component $z_{\nu_s}(t)$ is the reduced cost associated with the shifted version of the bound $x_{\nu_s} \ge 0$. By definition, the sth nonbasic reduced cost is negative at x, i.e., $z_{\nu_s}(0) < 0$. Moreover, a simple calculation shows that $z_{\nu_s}(t)$ is an increasing linear function of t with $z_{\nu_s}(\alpha_*) = 0$ if α_* is bounded.

A zero reduced cost at $t = \alpha_*$ means that the shifted bound can be removed from the equality-constraint problem (2.39) (defined at $x = \bar{x}$) without changing its minimizer. Hence, if $\bar{x} = x + \alpha_* p$, the index ν_s is moved to the basic set, which adds column a_{ν_s} to $A_{\mathcal{B}}$ for the next iteration. The shifted variable has been removed from the nonbasic set, which implies that $(\bar{x}, \bar{y})$ is a *standard* subspace minimizer.

If we take a shorter step to the boundary of the feasible region, i.e., $\alpha_M < \alpha_*$, then at least one basic variable lies on its bound at $\bar{x} = x + \alpha p$, and one of these, x_{β_r} say, is made nonbasic. If $\bar{A}_{\mathcal{B}}$ denotes the matrix $A_{\mathcal{B}}$ with column r deleted, then $\bar{A}_{\mathcal{B}}$ is not guaranteed to have full row rank (for example, if x is a vertex, $A_{\mathcal{B}}$ is square and $\bar{A}_{\mathcal{B}}$ has more rows than columns). The linear independence of the rows of $\bar{A}_{\mathcal{B}}$ is characterized by the so-called "singularity vector" $u_{\mathcal{B}}$ given by the solution of the equations

$$\begin{pmatrix} H_{\mathcal{B}} & -A_{\mathcal{B}}^{\mathrm{T}} \\ A_{\mathcal{B}} & 0 \end{pmatrix} \begin{pmatrix} u_{\mathcal{B}} \\ v_y \end{pmatrix} = \begin{pmatrix} e_r \\ 0 \end{pmatrix}. \tag{2.45}$$

The matrix $\bar{A}_{\mathcal{B}}$ has full rank if and only if $u_{\mathcal{B}} \neq 0$. If $\bar{A}_{\mathcal{B}}$ is rank deficient, $\bar{x}$ is a subspace minimizer with respect to the basis defined by removing x_{ν_s}, i.e., x_{ν_s} is effectively replaced by x_{β_r} in the nonbasic set. In this case, it is necessary to update the dual variables again to reflect the change of basis (see Gill and Wong [43] for more details). The new multipliers are $\bar{y} + \sigma v_y$, where $\sigma = \widehat{g}(\bar{x})^{\mathrm{T}} p / [\, p_{\mathcal{B}} \,]_r$.

As defined above, this method requires the solution of two KKT systems at each step (i.e., equations (2.41) and (2.45)). However, if the solution of (2.45) is such that $u_{\mathcal{B}} \neq 0$, then the vectors $p_{\mathcal{B}}$ and q_y needed at $\bar{x}$ can be updated in $O(n)$ operations using the vectors $u_{\mathcal{B}}$ and v_y. Hence, it is unnecessary to solve (2.41) when a basic variable is removed from $\mathcal{B}$ following a restricted step.

Given an initial standard subspace minimizer x_0 and basic set $\mathcal{B}_0$, this procedure generates a sequence of primal-dual iterates $\{(x_j, y_j)\}$ and an associated sequence of basic sets $\{\mathcal{B}_j\}$. The

iterates occur in groups of consecutive iterates that start and end at a standard subspace minimizer. Each of the intermediate iterates is a nonstandard subspace minimizer at which the same nonbasic variable may not be on its bound. At each intermediate iterate, a variable moves from $\mathcal{B}$ to $\mathcal{N}$. At the first (standard) subspace minimizer of the group, a nonbasic variable with a negative reduced cost is targeted for inclusion in the basic set. In the subsequent set of iterations, this reduced cost is nondecreasing and the number of basic variables decreases. The group of consecutive iterates ends when the targeted reduced cost reaches zero, at which point the associated variable is made basic.

The method outlined above is based on a method first defined for constraints in all-inequality form by Fletcher [22], and extended to sparse QP by Gould [46]. Recent refinements, including the technique for reducing the number of KKT solves, are given by Gill and Wong [43]. Each of these methods is an example of an *inertia-controlling method*. The idea of an inertia-controlling method is to use the active-set strategy to limit the number of zero and negative eigenvalues in the KKT matrix K_B so that it has inertia $(n_B, m, 0)$ (for a survey, see Gill et al. [38]). At an arbitrary feasible point, a subspace minimizer can be defined by making sufficiently many variables temporarily nonbasic at their current value (see, e.g., Gill, Murray and Saunders [35] for more details).

Binding-direction methods

The next method employs a more conventional active-set strategy in which the nonbasic variables are always active. We start by assuming that the QP is *strictly convex*, i.e., that H is positive definite. Suppose that (x, y) is a feasible primal-dual pair such that $x_i = 0$ for $i \in \mathcal{N}$, where $\mathcal{N}$ is chosen so that A_B has rank m. As in a nonbinding direction method, the primal-dual

direction (p, q_y) is computed from an equality constrained QP subproblem. However, in this case the constraints of the subproblem not only force $Ap = 0$ but also require that *every* nonbasic variable remains unchanged for steps of the form $x + \alpha p$. This is done by fixing the nonbasic components of p at zero, giving the equality constraints $Ap = A_B p_B + A_N p_N = 0$ and $p_N = 0$. The resulting subproblem defines a direction that is *binding*, in the sense that it is "bound" or "attached" to the constraints in the nonbasic set. The QP subproblem that gives the best improvement in φ is then

$$\underset{p}{\text{minimize}} \;\; \widehat{g}(x)^{\mathrm{T}} p + \tfrac{1}{2} p^{\mathrm{T}} H p \quad \text{subject to} \quad A_B p_B = 0, \;\; p_N = 0. \tag{2.46}$$

The optimality conditions imply that p_B and q_y satisfy the KKT system

$$\begin{pmatrix} H_B & -A_B^{\mathrm{T}} \\ A_B & 0 \end{pmatrix} \begin{pmatrix} p_B \\ q_y \end{pmatrix} = - \begin{pmatrix} \widehat{g}_B(x) - A_B^{\mathrm{T}} y \\ 0 \end{pmatrix}. \tag{2.47}$$

These equations are nonsingular under our assumptions that H is positive definite and A_B has rank m. If (x, y) is a subspace stationary point, then $z_B = \widehat{g}_B(x) - A_B^{\mathrm{T}} y = 0$ and the solution (p_B, q_y) is zero. In this case, no improvement can be made in φ along directions in the null-space of A_B. If the components of $z = \widehat{g}(x) - A^{\mathrm{T}} y$ are nonnegative then x is optimal for (2.37). Otherwise, a nonbasic variable with a negative reduced cost is selected and moved to the basic set (with no change to x), thereby defining (2.47) with new A_B, H_B and (necessarily nonzero) right-hand side. Given a nonzero solution of (2.47), $x + p$ is either feasible or infeasible with respect to the bounds. If $x + p$ is infeasible, $\mathcal{N}$ cannot be the correct nonbasic set and feasibility is maintained by limiting the step by the maximum feasible step α_M as in (2.43). At the point $\bar{x} = x + \alpha p$, at least one of the basic variables must reach its bound and it is moved to the nonbasic set for the next iteration. Alternatively, if $x + p$ is feasible, $\bar{x} = x + p$ is a subspace minimizer and a nonoptimal nonbasic

variable is made basic as above.

The method described above defines groups of consecutive iterates that start with a variable being made basic. No more variables are made basic until either an unconstrained step is taken (i.e., $\alpha = 1$), or a sequence of constrained steps results in the definition of a subspace minimizer (e.g., at a vertex). At each constrained step, the number of basic variables decreases.

As H is positive definite in the strictly convex case, the KKT equations (2.47) remain nonsingular as long as A_B has rank m. One of the most important properties of a binding-direction method is that once an initial nonbasic set is chosen (with the implicit requirement that the associated A_B has rank m), then all subsequent A_B will have rank m (and hence the solution of the KKT system is always well defined). This result is of sufficient importance that we provide a brief proof.

If a variable becomes basic, a column is added to A_B and the rank does not change. It follows that the only possibility for A_B to lose rank is when a basic variable is made nonbasic. Assume that A_B has rank m and that the *first* basic variable is selected to become nonbasic, i.e., $r = 1$. If $\bar{A}_B$ denotes the matrix A_B without its first column, then $A_B = \begin{pmatrix} a_{\beta_r} & \bar{A}_B \end{pmatrix}$. If $\bar{A}_B$ does not have rank m then there must exist a nonzero m-vector $\bar{v}$ such that $\bar{A}_B^{\mathrm{T}}\bar{v} = 0$. If σ denotes the quantity $\sigma = -a_{\beta_r}^{\mathrm{T}}\bar{v}$, then the $(m+1)$-vector $v = (\bar{v}, \sigma)$ satisfies

$$\begin{pmatrix} a_{\beta_r}^{\mathrm{T}} & 1 \\ \bar{A}_B^{\mathrm{T}} & 0 \end{pmatrix} \begin{pmatrix} \bar{v} \\ \sigma \end{pmatrix} = 0, \quad \text{or equivalently,} \quad \begin{pmatrix} A_B^{\mathrm{T}} & e_r \end{pmatrix} v = 0.$$

The scalar σ must be nonzero or else $A_B^{\mathrm{T}}\bar{v} = 0$, which would contradict the assumption that A_B has rank m. Then

$$v^{\mathrm{T}} \begin{pmatrix} A_B \\ e_r^{\mathrm{T}} \end{pmatrix} p_B = v^{\mathrm{T}} \begin{pmatrix} 0 \\ [\, p_B \,]_r \end{pmatrix} = \sigma [\, p_B \,]_r = 0,$$

which implies that $[\, p_B \,]_r = 0$. This is a contradiction because the ratio test (2.43) will choose β_r as

the outgoing basic variable only if $[\,p_{B}\,]_r < 0$. It follows that $\bar{v} = 0$, and hence $\bar{A}_{B}$ must have rank m.

If H is not positive definite, the KKT matrix K_{B} associated with the equations (2.47) may have fewer than n_{B} positive eigenvalues (cf. (2.38)), i.e., the reduced Hessian $Z_{B}^{\mathrm{T}} H_{B} Z_{B}$ may be singular or indefinite. In this situation, the subproblem (2.46) is unbounded and the equations (2.47) cannot be used directly to define p. In this case we seek a direction p such that $p_{N} = 0$ and $A_{B} p_{B} = 0$, where

$$g_{B}^{\mathrm{T}} p_{B} < 0, \quad \text{and} \quad p_{B}^{\mathrm{T}} H_{B} p_{B} \leq 0. \tag{2.48}$$

The QP objective decreases without bound along such a direction, so either the largest feasible step α_{M} (2.43) is infinite, or a basic variable must become nonbasic at some finite α_{M} such that $\varphi(x + \alpha_{M} p) \leq \varphi(x)$. If $\alpha_{M} = +\infty$, the QP problem is unbounded and the algorithm is terminated.

Chapter 3

Stabilized and Primal-Dual SQP Methods

If J_F does not have full rank, the equations (2.30) are singular with no unique solution. In this case, one remedy is to use the constraint regularization techniques described in Section 2.1.2 to define a *stabilized SQP method* in which the QP subproblem (2.27) is replaced by

$$\begin{array}{ll} \underset{x,y}{\text{minimize}} & g_k^{\mathrm{T}}(x - x_k) + \frac{1}{2}(x - x_k)^{\mathrm{T}}\widehat{H}_k(x - x_k) + \frac{1}{2}\mu_k\|y\|^2 \\ \text{subject to} & c_k + J_k(x - x_k) + \mu_k(y - y_k) = 0, \quad x \geq 0, \end{array} \tag{3.1}$$

where $\{\mu_k\}$ is a positive sequence such that $\mu_k \to 0$ as $x_k \to x^*$ (see, e.g., Wright [70], Hager [49], Li and Qi [52], and Fernández and Solodov [20]). The QP (3.1) is often referred to as a stabilized subproblem because of its calming effect on multiplier estimates for degenerate problems (see, e.g., [49, 70]). Under certain assumptions, stabilized SQP methods exhibit fast local convergence.

However, there is no guarantee of convergence to a local solution for an arbitrary starting point. Under suitable assumptions, the method proposed in this chapter is guaranteed to be globally convergent and is equivalent to stabilized SQP in the limit.

3.1 A Regularized Primal-Dual Line-Search SQP Algorithm

This section defines a regularized SQP line-search method based on the primal-dual augmented Lagrangian function

$$M(x, y\,; y^E, \mu) = f(x) - c(x)^{\mathrm{T}} y^E + \frac{1}{2\mu}\|c(x)\|^2 + \frac{1}{2\mu}\|c(x) + \mu(y - y^E)\|^2, \tag{3.2}$$

where μ is the penalty parameter and y^E is an estimate of an optimal Lagrange multiplier vector y^*. (A trust-region-based method could also be given, but in this thesis we focus on line-search methods.) The function (3.2), proposed by Robinson [62], and Gill and Robinson [39], may be derived by applying the primal-dual penalty function of Forsgren and Gill [28] to a problem in which the constraints are shifted by a constant vector (see Powell [61]). With the notation $c = c(x)$, $g = \nabla f(x)$, and $J = J(x)$, the gradient of $M(x, y\,; y^E, \mu)$ may be written as

$$\nabla M(x, y\,; y^E, \mu) = \begin{pmatrix} g - J^{\mathrm{T}}\big(2(y^E - \frac{1}{\mu}c) - y\big) \\ c + \mu(y - y^E \end{pmatrix} \tag{3.3a}$$

$$= \begin{pmatrix} g - J^{\mathrm{T}}\big(\pi + (\pi - y)\big) \\ \mu(y - \pi) \end{pmatrix}, \tag{3.3b}$$

where $\pi = \pi(x\,; y^E, \mu)$ denotes the vector-valued function

$$\pi(x\,; y^E, \mu) = y^E - \frac{1}{\mu}c(x). \tag{3.4}$$

Similarly, the Hessian of $M(x, y\,; y^E, \mu)$ may be written as

$$\nabla^2 M(x, y\,; y^E, \mu) = \begin{pmatrix} H\big(x, \pi + (\pi - y)\big) + \frac{2}{\mu} J^{\mathrm{T}} J & J^{\mathrm{T}} \\ J & \mu I \end{pmatrix}. \tag{3.5}$$

Our approach is motivated by the following theorem, which shows that under certain assumptions, minimizers of problem (NP) are also minimizers of the bound constrained problem

$$\underset{x,y}{\text{minimize}} \;\; M(x, y\,; y^E, \mu) \quad \text{subject to} \quad x \geq 0. \tag{3.6}$$

Theorem 3.1.1 (Robinson [62, Theorem 4.6.1]). *If (x^*, y^*) satisfies second-order sufficient conditions for a solution of problem* (NP)*, then there exists a positive $\bar{\mu}$ such that, for all $0 < \mu < \bar{\mu}$ and $y^E = y^*$, the point (x^*, y^*) is a minimizer of problem* (3.6).

The reader is referred to Robinson [62] and Gill and Robinson [39] for additional details. In this context, Theorem 3.1.1 is used as motivation for the algorithm described below.

3.2 Definition of the Primal-Dual Search Direction

Given the kth iterate $v_k = (x_k, y_k)$, a Lagrange multiplier estimate y_k^E, and a positive regularization parameter μ_k^R, a symmetric matrix $\widehat{H}(x_k, y_k) \approx H(x_k, y_k)$ is defined such that $\widehat{H}(x_k, y_k) + (1/\mu_k^R) J(x_k)^{\mathrm{T}} J(x_k)$ is positive definite. One may choose $\widehat{H}(x_k, y_k)$ itself to be positive definite, but we explore a more sophisticated strategy in Section 3.3.1 that allows for an *indefinite* matrix $\widehat{H}(x_k, y_k)$ that more faithfully approximates $H(x_k, y_k)$. With this assumption on the matrix $\widehat{H}$, part (i) of Lemma 3.2.1 given below may be applied with the quantities $H = \widehat{H}(x_k, y_k)$,

$J = J(x_k)$, and $\mu = \mu_k^R$, to infer that the matrix

$$H^M(x_k, y_k\,; \mu_k^R) = \begin{pmatrix} \widehat{H}(x_k, y_k) + \dfrac{2}{\mu_k^R} J(x_k)^{\mathrm{T}} J(x_k) & J(x_k)^{\mathrm{T}} \\ J(x_k) & \mu_k^R I \end{pmatrix} \tag{3.7}$$

is a positive-definite approximation to the Hessian of M. Given an appropriate matrix $H^M(v_k\,; \mu_k^R) \equiv H^M(x_k, y_k\,; \mu_k^R)$, the primal-dual search direction is given by

$$d_k = \widehat{v}_k - v_k, \tag{3.8}$$

where $\widehat{v}_k = (\widehat{x}_k, \widehat{y}_k)$ is a solution of the strictly convex bound-constrained QP subproblem:

$$\begin{aligned} &\underset{v}{\text{minimize}} \quad \varphi(v) = \nabla M(v_k\,; y_k^E, \mu_k^R)^{\mathrm{T}}(v - v_k) + \tfrac{1}{2}(v - v_k)^{\mathrm{T}} H^M(v_k\,; \mu_k^R)(v - v_k) \\ &\text{subject to} \quad v_i \geq 0, \quad i = 1, 2, \ldots, n. \end{aligned} \tag{3.9}$$

The following lemma provides the connections between the inertias of various matrices (part (i) may be used to conclude that the subproblem (3.9) is strictly convex).

Lemma 3.2.1. *Let μ be a positive scalar. Let H and J be matrices such that H is symmetric $n \times n$ and J is $m \times n$. If we define*

$$H^M = \begin{pmatrix} H + \frac{2}{\mu} J^{\mathrm{T}} J & J^{\mathrm{T}} \\ J & \mu I_m \end{pmatrix} \quad \textit{and} \quad K = \begin{pmatrix} H & J^{\mathrm{T}} \\ J & -\mu I_m \end{pmatrix},$$

then the following properties hold.

(i) *The matrix $H + \frac{1}{\mu} J^{\mathrm{T}} J$ is positive definite if and only if* $\mathrm{In}(H^M) = (n + m, 0, 0)$.

(ii) *The matrix $H + \frac{1}{\mu} J^{\mathrm{T}} J$ is positive definite if and only if* $\mathrm{In}(K) = (n, m, 0)$.

Proof. It may be verified by direct multiplication that

$$L^{\mathrm{T}} H^{M} L = \begin{pmatrix} H + \frac{1}{\mu} J^{\mathrm{T}} J & 0 \\ 0 & \mu I_m \end{pmatrix}, \quad \text{where} \quad L = \begin{pmatrix} I_n & 0 \\ -\frac{1}{\mu} J & I_m \end{pmatrix}.$$

The matrix L is nonsingular, and Sylvester's law of inertia gives

$$\mathrm{In}(H^{M}) = \mathrm{In}(L^{\mathrm{T}} H^{M} L) = \mathrm{In}\left(H + \frac{1}{\mu} J^{\mathrm{T}} J\right) + (m, 0, 0),$$

which implies the result of part (i).

To prove part (ii), consider the identity

$$S^{\mathrm{T}} K S = \begin{pmatrix} H + \frac{1}{\mu} J^{\mathrm{T}} J & 0 \\ 0 & -\mu I_m \end{pmatrix}, \quad \text{where} \quad S = \begin{pmatrix} I_n & 0 \\ \frac{1}{\mu} J & I_m \end{pmatrix}.$$

It now follows from the nonsingularity of S and Sylvester's law of inertia that

$$\mathrm{In}(K) = \mathrm{In}(S^{\mathrm{T}} K S) = \mathrm{In}\left(H + \frac{1}{\mu} J^{\mathrm{T}} J\right) + (0, m, 0),$$

from which part (ii) follows directly. □

The first-order optimality conditions for any primal-dual QP solution $\widehat{v}_k = (\widehat{x}_k, \widehat{y}_k)$ of the bound-constrained QP (3.9) may be written in matrix form

$$\begin{pmatrix} \widehat{H}_{F} & J_{F}^{\mathrm{T}} \\ J_{F} & -\mu_k^{R} I \end{pmatrix} \begin{pmatrix} [\,\widehat{x}_k - x_k\,]_{F} \\ -(\widehat{y}_k - y_k) \end{pmatrix} = - \begin{pmatrix} [\,g_k + \widehat{H}_k \eta_k - J_k^{\mathrm{T}} y_k\,]_{F} \\ c_k + J_k \eta_k + \mu_k^{R}(y_k - y_k^{E}) \end{pmatrix}, \tag{3.10}$$

where c_k, g_k and J_k denote the quantities $c(x)$, $\nabla f(x)$ and $J(x)$ evaluated at x_k, and the quantities with suffix "F" are defined in terms of the index set $\mathcal{F}(\widehat{x}_k)$, i.e., $\widehat{H}_{F}$ is the matrix of free rows and

columns of $\widehat{H}_k = \widehat{H}(x_k, y_k)$, and $J_{\scriptscriptstyle F}$ is the matrix of free columns of J_k at $\widehat{x}_k$. The vector η_k is nonpositive with components

$$[\,\eta_k\,]_i = \begin{cases} -[\,x_k\,]_i & \text{if } i \in \mathcal{A}(\widehat{x}_k); \\ \\ 0 & \text{if } i \in \mathcal{F}(\widehat{x}_k). \end{cases}$$

As $\widehat{H}_k + (1/\mu_k^{\scriptscriptstyle R})J_k^{\scriptscriptstyle \mathrm{T}} J_k$ is positive definite by construction, it follows immediately that the principal submatrix $\widehat{H}_{\scriptscriptstyle F} + (1/\mu_k^{\scriptscriptstyle R})J_{\scriptscriptstyle F}^{\scriptscriptstyle \mathrm{T}} J_{\scriptscriptstyle F}$ is also positive definite. We may then apply part (ii) of Lemma 3.2.1 with values $H = \widehat{H}_{\scriptscriptstyle F}$, $J = J_{\scriptscriptstyle F}$, and $\mu = \mu_k^{\scriptscriptstyle R}$, to infer that the matrix associated with the equations (3.10) is nonsingular. It follows that if $\mathcal{A}(\widehat{x}_k) = \mathcal{A}(x_k)$, then η_k is zero and $(\widehat{x}_k, \widehat{y}_k)$ satisfies the perturbed Newton equations

$$\begin{pmatrix} \widehat{H}_{\scriptscriptstyle F} & J_{\scriptscriptstyle F}^{\scriptscriptstyle \mathrm{T}} \\ J_{\scriptscriptstyle F} & -\mu_k^{\scriptscriptstyle R} I \end{pmatrix} \begin{pmatrix} [\,\widehat{x}_k - x_k\,]_{\scriptscriptstyle F} \\ -(\widehat{y}_k - y_k) \end{pmatrix} = - \begin{pmatrix} [\,g_k - J_k^{\scriptscriptstyle \mathrm{T}} y_k\,]_{\scriptscriptstyle F} \\ c_k + \mu_k^{\scriptscriptstyle R}(y_k - y_k^{\scriptscriptstyle E}) \end{pmatrix}. \tag{3.11}$$

A key property is that if $\mu_k^{\scriptscriptstyle R} = 0$ and $J_{\scriptscriptstyle F}$ has full rank, then this equation is identical to the equation for the conventional SQP step given by (2.29). This provides the motivation to use a small penalty parameter $\mu_k^{\scriptscriptstyle R}$ for the step computation and a different larger penalty parameter μ_k for the merit function. In this context, $\mu_k^{\scriptscriptstyle R}$ plays the role of a *regularization* parameter rather than a *penalty* parameter, thereby providing an $O(\mu_k^{\scriptscriptstyle R})$ estimate of the conventional SQP direction. This approach is nonstandard because a small "penalty parameter" $\mu_k^{\scriptscriptstyle R}$ is used by design, whereas conventional augmented Lagrangian-based methods attempt to keep μ as large as possible [11, 35].

The discussion above has established the relationship between the computation of the primal-dual bound-constrained step and the solution of a regularized QP. The next result formalizes the connection between the primal-dual step and the step associated with a *stabilized* SQP method.

Result 3.2.1. *Let μ_k^R denote a fixed scalar such that $\mu_k^R > 0$. Let $v_k = (x_k, y_k)$, $g_k = g(x_k)$, $c_k = c(x_k)$, and $J_k = J(x_k)$. Given a matrix $\widehat{H}_k = \widehat{H}(x_k, y_k)$ such that $\widehat{H}_k + (1/\mu_k^R) J_k^{\mathrm{T}} J_k$ is positive definite, consider the subproblem*

$$\begin{aligned} \underset{x,y}{\text{minimize}} \quad & g_k^{\mathrm{T}}(x - x_k) + \tfrac{1}{2}(x - x_k)^{\mathrm{T}} \widehat{H}_k (x - x_k) + \tfrac{1}{2}\mu_k^R \|y\|^2 \\ \text{subject to} \quad & c_k + J_k(x - x_k) + \mu_k^R (y - y_k^E) = 0, \;\; x \geq 0, \end{aligned} \tag{3.12}$$

which is the stabilized SQP subproblem (3.1) *defined with $\mu_k = \mu_k^R$ and $y_k = y_k^E$. The following results hold.*

(i) *The stabilized QP* (3.12) *has a unique bounded primal-dual solution $\widehat{v}_k = (\widehat{x}_k, \widehat{y}_k)$.*

(ii) *The unique solution $\widehat{v}_k = (\widehat{x}_k, \widehat{y}_k)$ of the stabilized QP* (3.12) *is also the* unique *solution of* (3.9).

Proof. To simplify notation, the regularization parameter μ_k^R will be denoted by μ. For part (i), given the particular feasible point $v_0 = (x_k, \pi_k)$ with $\pi_k = y_k^E - c_k/\mu$, any feasible point $v = (x, y)$ may be written as

$$v = v_0 + Nw \;\text{ for some vector } w \in \mathbb{R}^n, \text{ where } \; N = \begin{pmatrix} \mu I \\ -J_k \end{pmatrix}.$$

The matrix N is $(n + m) \times n$ with rank n, and its columns form a basis for the null-space of the constraint matrix $\begin{pmatrix} J_k & \mu I \end{pmatrix}$. Applying this equivalent form of v to (3.12) gives the equivalent problem

$$\underset{w \in \mathbb{R}^n}{\text{minimize}} \;\; \frac{\mu}{2} w^{\mathrm{T}} \Big(\widehat{H}_k + \frac{1}{\mu} J_k^{\mathrm{T}} J_k \Big) w + w^{\mathrm{T}} \big(g_k - J_k^{\mathrm{T}} \pi_k \big) \;\; \text{subject to} \;\; \mu w \geq -x_k.$$

The matrix $\widehat{H}_k + (1/\mu)J_k^{\mathrm{T}}J_k$ is positive definite by assumption, and it follows that the stabilized QP (3.12) is equivalent to a convex program with a strictly convex objective. The existence of a unique bounded solution follows directly.

For part (ii), it is sufficient to show that the optimality conditions for the problems (3.12) and (3.9) are equivalent. The first-order conditions for (x, y) to be a solution of the stabilized QP (3.12) are:

$$\begin{aligned} c_k + J_k(x - x_k) + \mu(y - y_k^E) = 0, \qquad & \mu y = \mu\pi, \\ g_k + \widehat{H}_k(x - x_k) - J_k^{\mathrm{T}}\pi - z = 0, \qquad & z \geq 0, \\ z \cdot x = 0, \qquad & x \geq 0, \end{aligned}$$

where π and z denote the dual variables for the equality and inequality constraints of problem (3.12), respectively. Eliminating π using the equation $\pi = y$ gives

$$c_k + J_k(x - x_k) + \mu(y - y_k^E) = 0, \tag{3.13a}$$

$$g_k + \widehat{H}_k(x - x_k) - J_k^{\mathrm{T}}y - z = 0, \qquad z \geq 0, \tag{3.13b}$$

$$z \cdot x = 0, \qquad x \geq 0. \tag{3.13c}$$

The optimality conditions for the bound-constrained QP (3.9) are

$$\nabla M(v_k; y_k^E, \mu) + H^M(v_k\,; \mu)(v - v_k) = \begin{pmatrix} z \\ 0 \end{pmatrix}, \qquad z \geq 0, \tag{3.14a}$$

$$z \cdot x = 0, \qquad x \geq 0. \tag{3.14b}$$

Premultiplying the equality of (3.14a) by the nonsingular matrix T defined by

$$T = \begin{pmatrix} I_n & -\frac{2}{\mu} J_k^{\mathrm{T}} \\ 0 & I_m \end{pmatrix},$$

and using the definitions (3.3) and (3.4) yields the equivalent conditions

$$g_k + \widehat{H}_k(x - x_k) - J_k^{\mathrm{T}} y - z = 0 \quad \text{and} \quad c_k + J_k(x - x_k) + \mu(y - y_k^E) = 0,$$

which are identical to the relevant equalities in (3.13). Thus, the solutions of (3.12) and (3.9) are identical.

The uniqueness of the solution $v = (x, y)$ follows from part (i) of Lemma 3.2.1, which implies that the objective Hessian of the bound constrained QP (3.9) is positive definite, thereby ensuring a strictly convex QP. □

3.2.1 Definition of the new iterate

Once the search direction $d_k = \widehat{v}_k - v_k$ has been determined, a "flexible" backtracking line search is performed on the primal-dual augmented Lagrangian. A conventional backtracking line search defines $v_{k+1} = v_k + \alpha_k d_k$, where $\alpha_k = 2^{-j}$ and j is the smallest nonnegative integer such that

$$M(v_k + \alpha_k d_k \,; y_k^E, \mu_k) \leq M(v_k \,; y_k^E, \mu_k) + \alpha_k \eta_S d_k^{\mathrm{T}} \nabla M(v_k \,; y_k^E, \mu_k)$$

for a given $\eta_S \in (0, \frac{1}{2})$. However, this approach would suffer from the Maratos effect [55] simply because the penalty parameter μ_k and the regularization parameter μ_k^R used to compute the trial step have different values in general. This difficulty is avoided by using an augmented Lagrangian

version of the "flexible penalty function" proposed by Curtis and Nocedal [16]. This method defines a step length of the form $\alpha_k = 2^{-j}$, where j is the smallest nonnegative integer satisfying

$$M(v_k + \alpha_k d_k \,; y_k^E, \mu_k^F) \leq M(v_k \,; y_k^E, \mu_k^F) + \alpha_k \eta_S \delta_k \tag{3.15}$$

for some value $\mu_k^F \in [\mu_k^R, \mu_k]$, and δ_k such that

$$\delta_k = \max\left(d_k^{\mathrm{T}} \nabla M(v_k \,; y_k^E, \mu_k^R), -\eta_D \|d_k\|^2\right) \leq 0, \tag{3.16}$$

with η_D a small positive constant. The use of the second term in the definition of δ_k increases the chance that a step is accepted during the early iterations when $|d_k^{\mathrm{T}} \nabla M(v_k \,; y_k^E, \mu_k^R)|$ is large. Once an appropriate value for α_k is found, the new primal-dual solution estimate is given by

$$x_{k+1} = x_k + \alpha_k(\widehat{x}_k - x_k) \quad \text{and} \quad y_{k+1} = y_k + \alpha_k(\widehat{y}_k - y_k).$$

In a practical algorithm, the step is reduced until the Armijo condition (3.15) is satisfied for one of the values $\mu_k^F = \mu_k$ or $\mu_k^F = \mu_k^R$ (where the condition for $\mu_k^F = \mu_k$ is tried first). The following simple argument shows that the acceptance criterion (3.15) is well-defined, i.e., the sequence $\{2^{-j}\}$ must terminate with an acceptable α_k. As $v = v_k$ is feasible for the strictly convex problem (3.9), the search direction $d_k = (\widehat{x}_k - x_k, \widehat{y}_k - y_k)$ is a feasible descent direction for $M(v \,; y_k^E, \mu_k^R)$ at $v_k = (x_k, y_k)$. If follows from standard theory that the weakened Armijo condition (3.15) will be satisfied for $\mu_k^F = \mu_k^R$ and all $\alpha_k > 0$ sufficiently small.

3.2.2 Updating the multiplier estimate

The QP equivalence established in Result 3.2.1, together with the definition of the stabilized SQP subproblem (3.1) imply that setting $y_k^E = y_k$ in the definition of the subproblem (3.12) (or, equivalently, in the bound-constrained QP (3.9)) makes the proposed trial step identical to that of the stabilized SQP method. This motivates an update strategy that allows the definition $y_k^E = y_k$ as often as possible. The idea is to define $y_{k+1}^E = y_{k+1}$ for the next subproblem if the line search gives an (x_{k+1}, y_{k+1}) that improves at least one of two merit functions that measure the accuracy of (x_{k+1}, y_{k+1}) as an estimate of (x^*, y^*). Let β denote a small positive parameter and consider the merit functions

$$\phi_V(x,y) = \eta(x) + \beta\omega(x,y), \quad \text{and} \quad \phi_O(x,y) = \beta\eta(x) + \omega(x,y), \tag{3.17}$$

where $\eta(x)$ and $\omega(x,y)$ are the feasibility violation and optimality measures

$$\eta(x) = \|c(x)\| \quad \text{and} \quad \omega(x,y) = \left\|\min\left(x, g(x) - J(x)^{\mathrm{T}}y\right)\right\|. \tag{3.18}$$

These functions provide two alternative weighted measures of the accuracy of (x,y) as an approximate solution of problem (NP) rather than as an approximate minimizer of M. Both measures are bounded below by zero, and are equal to zero if v is a first-order solution to problem (NP).

Given these definitions, the estimate y_k^E is updated when any iterate $v_k = (x_k, y_k)$ satisfies either $\phi_V(v_k) \leq \frac{1}{2}\phi_V^{\max}$ or $\phi_O(v_k) \leq \frac{1}{2}\phi_O^{\max}$, where $\phi_V^{\max}$ and $\phi_O^{\max}$ are bounds that are updated throughout the solution process. To ensure global convergence, an update to y_k^E forces a decrease in either $\phi_V^{\max}$ or $\phi_O^{\max}$. The idea is to choose the parameter β of (3.18) to be relatively small, say $\beta = 10^{-5}$. This allows frequent updates to y_k^E.

Finally, y_k^E is also updated if an approximate first-order solution to problem (3.6) has been found for the values $y^E = y_k^E$ and $\mu = \mu_k^R$. The test for optimality is

$$\|\nabla_y M(v_{k+1}\,; y_k^E, \mu_k^R)\| \leq \tau_k \;\text{ and }\; \big\|\min\big(x_{k+1}, \nabla_x M(v_{k+1}\,; y_k^E, \mu_k^R)\big)\big\| \leq \tau_k \tag{3.19}$$

for some small tolerance $\tau_k > 0$. This condition is rarely triggered in practice, but the test is needed to ensure global convergence. Nonetheless, if condition (3.19) is satisfied, y_k^E is updated with the safeguarded estimate

$$y_{k+1}^E = \max\big(-y_{\max}e,\, \min(y_{k+1}, y_{\max}e)\big), \tag{3.20}$$

for some large positive scalar constant $y_{\max}$.

3.2.3 Updating the penalty parameters

The following definition is designed to decrease μ_k^R only in the neighborhood of an optimal point (assuming that the problem is not locally infeasible):

$$\mu_{k+1}^R = \begin{cases} \min\big(\tfrac{1}{2}\mu_k^R, \|r_{\text{opt}}(v_{k+1})\|^{3/2}\big), & \text{if (3.19) is satisfied;} \\ \min\big(\;\mu_k^R, \|r_{\text{opt}}(v_{k+1})\|^{3/2}\big), & \text{otherwise,} \end{cases} \tag{3.21}$$

where r_{opt} is the vector-valued function

$$r_{\text{opt}}(v) = \begin{pmatrix} c(x) \\ \min\big(x,\, g(x) - J(x)^{\mathrm{T}} y\big) \end{pmatrix}. \tag{3.22}$$

The update to μ_k is motivated by a different goal. Namely, μ_k should be decreased only when the trial step indicates that the merit function defined with penalty parameter μ_k *increases*. This

motivates the definition

$$\mu_{k+1} = \begin{cases} \mu_k, & \text{if } M(v_{k+1}\,; y_k^E, \mu_k) \le M(v_k\,; y_k^E, \mu_k) + \widehat{\alpha}_k \eta_S \delta_k; \\ \max\left(\frac{1}{2}\mu_k, \mu_{k+1}^R\right), & \text{otherwise,} \end{cases} \tag{3.23}$$

where $\widehat{\alpha}_k = \min(\alpha_{\min}, \alpha_k)$ for some positive $\alpha_{\min}$, and δ_k is defined by (3.16). The use of the scalar $\alpha_{\min}$ increases the likelihood that μ_k will not be decreased.

3.3 Solution of the Bound-Constrained Subproblem

In this section we consider methods for the solution of a bound-constrained QP (3.9). The remainder of this section focuses on the solution of a single QP subproblem, and the notation is simplified so that $v_k = (x_k, y_k) = (x, y)$, $J = J_k$, $H = H(x_k, y_k)$, $\widehat{H} = \widehat{H}(x_k, y_k)$, $H^M = H^M(x_k, y_k\,; \mu_k^R)$, and $\mu = \mu_k^R$. Similarly, J_F and J_A denote the columns of J associated with the index sets $\mathcal{F}(x)$ and $\mathcal{A}(x)$ of free and fixed variables at x. Throughout this section, if S is a symmetric matrix, then S_F and S_A denote the symmetric matrices with elements s_{ij} for i, j in $\mathcal{F}$ and $\mathcal{A}$ respectively. Given these definitions, the problem to be solved is

$$\min_v \ \nabla M^{\mathrm{T}}(v - \bar{v}) + \tfrac{1}{2}(v - \bar{v})^{\mathrm{T}} H^M (v - \bar{v}) \quad \text{subject to} \quad v_i \ge 0,\ i = 1, 2, \ldots, n, \tag{3.24}$$

where v is the vector of $n + m$ primal-dual variables $v = (x, y)$, $\bar{v}$ is the constant vector $\bar{v} = (\bar{x}, \bar{y})$, and

$$\nabla M = \begin{pmatrix} g - J^{\mathrm{T}}\big(\pi + (\pi - \bar{y})\big) \\ c + \mu(\bar{y} - y^E). \end{pmatrix}, \quad H^M = \begin{pmatrix} \widehat{H} + \frac{2}{\mu} J^{\mathrm{T}} J & J^{\mathrm{T}} \\ J & \mu I \end{pmatrix},$$

where $\widehat{H}$ is a symmetric approximation of the Hessian of the Lagrangian.

First, we assume that the matrix $\widehat{H}$ is such that $\widehat{H} + \frac{1}{\mu}J^{\mathrm{T}}J$ is positive definite. It follows from Lemma 3.2.1 that the matrix H^{M} is positive definite and the bound constrained problem (3.24) is a strictly convex QP that may be solved using a conventional active-set method.

At the jth iterate $v_j = (x_j, y_j)$, the index sets of active and free variables are given by $\widehat{\mathcal{A}}(v_j)$ and $\widehat{\mathcal{F}}(v_j)$, where

$$\widehat{\mathcal{A}}(v) = \mathcal{A}(x) = \{\, i : x_i = 0 \,\} \quad \text{and} \quad \widehat{\mathcal{F}}(v) = \{1, 2, \ldots, n+m\} \setminus \widehat{\mathcal{A}}(v).$$

(As the dual variables are not subject to bounds, the vector of free components of any $v = (x, y)$ has the form $v_{\hat{F}} = (x_{F}, y)$ with x_{F} defined in terms of $\mathcal{F}$.) Given $v_j = (x_j, y_j)$, the next QP iterate is defined as $v_{j+1} = v_j + \alpha_j d_j$, where the free components of the vector $d_j = (p_j, q_j)$ satisfy the equations

$$H^{M}_{\hat{F}} d_{\hat{F}} = -[\, \nabla M + H^{M}(v_j - \bar{v}) \,]_{\hat{F}}, \tag{3.25}$$

with $d_{\hat{F}} = (p_{F}, q_j)$. The equations (3.25) appear to be ill-conditioned for small μ because of the $O(1/\mu)$ term in the $(1,1)$ block of the matrix H^{M}. However, this ill-conditioning is superficial. The next result shows that d_{F} may be determined by solving an equivalent nonsingular primal-dual system with conditioning dependent on that of the original problem.

Theorem 3.3.1. *Consider the application of the active-set method to the bound constrained QP* (3.24). *The free components of the QP search direction (p_j, q_j) satisfy the nonsingular primal-dual system*

$$\begin{pmatrix} \widehat{H}_{F} & -J_{F}^{\mathrm{T}} \\ J_{F} & \mu I \end{pmatrix} \begin{pmatrix} p_{F} \\ q_j \end{pmatrix} = - \begin{pmatrix} [\, g + \widehat{H}(x_j - \bar{x}) - J^{\mathrm{T}} y_j \,]_{F} \\ c + \mu(y_j - y^{E}) + J(x_j - \bar{x}) \end{pmatrix}. \tag{3.26}$$

Proof. It suffices to show that the linear systems (3.25) and (3.26) are equivalent. Consider the

matrix

$$U = \begin{pmatrix} I & -\frac{2}{\mu} J_{\scriptscriptstyle F}^{\mathrm{T}} \\ 0 & I_m \end{pmatrix},$$

where the identity matrix I has dimension $n_{\scriptscriptstyle F}$, the column dimension of $J_{\scriptscriptstyle F}$. The matrix U is nonsingular with $n_{\scriptscriptstyle F} + m$ rows and columns. It follows that the equations

$$U H_{\hat{F}}^{M} d_{\hat{F}} = -U[\, \nabla M + H^{M}(v_j - \bar{v})\,]_{\hat{F}}$$

have the same solution as those of (3.25). The primal-dual equations (3.26) follow by direct multiplication. The nonsingularity of the equations (3.26) follows from the nonsingularity of U, and the fact that H^{M} is positive definite (as are all symmetric submatrices formed from its rows and columns). □

3.3.1 Convexification of the bound-constrained subproblem

An important aspect of the proposed method is the definition of $\widehat{H}(x_k, y_k)$, which is used to ensure that the bound constrained QP subproblem (3.9) is convex. A conventional QP subproblem defined with the Hessian of the Lagrangian is not convex, in general. To avoid solving an indefinite subproblem, most existing methods are based on solving a convex QP based on a positive-semidefinite approximation $\widehat{H}(x_k, y_k)$ of the Hessian $H(x_k, y_k)$. This convex subproblem is used to either define the search direction directly, or identify the constraints for an equality-constrained QP subproblem that uses the exact Hessian (see, e.g., [35, 8, 47]).

In Chapter 5 we describe a different approach and define a *convexified* QP subproblem in terms of the exact Hessian of the Lagrangian. The convex problem is defined in such a way that if the inner iterations do not alter the active set, then the computed direction is equivalent

to a second-derivative stabilized SQP direction, provided that $y_k^E = y_k$. The method is based on the implicit computation of a symmetric matrix $\widehat{H}(x_k, y_k)$ (not necessarily positive definite) as a modification of $H(x_k, y_k)$ that gives a bounded convex primal-dual subproblem (3.9).

Convexification is a process for defining a local convex approximation of a nonconvex problem. This approximation may be defined on the full space of variables or just on some subset. Many model-based optimization methods use some form of convexification. For example, line-search methods for unconstrained and linearly-constrained optimization define a convex local quadratic model in which the Hessian $H(x_k, y_k)$ is replaced by a positive-definite matrix $H(x_k, y_k) + E_k$ (see, e.g., Greenstadt [48], Gill and Murray [34], Schnabel and Eskow [66], and Forsgren and Murray [30]). All of these methods are based on convexifying an unconstrained or equality-constrained local model. Here we consider a method that convexifies the inequality-constrained subproblem directly. The method extends some approaches proposed by Gill and Robinson [40, Section 4] and Kungurtsev [51].

In the context of SQP methods, the purpose of the convexification is to find a matrix ΔH_k such that

$$p_k^{\mathrm{T}}\big(H(x_k, y_k) + \Delta H_k\big)p_k \geq \lambda_{\min} p_k^{\mathrm{T}} p_k,$$

for a given primal-dual pair (x_k, y_k), where $\lambda_{\min}$ is a fixed positive scalar that defines a minimum acceptable value of the curvature of the Lagrangian.

The proposed convexification scheme can take three forms: *pre-convexification*, *concurrent convexification*, and *post-convexification*. This process gives a matrix $\widehat{H}$ of the form

$$\widehat{H} = H + \Delta + \Sigma + \Gamma, \tag{3.27}$$

where Δ is a symmetric positive semidefinite matrix, and Σ is a positive-semidefinite diagonal. It must be emphasized that $\widehat{H}$ itself is not necessarily positive definite. We emphasize that not all of these modifications are necessarily needed at a given iteration. These convexification schemes will be discussed in detail in Chapter 5.

Algorithm 1 Bound-constrained minimization.

Input $v = (x,\ y)$ such that $x \geq 0$;
Compute $\mathcal{A} = \{\, i : x_i = 0 \,\}$ and $\mathcal{F} = \{\, i : x_i > 0 \,\}$; Set $\widehat{\mathcal{F}} = \mathcal{F} \cup \{\, 1, 2, \ldots, m \,\}$;
repeat
    repeat
        Solve $H^{M}_{\widehat{F}} d_{\widehat{F}} = -[\, \nabla\varphi(v) \,]_{\widehat{F}}$; $d_{A} = 0$;
        if $[\, v + d \,]_{F} < 0$ **then**
            $k \leftarrow \underset{i \in \mathcal{F},\, d_i < 0}{\operatorname{argmin}} \; v_i/(-d_i)$;
            $\alpha_{\max} \leftarrow v_k/(-d_k)$; [Compute the maximum feasible step]
            $\alpha_{\text{opt}} = -\nabla\varphi(v)^{\mathrm{T}} d / d^{\mathrm{T}} H^{M} d$;
            $\alpha = \min\{\, \alpha_{\text{opt}}, \alpha_{\max} \,\}$;
            $\mathcal{A} \leftarrow \mathcal{A} \cup \{\, k \,\}$, $\mathcal{F} \leftarrow \mathcal{F} \setminus \{\, k \,\}$; [fix v_k on its bound]
            $v \leftarrow v + \alpha d$;
        else
            $v \leftarrow v + d$;
        end if
        Set $\widehat{\mathcal{F}} = \mathcal{F} \cup \{\, 1, 2, \ldots, m \,\}$;
    until $\|[\, \nabla\varphi(v) \,]_{\widehat{F}}\| = 0$
    $w \leftarrow \nabla\varphi(v)$; $w_{\min} \leftarrow \min_{i \in \mathcal{A}} w_i$; $\ell \leftarrow \underset{i \in \mathcal{A}}{\operatorname{argmin}} \; w_i$;
    if $w_{\min} < 0$ **then**
        $\mathcal{A} \leftarrow \mathcal{A} \setminus \{\, \ell \,\}$; $\mathcal{F} \leftarrow \mathcal{F} \cup \{\, \ell \,\}$; [free v_ℓ from its bound]
    end if
    Set $\widehat{\mathcal{F}} = \mathcal{F} \cup \{\, 1, 2, \ldots, m \,\}$;
until $w_{\min} \geq 0$
return $(x,\ y) = v$;

Chapter 4

Modifying Matrix Factorizations

As mentioned in Section 1.4, both interior and SQP methods require modification of a $(n+m)$-dimensional KKT matrix of the general form

$$K = \begin{pmatrix} H & J^{\mathrm{T}} \\ J & -D \end{pmatrix},$$

where H is $n \times n$ and symmetric, J is $m \times n$, and D is positive semidefinite and diagonal. Here we regard these values as arbitrary constants of the correct dimension, while keeping in mind they typically represent quantities associated with the current state of an optimization algorithm, e.g., $H \equiv H(x_k, y_k)$, $J \equiv J(x_k)$ and, in the SQP context, $D \equiv \mu I_m$ (see Table 1.1 for the relevant definitions). In the situation where D is positive definite, we are interested in K because of its relationship with an approximate merit function Hessian, which can be written

$$H^{M} = \begin{pmatrix} H + 2J^{\mathrm{T}}D^{-1}J & J^{\mathrm{T}} \\ J & D \end{pmatrix}.$$

The link between K and H^M comes from the inertial relationships

$$\text{In}(H^M) = \text{In}(H + J^{\text{T}} D^{-1} J) + (m,\ 0,\ 0), \tag{4.1}$$

$$\text{In}(K) = \text{In}(H + J^{\text{T}} D^{-1} J) + (0,\ m,\ 0). \tag{4.2}$$

In what follows, the goal is make H^M positive definite *implicitly* by modifying K. In particular, only the $(1,1)$ or $(2,2)$ blocks of K can be modified. This is because the effect of perturbing J on the inertias of K and H^M is not clear in general.

4.1 Tiling

Define a $2m \times 2m$ matrix of tiles T by

$$T = \begin{pmatrix} T_{11} & T_{21}^{\text{T}} & \dots \\ T_{21} & T_{22} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix},$$

where each tile T_{ij} is a 2×2 matrix of the form

$$T_{ij} = \begin{pmatrix} h & a \\ b & -d \end{pmatrix},$$

and the elements $h \in H$, $a, b \in J$, and $d \in D$. Define H_{11} and J_1 to be the submatrices of H and J from which all the h, a, b elements appearing in tiles originate. This induces a partition of K

$$K = \begin{pmatrix} H_{11} & H_{21}^{\text{T}} & J_1^{\text{T}} \\ H_{21} & H_{22} & J_2^{\text{T}} \\ J_1 & J_2 & -D \end{pmatrix}.$$

Let P_1 be a permutation matrix that brings a checkered pattern of tiles into the leading principal submatrix of the KKT matrix K and let $C = P_1^{\mathrm{T}} K P_1$. Let P_2 be the permutation that symmetrically separates the submatrices $H_{11}, J_1, -D$ within T:

$$P_2^{\mathrm{T}} P_1^{\mathrm{T}} K P_1 P_2 = P_2^{\mathrm{T}} C P_2 = P_2^{\mathrm{T}} \begin{pmatrix} T & S^{\mathrm{T}} \\ S & H_{22} \end{pmatrix} P_2 = \begin{pmatrix} H_{11} & J_1^{\mathrm{T}} & H_{21}^{\mathrm{T}} \\ J_1 & -D & J_2 \\ H_{21} & J_2^{\mathrm{T}} & H_{22} \end{pmatrix} = \begin{pmatrix} \widetilde{T} & \widetilde{S}^{\mathrm{T}} \\ \widetilde{S} & H_{22} \end{pmatrix} = \widetilde{C},$$

with $\widetilde{S} = \begin{pmatrix} H_{21} & J_2^{\mathrm{T}} \end{pmatrix}$.

Proposition 4.1.1. *Let T be a nonsingular tile and*

$$P_2^{\mathrm{T}} C P_2 = P_2^{\mathrm{T}} \begin{pmatrix} T & S^{\mathrm{T}} \\ S & H_{22} \end{pmatrix} P_2 = \begin{pmatrix} \widetilde{T} & \widetilde{S}^{\mathrm{T}} \\ \widetilde{S} & H_{22} \end{pmatrix} = \widetilde{C}$$

If the permutation P_2 only permutes the first $2m$ columns, $C/T = \widetilde{C}/\widetilde{T}$.

Proof. Assuming P_2 only acts on the first $2m$ columns it must have the form

$$P_2 = \begin{pmatrix} Q_2 & 0 \\ 0 & I_{n-m} \end{pmatrix},$$

where Q_2 is also a permutation matrix. By equating blocks of C and $\widetilde{C}$ it is evident that $Q_2^{\mathrm{T}} T Q_2 = \widetilde{T}$

and $SQ_2 = \widetilde{S}$. It now follows that

$$\begin{aligned}
\widetilde{C}/\widetilde{T} &= H_{22} - \widetilde{S}\widetilde{T}^{-1}\widetilde{S}^{\mathrm{T}} \\
&= H_{22} - SQ_2(Q_2^{\mathrm{T}}TQ_2)^{-1}Q_2^{\mathrm{T}}S^{\mathrm{T}} \\
&= H_{22} - ST^{-1}S^{\mathrm{T}} \\
&= C/T.
\end{aligned}$$

□

The $D = 0$ case

The previous result is useful in the case when $D = 0$ for showing that the Schur complement of T is in fact the reduced Hessian.

Proposition 4.1.2. *If $D = 0$ and T is a nonsingular tiling then the Schur complement C/T is the reduced Hessian $Z^{\mathrm{T}}HZ$, where Z is the particular basis for* $\mathrm{null}(J)$ *given by*

$$Z = \begin{pmatrix} -J_1^{-1}J_2 \\ I_{n-m} \end{pmatrix}.$$

Proof. Define $W = \widetilde{T}^{-1}\widetilde{S}^{\mathrm{T}}$ and write out the system

$$\widetilde{T}W = \begin{pmatrix} H_{11} & J_1^{\mathrm{T}} \\ J_1 & 0 \end{pmatrix} \begin{pmatrix} W_1 \\ W_2 \end{pmatrix} = \begin{pmatrix} H_{21}^{\mathrm{T}} \\ J_2 \end{pmatrix} = \widetilde{S}^{\mathrm{T}}.$$

Carrying out the multiplication shows $W_1 = J_1^{-1}J_2$ and $W_2 = J_1^{-\mathrm{T}}(H_{21}^{\mathrm{T}} - H_{11}J_1^{-1}J_2)$ and conse-

quently

$$\begin{aligned}
\widetilde{C}/\widetilde{T} &= H_{22} - \widetilde{S}\widetilde{T}^{-1}\widetilde{S}^{\mathrm{T}} \\
&= H_{22} - (H_{21}W_1 + J_2^{\mathrm{T}}W_2) \\
&= H_{22} - H_{21}J_1^{-1}J_2 - J_2^{\mathrm{T}}J_1^{-\mathrm{T}}H_{21}^{\mathrm{T}} + J_2^{\mathrm{T}}J_1^{-\mathrm{T}}H_{11}J_1^{-1}J_2 \\
&= \begin{pmatrix} -J_2^{\mathrm{T}}J_1^{-\mathrm{T}} & I \end{pmatrix} \begin{pmatrix} H_{11} & H_{21}^{\mathrm{T}} \\ H_{21} & H_{22} \end{pmatrix} \begin{pmatrix} -J_1^{-1}J_2 \\ I \end{pmatrix} \\
&= Z^{\mathrm{T}}HZ.
\end{aligned}$$

As $\widetilde{C}/\widetilde{T} = C/T$, it follows $C/T = Z^{\mathrm{T}}HZ$. □

To relate the inertia of K with the reduced Hessian requires that J has full rank, and the QR factorization $J^{\mathrm{T}} = Q\begin{pmatrix} R^{\mathrm{T}} & 0 \end{pmatrix}^{\mathrm{T}} = \begin{pmatrix} Y & Z \end{pmatrix}\begin{pmatrix} R^{\mathrm{T}} & 0 \end{pmatrix}^{\mathrm{T}}$. Substituting this in for J and letting $U = \mathrm{diag}(Q,\ I_m)$ yields

$$U^{\mathrm{T}}KU = \begin{pmatrix} Y^{\mathrm{T}}HY & Y^{\mathrm{T}}HZ & R \\ Z^{\mathrm{T}}HY & Z^{\mathrm{T}}HZ & 0 \\ R^{\mathrm{T}} & 0 & 0 \end{pmatrix}.$$

Define the nonsingular matrix

$$V = \begin{pmatrix} I_m & 0 & -\frac{1}{2}Y^{\mathrm{T}}HYR^{-\mathrm{T}} \\ 0 & I_{n-m} & -Z^{\mathrm{T}}HR^{-\mathrm{T}} \\ 0 & 0 & R^{-\mathrm{T}} \end{pmatrix},$$

then direct multiplication shows

$$VU^{\mathrm{T}}KUV^{\mathrm{T}} = \begin{pmatrix} 0 & 0 & I_m \\ 0 & Z^{\mathrm{T}}HZ & 0 \\ I_m & 0 & 0 \end{pmatrix},$$

which has m positive and m negative unit eigenvalues along with the eigenvalues of $Z^{\mathrm{T}}HZ$. This implies $\mathrm{In}(K) = (m,\ m,\ 0) + \mathrm{In}(Z^{\mathrm{T}}HZ)$. In particular, this means there is a one-to-one correspondence between negative eigenvalues of the reduced Hessian and negative eigenvalues of K exceeding m. It follows

$$\mathrm{In}(T) = \mathrm{In}(K) - \mathrm{In}(Z^{\mathrm{T}}HZ) = (n-p,\ m+p,\ 0) - (n-m-p,\ p,\ 0) = (m,\ m,\ 0).$$

4.1.1 Two-stage factorization

An overall LBL^{T} factorization of K can be computed by piecing together two related factorizations; one of the permuted tiling $\widetilde{T}$ and the other of a Schur complement term that remains. Both factorizations can use state-of-the-art symmetric indefinite factorization software as is (e.g. `MA57`). Suppose such a factorization of $\widetilde{T}$ is computed:

$$Q_3^{\mathrm{T}}\widetilde{T}Q_3 = L_{11}B_1L_{11}^{\mathrm{T}}.$$

When these factors are inserted in place of $\widetilde{T}$ the result is

$$P_2^{\mathrm{T}}P_1^{\mathrm{T}}KP_1P_2 = \begin{pmatrix} Q_3 & 0 \\ 0 & I_{n-m} \end{pmatrix} \begin{pmatrix} L_{11}B_1L_{11}^{\mathrm{T}} & Q_3^{\mathrm{T}}\widetilde{S}^{\mathrm{T}} \\ \widetilde{S}Q_3 & H_{22} \end{pmatrix} \begin{pmatrix} Q_3^{\mathrm{T}} & 0 \\ 0 & I_{n-m} \end{pmatrix}.$$

Next, define $E = \widetilde{S}Q_3L_{11}^{-\mathrm{T}}$ and $P_3 = \mathrm{diag}(Q_3, I_{n-m})$ so that the triangular L_{11} term can be factored out

$$P_3^{\mathrm{T}}P_2^{\mathrm{T}}P_1^{\mathrm{T}}KP_1P_2P_3 = \begin{pmatrix} L_{11} & 0 \\ EB_1^{-1} & I_{n-m} \end{pmatrix} \begin{pmatrix} B_1 & 0 \\ 0 & H_{22} - EB_1^{-1}E^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} L_{11}^{\mathrm{T}} & B_1^{-1}E^{\mathrm{T}} \\ 0 & I_{n-m} \end{pmatrix}.$$

The (2, 2) block of the block diagonal matrix is the Schur complement term to be factored next, giving

$$H_{22} - EB_1^{-1}E^{\mathrm{T}} = Q_4 L_{22} B_2 L_{22}^{\mathrm{T}} Q_4^{\mathrm{T}}.$$

The definitions $L_{21} = Q_4^{\mathrm{T}} E B_1^{-1}$, $P_4 = \mathrm{diag}(I_{2m}, Q_4)$, and $P = P_1 P_2 P_3 P_4$ gives the complete factorization

$$P^{\mathrm{T}} K P = \begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{pmatrix} \begin{pmatrix} B_1 & 0 \\ 0 & B_2 \end{pmatrix} \begin{pmatrix} L_{11}^{\mathrm{T}} & L_{21}^{\mathrm{T}} \\ 0 & L_{22}^{\mathrm{T}} \end{pmatrix} = LBL^{\mathrm{T}}.$$

Factor modification

Any modification to K made indirectly by modification of B needs to only affect the H part of K. This two-stage factorization produces B_2 that can be modified safely, meaning the resulting change to K only changes H.

Proposition 4.1.3. *The two-stage factorization* (2, 2) *block can be modified, if needed, without any change to the J or D blocks of K. Specifically,*

$$\Delta K = \begin{pmatrix} * & 0 \\ 0 & 0_m \end{pmatrix}$$

Proof. Suppose a perturbation $\Delta B = \mathrm{diag}(0, \Delta B_2)$ is made to B, and keep in mind that P_2 and

P_3 only permute the first $2m$ rows and columns. It then follows

$$\begin{aligned}
\Delta K &= P \begin{pmatrix} 0 & 0 \\ 0 & L_{22}\Delta B_2 L_{22}^{\mathrm{T}} \end{pmatrix} P^{\mathrm{T}} \\
&= P_1 \begin{pmatrix} 0 & 0 \\ 0 & Q_4 L_{22}\Delta B L_{22}^{\mathrm{T}} Q_4^{\mathrm{T}} \end{pmatrix} P_1^{\mathrm{T}} \\
&= P_1 \begin{pmatrix} 0 & 0 \\ 0 & \Delta H_{22} \end{pmatrix} P_1^{\mathrm{T}} \\
&= \begin{pmatrix} * & 0 \\ 0 & 0 \end{pmatrix}.
\end{aligned}$$

The final equality holds P_1 was chosen to symmetrically move all elements of H_{22} to the trailing $(n-m) \times (n-m)$ submatrix. Therefore, applying them in the reverse order moves the trailing submatrix ΔH_{22} to the positions originally occupied by elements of H_{22}, which could be anywhere in the first n rows and columns. □

Recall when $D = 0$ that $\mathrm{In}(T) = (m, m, 0)$, and as $\mathrm{In}(T) = \mathrm{In}(B_1)$ it must hold that

$$\mathrm{In}(K) = \mathrm{In}(B) = \mathrm{In}(T) + \mathrm{In}(B_2).$$

If B_2 is sufficiently positive definite, then $\mathrm{In}(K) = (m,\ m,\ 0) + (n-m,\ 0,\ 0) = (n,\ m,\ 0)$ as desired. Otherwise, a perturbation to B_2 can be made to correct the inertia with the induced perturbation to K changing only the H_{22} block.

The $D \neq 0$ case

The inertial relationships invloving the Schur complement of a nonsingular tiling with $D \neq 0$ can be derived by

$$\begin{aligned}\mathrm{In}(C/T) &= \mathrm{In}(K) - \mathrm{In}(T) \\ &= \left\{\mathrm{In}(H + J^{\mathrm{T}}D^{-1}J) + \mathrm{In}(-D)\right\} - \left\{\mathrm{In}(H_{11} + J_1^{\mathrm{T}}D^{-1}J_1) + \mathrm{In}(-D)\right\} \\ &= \mathrm{In}(H + J^{\mathrm{T}}D^{-1}J) - \mathrm{In}(H_{11} + J_1^{\mathrm{T}}D^{-1}J_1).\end{aligned}$$

This hints at the fact that C/T might be the Schur complement of the augmented $m \times m$ matrix $H_{11} + J_1^{\mathrm{T}}D^{-1}J_1$ within the augmented $n \times n$ matrix $H + J^{\mathrm{T}}D^{-1}J$, and this is true if the former is nonsingular, which is shown next.

Proposition 4.1.4. *Let D is a positive-definite diagonal matrix and let H_1, J_1 denote the submatrices of H, J from which a nonsingular tile T is formed, such that*

$$T = \begin{pmatrix} H_{11} & J_1^{\mathrm{T}} \\ J_1 & -D \end{pmatrix} \qquad \textit{and} \qquad P_2^{\mathrm{T}} C P_2 = P_2^{\mathrm{T}} \begin{pmatrix} T & S^{\mathrm{T}} \\ S & H_{22} \end{pmatrix} P_2 = \begin{pmatrix} \widetilde{T} & \widetilde{S}^{\mathrm{T}} \\ \widetilde{S} & H_{22} \end{pmatrix} = \widetilde{C}$$

If $H_{11} + J_1^{\mathrm{T}}D^{-1}J_1$ is nonsingular, then the Schur complement of T in C is equal to the Schur complement of $H_{11} + J_1^{\mathrm{T}}D^{-1}J_1$ in $H + J^{\mathrm{T}}D^{-1}J$, that is,

$$C/T = \left\{H_{11} + J_1^{\mathrm{T}}D^{-1}J_1\right\} / \left\{H + J^{\mathrm{T}}D^{-1}J\right\}.$$

Proof. Following the strategy used to show $C/T = Z^{\mathrm{T}}HZ$ in the case where $D = 0$, define $W =$

$\widetilde{T}^{-1}\widetilde{S}^{\mathrm{T}}$ and write out the system

$$\begin{pmatrix} H_{11} & J_1^{\mathrm{T}} \\ J_1 & -D \end{pmatrix} \begin{pmatrix} W_1 \\ W_2 \end{pmatrix} = \begin{pmatrix} H_{21}^{\mathrm{T}} \\ J_2 \end{pmatrix}.$$

Carrying out the multiplication yields $W_2 = D^{-1}(J_1 W_1 - J_2)$ which can be used to eliminate W_2, giving $(H_{11} + J_1^{\mathrm{T}} D^{-1} J_1) W_1 = H_{21}^{\mathrm{T}} + J_1^{\mathrm{T}} D^{-1} J_2$, and therefore

$$W_1 = (H_{11} + J_1^{\mathrm{T}} D^{-1} J_1)^{-1} (H_{21}^{\mathrm{T}} + J_1^{\mathrm{T}} D^{-1} J_2).$$

Now the Schur complement can be computed.

$$\begin{aligned} \widetilde{C}/\widetilde{T} &= H_{22} - \widetilde{S}\widetilde{T}^{-1}\widetilde{S}^{\mathrm{T}} \\ &= H_{22} - H_{21} W_1 - J_2^{\mathrm{T}} W_2 \\ &= H_{22} - H_{21} W_1 - J_2^{\mathrm{T}} D^{-1}(J_1 W_1 - J_2) \\ &= H_{22} - (H_{21} + J_2^{\mathrm{T}} D^{-1} J_1) W_1 + J_2^{\mathrm{T}} D^{-1} J_2 \\ &= H_{22} + J_2^{\mathrm{T}} D^{-1} J_2 - (H_{21} + J_2^{\mathrm{T}} D^{-1} J_1)(H_{11} + J_1^{\mathrm{T}} D^{-1} J_1)^{-1} (H_{21} + J_2^{\mathrm{T}} D^{-1} J_1)^{\mathrm{T}} \end{aligned}$$

Notice the elements appearing are from $H + J^{\mathrm{T}} D^{-1} J$, partitioned conformably with C:

$$\begin{pmatrix} H_{11} & H_{21}^{\mathrm{T}} \\ H_{21} & H_{22} \end{pmatrix} + \begin{pmatrix} J_1^{\mathrm{T}} \\ J_2^{\mathrm{T}} \end{pmatrix} D^{-1} \begin{pmatrix} J_1 & J_2 \end{pmatrix} = \begin{pmatrix} H_{11} + J_1^{\mathrm{T}} D^{-1} J_1 & H_{21}^{\mathrm{T}} + J_1^{\mathrm{T}} D^{-1} J_2 \\ H_{21} + J_2^{\mathrm{T}} D^{-1} J_1 & H_{22} + J_2^{\mathrm{T}} D^{-1} J_2 \end{pmatrix}.$$

Forming the Schur complement of the $(1,\ 1)$ block in $H + J^{\mathrm{T}} D^{-1} J$ gives

$$H_{22} + J_2^{\mathrm{T}} D^{-1} J_2 + (H_{21} + J_2^{\mathrm{T}} D^{-1} J_1)(H_{11} + J_1^{\mathrm{T}} D^{-1} J_1)^{-1} (H_{21} + J_2^{\mathrm{T}} D^{-1} J_1)^{\mathrm{T}},$$

which agrees with what was just computed for $\widetilde{C}/\widetilde{T} = C/T$. $\square$

Even if D is a positive-definite diagonal matrix or multiple of identity the outlined tiling procedure need not result in the inertia of T being $(m,\ m,\ 0)$. In order for the two-stage strategy to work, the first stage must achieve the inertia $(m,\ m,\ 0)$. However, if D has the form μI then the procedure does work, provided $\mu > 0$ is sufficiently small.

Refactoring (Wächter-Biegler)

The correct inertia for the first stage can be achieved by repeated application of diagonal modifications to H_{11} and refactoring. Once a tile is selected, form

$$\widetilde{T}(\sigma) = \begin{pmatrix} H_{11} + \sigma I & J_1^{\mathrm{T}} \\ J_1 & -D \end{pmatrix}$$

for increasing values of σ until $\mathrm{In}(T) = (m,\ m,\ 0)$.

Once the correct inertia is achieved, the two-stage factorization continues as described. If needed, a second perturbation would be made to H_{22} and the resulting overall modification would be contained to the first n rows and columns. As the tile is potentially significantly smaller than K, refactoring the tile could be computationally more efficient than repeated refactoring all of K.

Approximate perturbation

Another method of achieving the correct inertia is to use an approximation that is not norm-optimal, but guaranteed to produce the right inertia in a single modification to $\widetilde{T}$. This technique is based on the following result.

Proposition 4.1.5. *For a general nonsingular $n \times n$ symmetric matrix M and $n \times k$ matrix W,*

and for $s \leq k$,

$$i_+(M + WW^{\mathrm{T}}) + i_0(M + WW^{\mathrm{T}}) = i_+(M) + s$$

if and only if $-I_k - W^{\mathrm{T}}M^{-1}W$ *has exactly* s *nonnegative eigenvalues.*

Proof. This result follows from the congruences

$$\begin{pmatrix} M & 0 \\ 0 & -I_k - W^{\mathrm{T}}M^{-1}W \end{pmatrix} \sim \begin{pmatrix} M & W \\ W^{\mathrm{T}} & -I_k \end{pmatrix} \sim \begin{pmatrix} -I_k & W^{\mathrm{T}} \\ W & M \end{pmatrix} \sim \begin{pmatrix} -I_k & 0 \\ 0 & M + WW^{\mathrm{T}} \end{pmatrix},$$

which yield $\mathrm{In}(M + WW^{\mathrm{T}}) = \mathrm{In}(M) - \mathrm{In}(-I_k) - \mathrm{In}(-I_k - W^{\mathrm{T}}M^{-1}W)$.

If $-I_k - W^{\mathrm{T}}M^{-1}W$ has exactly s nonnegative eigenvalues then its inertia can be written $(s - l,\ k - s,\ l)$ and therefore

$$\mathrm{In}(M + WW^{\mathrm{T}}) = \mathrm{In}(M) - (0,\ k,\ 0) + (s - l,\ k - s,\ l) = (i_+(M) + s - l,\ i_-(M) - s,\ l).$$

Adding the nonnegative eigenvalues gives

$$i_+(M + WW^{\mathrm{T}}) + i_0(M + WW^{\mathrm{T}}) = (i_+(M) + s - l) + l = i_+(M) + s.$$

□

This result can be applied to the $2m \times 2m$ matrix $\widetilde{T}$ by constructing a modification only affecting the $m \times m$ block H_{11}, i.e.,

$$W = \begin{pmatrix} \gamma I_m \\ 0 \end{pmatrix},$$

The resulting perturbation to H_{11} is $\gamma^2 I$. The problem is then to find γ that minimizes $\|\Delta H_{11}\|$ subject to $-I_m - W^{\mathrm{T}}\widetilde{T}^{-1}W$ having s nonnegative eigenvalues. Define G to be the first m rows and

columns of $\widetilde{T}^{-1}$, then the constraint is equivalent to $\lambda_s(-I_m-\gamma^2 G) \geq 0$ for $i \in \{1,\ldots,s\}$. Observe that $\lambda_s(-I_m-\gamma^2 G) = -1-\gamma^2\lambda_{m-s+1}(G)$ and that it is assumed $\lambda_{m-s+1}(G) < 0$ (otherwise the constraint is already met), therefore the requirement reduces to

$$\gamma^2 \geq \frac{1}{-\lambda_{m-s+1}(G)}.$$

As $\widetilde{T},\ G$ satisfy

$$\widetilde{T}^{-1} = \begin{pmatrix} G & * \\ * & * \end{pmatrix},$$

Cauchy's eigenvalue interlacing theorem implies that

$$\lambda_{k+m}(\widetilde{T}^{-1}) \leq \lambda_k(G) \leq \lambda_k(\widetilde{T}^{-1}) \qquad \text{for} \qquad k \in \{1,\ldots,m\}.$$

The factorization $\widetilde{T} = Q_3 L_{11} B_1 L_{11}^{\mathrm{T}} Q_3^{\mathrm{T}}$ can be used to obtain an upper bound

$$\begin{aligned} \lambda_{m-s+1}(G) &\leq \lambda_{m-s+1}(\widetilde{T}^{-1}) \\ &= \lambda_{m-s+1}(L_{11}^{-\mathrm{T}} B_1^{-1} L_{11}^{-1}) \\ &= \lambda_{m-s+1}(B_1^{-1})\theta \end{aligned}$$

for some $\lambda_{2m}\left((L_{11}L_{11}^{\mathrm{T}})^{-1}\right) \leq \theta \leq \lambda_1\left((L_{11}L_{11}^{\mathrm{T}})^{-1}\right)$. As $L_{11}L_{11}^{\mathrm{T}}$ is automatically positive definite, it holds that

$$\lambda_{2m}\left((L_{11}L_{11}^{\mathrm{T}})^{-1}\right) = \frac{1}{\lambda_1(L_{11}L_{11}^{\mathrm{T}})} \qquad \text{and} \qquad \lambda_1\left((L_{11}L_{11}^{\mathrm{T}})^{-1}\right) = \frac{1}{\lambda_{2m}(L_{11}L_{11}^{\mathrm{T}})}.$$

Suppose $\widetilde{T}$ doesn't have the correct inertia, and that $\mathrm{In}(\widetilde{T}) = (m-s,\ m+s,\ 0)$ with $s > 0$.

This means that $\lambda_{m-s+1}(\widetilde{T}) < 0$ and therefore $\lambda_{m-s+1}(B_1) < 0$. The eigenvalues of B_1^{-1} are the reciprocals of the eigenvalues of B_1, and it must hold that $\lambda_{m-s+1}(B_1^{-1}) < 0$ as well. This is because taking the reciprocals only reverses the order within negatives and positives but does not alter their sign. The goal is then to select the smallest γ for which

$$\lambda_{m-s+1}(G) \leq \frac{\lambda_{m-s+1}(B_1^{-1})}{\lambda_1(L_{11}L_{11}^{\mathrm{T}})} \leq -\frac{1}{\gamma^2}.$$

Assuming L_{11} is computed using `MA57`, its entries are bounded by some constant ρ, e.g., $\rho = 2.781$ for the bounded Bunch-Kauffman pivoting strategy, and that

$$\lambda_1(L_{11}L_{11}^{\mathrm{T}}) = \|L_{11}L_{11}^{\mathrm{T}}\|_2 \leq \|L_{11}\|_F^2 \leq m + \frac{\rho^2}{2}m(m-1).$$

This implies that γ can be chosen such that

$$\gamma^2 = \frac{\|L_{11}\|_F^2}{-\lambda_{m-s+1}(B_1^{-1})} \qquad \text{or} \qquad \gamma^2 = \frac{m(\rho^2(m-1)+2)}{-2\lambda_{m-s+1}(B_1^{-1})}.$$

The latter would only be chosen to avoid computing the norm of L_{11}. Then it must hold that the perturbation increases the number of nonnegative eigenvalues by s and therefore $\mathrm{In}(\widetilde{T}(\gamma^2)) = (m,\ m,\ 0)$. The perturbed matrix would need to be refactored before continuing on and factoring its Schur complement.

4.2 First-Stage Strategy

In the case when $D \neq 0$, tiling may not be helpful. Perhaps the two-stage factorization can still be used, but with a different matrix than $\widetilde{T}$ for the first factorization. The goal here is

to find a way to construct a matrix K_{11} that can be brought to the leading principal submatrix position within K by symmetric permutation such that $\text{In}(K_{11}) = (l,\ m,\ 0)$ for some $l > 0$, and that

$$K_{11} = \begin{pmatrix} H_{11} & J_1^{\mathrm{T}} \\ J_1 & -D \end{pmatrix},$$

with $H_{11} \in_R [l \times l]$.

From the inertial relationship $\text{In}(K_{11}) = \text{In}(H_{11} + J_1^{\mathrm{T}} D^{-1} J_1) + \text{In}(-D)$, it would be ideal to find submatrices for which $H_{11} + J_1^{\mathrm{T}} D^{-1} J_1$ is positive definite. To simplify the search, note that

$$\begin{aligned} \lambda_l(H_{11} + J_1^{\mathrm{T}} D^{-1} J_1) &= \inf_{x \neq 0} \left\{ \frac{x^{\mathrm{T}}(H_{11} + J_1^{\mathrm{T}} D^{-1} J_1)x}{x^{\mathrm{T}} x} \right\} \\ &= \inf_{x \neq 0} \left\{ \frac{x^{\mathrm{T}} H_{11} x}{x^{\mathrm{T}} x} + \frac{\left\| D^{-\frac{1}{2}} J_1 x \right\|^2}{x^{\mathrm{T}} x} \right\} \\ &> \inf_{x \neq 0} \left\{ \frac{x^{\mathrm{T}} H_{11} x}{x^{\mathrm{T}} x} \right\} = \lambda_l(H_{11}), \end{aligned}$$

so if H_{11} can be chosen positive semidefinite then $\text{In}(K_{11}) = (l,\ m,\ 0)$.

4.2.1 Submatrix search

Consider the following recursive algorithm for constructing H_{11}. For simplicity, drop the subscripts on H_{11} and J_1 so that $H_k = (H_{11})^{(k)}$ is the kth result of the following process, and $H_l = (H_{11})^{(l)} = H_{11}$. The idea here is that, starting with a positive scalar H_0, a leading submatrix will be built by appending a trailing border. The components of the border will be chosen so that the submatrix can be gathered using symmetric permutations while retaining positive definiteness.

Let $H_0 = \min\{h_{ii} \ :\ h_{ii} > 0\}$, i.e., the smallest positive diagonal element of H. Then H_0 is positive definite. Now suppose that at the kth stage of this process that H_k is positive definite

and

$$P_k^{\mathrm{T}} H P_k = \begin{pmatrix} H_k & J_k^{\mathrm{T}} \\ J_k & S_k \end{pmatrix}.$$

Then a row b_k^{T} of J_k and the corresponding diagonal element c_k of S_k are sought such that

$$H_{k+1} = \begin{pmatrix} H_k & b_k \\ b_k^{\mathrm{T}} & c_k \end{pmatrix}$$

is positive definite. As $\mathrm{In}(H_{k+1}) = \mathrm{In}(H_k) + \mathrm{In}(c_k - b_k^{\mathrm{T}} H_k^{-1} b_k)$, adding such a border retains positive definiteness if and only if $c_k - b_k^{\mathrm{T}} H_k^{-1} b_k > 0$. This part of the algorithm terminates when $c \leq b^{\mathrm{T}} H_k^{-1} b$ for all $b = e_j^{\mathrm{T}} J_k$, $c = e_j^{\mathrm{T}} S_k e_j$, with $j \in \{1, \ldots, n - l_k\}$.

The result of the process is a positive-definite matrix $H_{11} = (H_{11})^{(l)}$ and a permutation matrix P_l such that

$$P_l^{\mathrm{T}} H P_l = \begin{pmatrix} H_{11} & H_{21}^{\mathrm{T}} \\ H_{21} & H_{22}, \end{pmatrix}$$

and so the KKT matrix K can be permuted as

$$\begin{pmatrix} P_l^{\mathrm{T}} & 0 \\ 0 & I \end{pmatrix} K \begin{pmatrix} P_l & 0 \\ 0 & I \end{pmatrix} = \begin{pmatrix} P_l^{\mathrm{T}} H P_l & P_l^{\mathrm{T}} J^{\mathrm{T}} \\ J P_l & -D \end{pmatrix} = \begin{pmatrix} H_{11} & H_{21}^{\mathrm{T}} & J_1^{\mathrm{T}} \\ H_{21} & H_{22} & J_2^{\mathrm{T}} \\ J_1 & J_2 & -D \end{pmatrix}.$$

By permuting this result, it is straightforward to obtain the permutation P such that

$$P^{\mathrm{T}} K P = \begin{pmatrix} H_{11} & J_1^{\mathrm{T}} & H_{21}^{\mathrm{T}} \\ J_1 & -D & J_2 \\ H_{21} & J_2^{\mathrm{T}} & H_{22} \end{pmatrix} = \begin{pmatrix} K_{11} & K_{21}^{\mathrm{T}} \\ K_{21} & K_{22} \end{pmatrix}$$

with H_{11} positive definite and therefore $\mathrm{In}(K_{11}) = (l,\ m,\ 0)$.

Principal inverse maintenance

Because of the relationship between H_k and H_{k+1}, the inverse can be maintained rather than recomputed.

Proposition 4.2.1. *For a general nonsingular symmetric matrix A, nonzero vector b, and nonzero scalar c, define w such that $Aw = b$. Then, if $c - b^{\mathrm{T}}w \neq 0$,*

$$\begin{pmatrix} A & b \\ b^{\mathrm{T}} & c \end{pmatrix}^{-1} = \frac{1}{c - b^{\mathrm{T}}w} \begin{pmatrix} (c - b^{\mathrm{T}}w)A^{-1} + ww^{\mathrm{T}} & -w \\ -w^{\mathrm{T}} & 1 \end{pmatrix}$$

This result can be checked by direct multiplication.

Therefore, starting with $H_0^{-1} = 1/H_0$, all that is needed to compute H_{k+1}^{-1} is to form $w = H_k^{-1}b_k$ and substitute into the given expression.

Refined submatrix search

It may be that requiring a positive-definite H_{11} is too restrictive. For example, if only a few variables appear nonlinearly in the objective function then the largest positive-definite submatrix could be relatively small. What is actually required to produce K_{11} with the inertia $(l, m, 0)$ is that $H_{11} + J_1^{\mathrm{T}}D^{-1}J_1$ is positive definite. We now focus on the case where $D = \mu I_m$, and extend the submatrix search idea to construct K_{11} recursively.

The process begins with a given matrix

$$K = \begin{pmatrix} H & J^{\mathrm{T}} \\ J & -\mu I_m \end{pmatrix},$$

and proceeds by recursively constructing

$$K_i = \begin{pmatrix} H_i & J_i^{\mathrm{T}} \\ J_i & -\mu I_m \end{pmatrix},$$

where $K_i = (K_{11})^{(i)}$, $H_i = (H_{11})^{(i)}$, and $J_i = (J_1)^{(i)}$. The submatrices H_i and J_i are expanded by

$$H_{i+1} = \begin{pmatrix} H_i & b_{i+1} \\ b_{i+1}^{\mathrm{T}} & c_{i+1} \end{pmatrix} \text{ and } J_{i+1} = \begin{pmatrix} J_i & a_{i+1} \end{pmatrix},$$

where b_{i+1} is $i \times 1$, c_i is a scalar, and a_i is $m \times 1$. This can be thought of as inserting a "cross" shape into the center of K_i, i.e.,

$$K_{i+1} = \begin{pmatrix} H_i & b_{i+1} & J_i^{\mathrm{T}} \\ b_{i+1}^{\mathrm{T}} & c_{i+1} & a_{i+1}^{\mathrm{T}} \\ J_i & a_{i+1} & -\mu I_m \end{pmatrix}.$$

As the base of recursion, a diagonal element $H_1 = \begin{pmatrix} c_1 \end{pmatrix}$ and a column a_1 of J are selected such that $c_1 + a_1^{\mathrm{T}} a_1/\mu > 0$. With this choice, the inertia of K_1 can be deduced from

$$K_1 = \begin{pmatrix} c_1 & a_1^{\mathrm{T}} \\ a_1 & -\mu I \end{pmatrix} \sim \begin{pmatrix} -\mu I & \\ & c_1 + \frac{1}{\mu} a_1^{\mathrm{T}} a_1 \end{pmatrix} \qquad \text{thus} \qquad \mathrm{In}(K_1) = (1, m, 0).$$

Now suppose for some $i \geq 1$ we have that $H_i + \frac{1}{\mu} J_i^{\mathrm{T}} J_i$ is positive definite, and consequently that $\mathrm{In}(K_i) = (i, m, 0)$. Define intermediate quantities $E_i = H_i + \frac{1}{\mu} J_i^{\mathrm{T}} J_i$, $u_{i+1} = b_{i+1} + \frac{1}{\mu} J_i^{\mathrm{T}} a_{i+1}$, and $d_{i+1} = c_{i+1} + \frac{1}{\mu} a_{i+1}^{\mathrm{T}} a_{i+1}$. Choosing these values is essentially selecting a pairing of a column of H with a column of J because the scalar c_i is determined by the column chosen to define b_i. The

critical step is choosing b_{i+1}, c_{i+1} and a_{i+1} such that

$$d_{i+1} - u_{i+1}^{\mathrm{T}} E_i^{-1} u_{i+1} > 0. \tag{4.3}$$

The reason being, that

$$\begin{aligned} H_{i+1} + \frac{1}{\mu} J_{i+1}^{\mathrm{T}} J_{i+1} &= \begin{pmatrix} H_i + \frac{1}{\mu} J_i^{\mathrm{T}} J_i & b_{i+1} + \frac{1}{\mu} J_i^{\mathrm{T}} a_{i+1} \\ b_{i+1}^{\mathrm{T}} + \frac{1}{\mu} a_{i+1}^{\mathrm{T}} J_i & c_{i+1} + \frac{1}{\mu} a_{i+1}^{\mathrm{T}} a_{i+1} \end{pmatrix} \\ &= \begin{pmatrix} E_i & u_{i+1} \\ u_{i+1}^{\mathrm{T}} & d_{i+1} \end{pmatrix} \sim \begin{pmatrix} E_i & \\ & d_{i+1} - u_{i+1}^{\mathrm{T}} E_i^{-1} u_{i+1} \end{pmatrix}. \end{aligned} \tag{4.4}$$

Now, as E_i is positive definite by hypothesis, it follows $\mathrm{In}(K_{i+1}) = (i+1, m, 0)$ if and only if E_{i+1} is positive definite, which is true if and only if $d_i - u_i^{\mathrm{T}} E_i^{-1} u_i > 0$. As E_{i+1} satisfies

$$E_{i+1} = \begin{pmatrix} E_i & u_{i+1} \\ u_{i+1}^{\mathrm{T}} & d_{i+1} \end{pmatrix},$$

the same inverse maintenance algorithm (4.2.1) can be used to compute each E_i^{-1}.

When no pairing of the columns of H and J produces b_i, c_i, a_i such that (4.3) holds, the algorithm terminates and returns a permutation matrix P such that

$$P^{\mathrm{T}} K P = \begin{pmatrix} K_{11} & S^{\mathrm{T}} \\ S & H_{22} \end{pmatrix} = \begin{pmatrix} H_{11} & J_1^{\mathrm{T}} & H_{21}^{\mathrm{T}} \\ J_1 & -\mu I_m & J_2 \\ H_{21} & J_2^{\mathrm{T}} & H_{22} \end{pmatrix},$$

where $\mathrm{In}(K_{11}) = (l, m, 0)$. Note that H_{11} need not be positive definite.

If H_{11} is $m \times m$ then K_{11} is $2m \times 2m$ and this method is equivalent to a tiling, because

then

$$K_{11} = \begin{pmatrix} H_{11} & J_1^{\mathrm{T}} \\ J_1 & -\mu I \end{pmatrix} \sim \begin{pmatrix} T_{11} & T_{12} & \dots \\ T_{12}^{\mathrm{T}} & T_{22} & \dots \\ \vdots & \vdots & \ddots \end{pmatrix} = T.$$

The algorithm also affords the opportunity to decrease μ on the fly if doing so could result in $H_i + J_i^{\mathrm{T}} J_i/\mu$ becoming positive definite. Whether reducing μ can achieve this depends on the null spaces of H_i and J_i being "complementary". This requirement is exactly stated by Debreu's lemma, i.e., the reduced Hessian $Z_i^{\mathrm{T}} H_i Z_i$ must be positive definite.

One practical limitation of this approach is the potential for nearly singular K_{11}, making the formation of the Schur complement of K_{11} unstable. There are essentially two nested "layers" of Sylvester's Law of Inertia involved here at each stage of the described process. On one hand, K_{i+1} is second-order consistent if and only if $E_{i+1} = H_{i+1} + J_{i+1}^{\mathrm{T}} J_{i+1}/\mu$ is positive definite, which holds by nonsingular symmetric transformation. On the other hand, E_{i+1} is positive definite if and only if $d_{i+1} - u_{i+1}^{\mathrm{T}} E_i^{-1} u_{i+1} > 0$, which holds by another distinct nonsingular symmetric transformation (4.4). The question of how "near to singular" E_{i+1} can be so that K_{i+1} is "sufficiently" nonsingular is difficult to answer, and even if one knew, choosing a minimum tolerance for $d_{i+1} - u_{i+1}^{\mathrm{T}} E_i^{-1} u_{i+1} > 0$ that would achieve the needed positive definiteness of E_{i+1} is equally difficult.

4.3 Two-Stage Symmetric Indefinite factorization with Partial Cholesky Decomposition

A partial Cholesky decomposition can be used to determine a sequence of symmetric permutations that will gather a positive-definite leading submatrix for the first stage. The partial Cholesky algorithm also computes a decomposition of the leading submatrix, and it will be shown

how to use this decomposition as part of the multistage factorization of the KKT matrix

$$K = \begin{pmatrix} H & J^{\mathrm{T}} \\ J & -D \end{pmatrix},$$

with D positive definite and diagonal.

As H is not assumed positive definite, the classical Cholesky decomposition may not exist. Instead, applying the partial Cholesky decomposition algorithm to H yields a permutation matrix P_0 such that

$$P_0^{\mathrm{T}} H P_0 = \begin{pmatrix} H_{11} & H_{21}^{\mathrm{T}} \\ H_{21} & H_{22} \end{pmatrix},$$

with $H_{11} \in \mathbb{R}^{l \times l}$ and positive definite. The dimension of H_{11} will depend on properties of H. Extending P_0 by I_m yields a permutation that will permute the first n rows and columns of K. Define Π_0 to be its product with another permutation acting on the last $n+m-l$ rows and columns such that

$$\Pi_0^{\mathrm{T}} K \Pi_0 = \begin{pmatrix} H_{11} & J_1^{\mathrm{T}} & H_{21}^{\mathrm{T}} \\ J_1 & -D & J_2 \\ H_{21} & J_2^{\mathrm{T}} & H_{22} \end{pmatrix} = \begin{pmatrix} K_{11} & K_{21}^{\mathrm{T}} \\ K_{21} & K_{22} \end{pmatrix},$$

with $\begin{pmatrix} J_1 & J_2 \end{pmatrix} = JP_0$. Note that $H_{11} + J_1^{\mathrm{T}} D^{-1} J_1$ inherits positive definiteness from H_{11}, so it holds that

$$\mathrm{In}(K_{11}) = \mathrm{In}(H_{11} + J_1^{\mathrm{T}} D^{-1} J_1) + \mathrm{In}(-D) = (l,\ m,\ 0).$$

The standard LBL^{T} factorization of K_{11} is computed such that $P_1^{\mathrm{T}} K_{11} P_1 = L_{11} B_1 L_{11}^{\mathrm{T}}$, then with $\Pi_1 = \mathrm{diag}(P_1,\ I)$ and $E = K_{21} P_1 L_{11}^{-\mathrm{T}}$ one has

$$\Pi_1^{\mathrm{T}} \Pi_0^{\mathrm{T}} K \Pi_0 \Pi_1 = \begin{pmatrix} L_{11} & 0 \\ E B_1^{-1} & I \end{pmatrix} \begin{pmatrix} B_1 & 0 \\ 0 & H_{22} - E B_1^{-1} E^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} L_{11}^{\mathrm{T}} & B_1^{-1} E^{\mathrm{T}} \\ 0 & I \end{pmatrix}.$$

The next stage is to factor the Schur complement giving

$$H_{22} - EB_1^{-1}E^{\mathrm{T}} = P_2 L_{22} B_2 L_{22}^{\mathrm{T}} P_2^{\mathrm{T}}.$$

Let $\Pi_2 = \mathrm{diag}(I,\ P_2)$, $\Pi = \Pi_0 \Pi_1 \Pi_2$ and $L_{21} = P_2^{\mathrm{T}} E B_1^{-1}$ so that

$$\Pi^{\mathrm{T}} K \Pi = \begin{pmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{pmatrix} \begin{pmatrix} B_1 & 0 \\ 0 & B_2 \end{pmatrix} \begin{pmatrix} L_{11}^{\mathrm{T}} & L_{21}^{\mathrm{T}} \\ 0 & L_{22}^{\mathrm{T}} \end{pmatrix}.$$

It is worth emphasizing that the factors computed during the Cholesky algorithm are not used, only the permutation that defines the positive-definite H_{11}. The next section indicates how the computed factors are used.

Computation details

The main computational ingredients of the split LBL^{T} factorization with partial Cholesky decomposition are described here in more detail.

1. The partial Cholesky decomposition of H (size: n),
2. The LBL^{T} factorization of K_{11} (size: $l + m$),
3. Computing $E = K_{21} P_1 L_{11}^{-\mathrm{T}}$ (size: $m \times (l + m)$). This can be done efficiently using

$$E = \texttt{linsolve}(L_{11},\ P_1^{\mathrm{T}} K_{21}^{\mathrm{T}},\ \texttt{LT = true})^{\mathrm{T}}$$

   as L_{11} is lower triangular.
4. Form the Schur complement $H_{22} - EB_1^{-1}E^{\mathrm{T}}$ (size: $n - l$). Note that this involves inverting

B_1 which can be done efficiently because B_1 is block diagonal. Also, it involves the quantity EB_1^{-1} which is needed for forming L_{21}.

5. The LBL^{T} factorization of $H_{22} - EB_1^{-1}E^{\mathrm{T}}$ (size: $n-l$)

6. Solving the system $\Pi LBL^{\mathrm{T}}\Pi^{\mathrm{T}}p = b$ for some right-hand-side vector b. Let $\pi = \Pi^{\mathrm{T}}p$ and $\beta = \Pi^{\mathrm{T}}b$, then one has

$$\begin{pmatrix} L_{11} & \\ L_{21} & L_{22} \end{pmatrix} \begin{pmatrix} B_1 & \\ & B_2 \end{pmatrix} \begin{pmatrix} L_{11}^{\mathrm{T}} & L_{21}^{\mathrm{T}} \\ & L_{22}^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} \pi_1 \\ \pi_2 \end{pmatrix} = \begin{pmatrix} \beta_1 \\ \beta_2 \end{pmatrix}.$$

This is solved by first solving $Lq = \beta$, followed by $L^{\mathrm{T}}\pi = \bar{B}^{-1}q$. Explicitly, this can be carried out as four calls to `linsolve()`; two upper-triangular and two lower-triangular:

$$\begin{aligned} L_{11}q_1 &= \beta_1 && \text{size: } l+m \\ L_{22}q_2 &= \beta_2 - L_{21}q_1 && \text{size: } n-l \\ L_{22}^{\mathrm{T}}\pi_2 &= \bar{B}_2^{-1}q_2 && \text{size: } n-l \\ L_{11}^{\mathrm{T}}\pi_1 &= B_1^{-1}q_1 - L_{21}^{\mathrm{T}}\pi_2 && \text{size: } l+m \end{aligned}$$

4.3.1 Utilizing the partial Cholesky factors

The partial Cholesky algorithm applied to H produces a permutation P_0, a unit lower-triangular $R_{11} \in_R [l \times l]$, $R_{21} \in_R [(n-l) \times l]$, a positive-definite diagonal B_1 such that

$$P_0^{\mathrm{T}} H P_0 = \begin{pmatrix} R_{11} & 0 \\ R_{21} & I \end{pmatrix} \begin{pmatrix} B_{11} & 0 \\ 0 & H_{22} \end{pmatrix} \begin{pmatrix} R_{11}^{\mathrm{T}} & R_{21}^{\mathrm{T}} \\ 0 & I \end{pmatrix} = \begin{pmatrix} H_{11} & H_{21}^{\mathrm{T}} \\ H_{21} & H_{22} \end{pmatrix},$$

with $H_{11} = R_{11}B_{11}R_{11}^{\mathrm{T}} \in_R [l \times l]$ positive definite. This can be written simply as

$$P_0^{\mathrm{T}} H P_0 = L_0 B_0 L_0^{\mathrm{T}} \qquad \text{where} \qquad L_0 = \begin{pmatrix} R_{11} & 0 \\ R_{21} & I \end{pmatrix} \text{ and } B_0 = \begin{pmatrix} B_{11} & 0 \\ 0 & H_{22} \end{pmatrix}.$$

Define $\begin{pmatrix} J_1 & J_2 \end{pmatrix} = JP_0$ and $\Pi_0 = \mathrm{diag}(P_0,\ I)$ to get

$$\Pi_0^{\mathrm{T}} K \Pi_0 = \begin{pmatrix} P_0^{\mathrm{T}} H P_0 & (JP_0)^{\mathrm{T}} \\ JP_0 & -D \end{pmatrix} = \begin{pmatrix} H_{11} & H_{21}^{\mathrm{T}} & J_1^{\mathrm{T}} \\ H_{21} & H_{22} & J_2^{\mathrm{T}} \\ J_1 & J_2 & -D \end{pmatrix}.$$

Next, the triangular factors from the Cholesky decomposition are symmetrically factored out,

$$\begin{pmatrix} L_0 & 0 \\ 0 & I \end{pmatrix} \begin{pmatrix} B_0 & L_0^{-1} P_0^{\mathrm{T}} J^{\mathrm{T}} \\ JP_0 L_0^{-\mathrm{T}} & -D \end{pmatrix} \begin{pmatrix} L_0^{\mathrm{T}} & 0 \\ 0 & I \end{pmatrix}.$$

In this case, $E = \begin{pmatrix} E_1 & E_2 \end{pmatrix}$ is defined to be $E = JP_0 L_0^{-\mathrm{T}}$ with $E_1 = J_1 R_{11}^{-\mathrm{T}}$ and $E_2 = -J_1 R_{11}^{-\mathrm{T}} R_{21}^{\mathrm{T}} + J_2$, which gives

$$\Pi_0^{\mathrm{T}} K \Pi_0 = \begin{pmatrix} R_{11} & & \\ R_{21} & I & \\ & & I \end{pmatrix} \begin{pmatrix} B_{11} & & E_1^{\mathrm{T}} \\ & H_{22} & E_2^{\mathrm{T}} \\ E_1 & E_2 & -D \end{pmatrix} \begin{pmatrix} R_{11}^{\mathrm{T}} & R_{21}^{\mathrm{T}} & \\ & I & \\ & & I \end{pmatrix}.$$

To get a first stage with the correct inertia, B_0 needs to be paired with $-D$. Let Π_1 be the permutation that exchanges the middle $n-l$ with the last m rows and columns, then

$$\begin{aligned}\Pi_1^{\mathrm{T}}\Pi_0^{\mathrm{T}}K\Pi_0\Pi_1 &= \Pi_1^{\mathrm{T}}\begin{pmatrix} R_{11} & & \\ R_{21} & I & \\ & & I \end{pmatrix}\Pi_1\Pi_1^{\mathrm{T}}\begin{pmatrix} B_{11} & & E_1^{\mathrm{T}} \\ & H_{22} & E_2^{\mathrm{T}} \\ E_1 & E_2 & -D \end{pmatrix}\Pi_1\Pi_1^{\mathrm{T}}\begin{pmatrix} R_{11}^{\mathrm{T}} & R_{21}^{\mathrm{T}} & \\ & I & \\ & & I \end{pmatrix}\Pi_1 \\ &= \begin{pmatrix} R_{11} & & \\ & I & \\ R_{21} & & I \end{pmatrix}\begin{pmatrix} B_{11} & E_1^{\mathrm{T}} & \\ E_1 & -D & E_2 \\ & E_2^{\mathrm{T}} & H_{22} \end{pmatrix}\begin{pmatrix} R_{11}^{\mathrm{T}} & & R_{21}^{\mathrm{T}} \\ & I & \\ & & I \end{pmatrix}.\end{aligned}$$

The Schur complement of $-D$ must be formed and factored next. Note that $D + E_1B_{11}^{-1}E_1^{\mathrm{T}}$ is positive definite and so the $(2,\ 2)$ block retains the needed m negative eigenvalues. Suppose

$$D + E_1B_{11}^{-1}E_1^{\mathrm{T}} = P_2L_{22}B_{22}L_{22}^{\mathrm{T}}P_2^{\mathrm{T}},$$

with B_{22} diagonal and positive definite, and define $\Pi_2 = \mathrm{diag}(I_l,\ P_2,\ I_{n-l})$. Then $\Pi_2^{\mathrm{T}}\Pi_1^{\mathrm{T}}\Pi_0^{\mathrm{T}}K\Pi_0\Pi_1\Pi_2$ has the form

$$\begin{pmatrix} R_{11} & & \\ P_2^{\mathrm{T}}E_1B_{11}^{-1} & L_{22} & \\ R_{21} & & I \end{pmatrix}\begin{pmatrix} B_{11} & & \\ & -B_{22} & L_{22}^{-1}P_2^{\mathrm{T}}E_2 \\ & E_2^{\mathrm{T}}P_2L_{22}^{-\mathrm{T}} & H_{22} \end{pmatrix}\begin{pmatrix} R_{11}^{\mathrm{T}} & B_{11}^{-1}E_1^{\mathrm{T}}P_2 & R_{21}^{\mathrm{T}} \\ & L_{22}^{\mathrm{T}} & \\ & & I \end{pmatrix}.$$

To reduce the notation a bit, let $B_1 = \mathrm{diag}(B_{11},\ -B_{22})$, $F = \begin{pmatrix} 0 & E_2^{\mathrm{T}}P_2L_{22}^{-\mathrm{T}} \end{pmatrix}$, $S = \begin{pmatrix} R_{21} & 0 \end{pmatrix}$, and

$$L_{11} = \begin{pmatrix} R_{11} & \\ P_2^{\mathrm{T}}E_1B_{11}^{-1} & L_{22} \end{pmatrix},$$

so that

$$\Pi_2^{\mathrm{T}}\Pi_1^{\mathrm{T}}\Pi_0^{\mathrm{T}}K\Pi_0\Pi_1\Pi_2 = \begin{pmatrix} L_{11} & \\ S & I \end{pmatrix}\begin{pmatrix} B_1 & F^{\mathrm{T}} \\ F & H_{22} \end{pmatrix}\begin{pmatrix} L_{11}^{\mathrm{T}} & S^{\mathrm{T}} \\ & I \end{pmatrix}.$$

Lastly, form and factor the complement of H_{22}

$$H_{22} - FB_1^{-1}F^{\mathrm{T}} = P_3 L_{22} B_2 L_{22}^{\mathrm{T}} P_3^{\mathrm{T}},$$

and define $\Pi_3 = \mathrm{diag}(I,\ P_3)$, and $\Pi = \Pi_0\Pi_1\Pi_2\Pi_3$. Also let $L_{21} = P_3^{\mathrm{T}}(S + FB_1^{-1})$. Then it follows that

$$\begin{aligned}
&\begin{pmatrix} L_{11} & \\ S & I \end{pmatrix}\begin{pmatrix} B_1 & F^{\mathrm{T}} \\ F & H_{22} \end{pmatrix}\begin{pmatrix} L_{11}^{\mathrm{T}} & S^{\mathrm{T}} \\ & I \end{pmatrix} \\
=&\begin{pmatrix} L_{11} & \\ S + FB_1^{-1} & I \end{pmatrix}\begin{pmatrix} B_1 & \\ & H_{22} - FB_1^{-1}F^{\mathrm{T}} \end{pmatrix}\begin{pmatrix} L_{11}^{\mathrm{T}} & S^{\mathrm{T}} + B_1^{-1}F^{\mathrm{T}} \\ & I \end{pmatrix} \\
=&\Pi_3\begin{pmatrix} L_{11} & \\ P_3^{\mathrm{T}}(S + FB_1^{-1}) & L_{22} \end{pmatrix}\begin{pmatrix} B_1 & \\ & B_2 \end{pmatrix}\begin{pmatrix} L_{11}^{\mathrm{T}} & (S^{\mathrm{T}} + B_1^{-1}F^{\mathrm{T}})P_3 \\ & L_{22}^{\mathrm{T}} \end{pmatrix}\Pi_3^{\mathrm{T}},
\end{aligned}$$

and therefore

$$\Pi^{\mathrm{T}}K\Pi = \begin{pmatrix} L_{11} & \\ L_{21} & L_{22} \end{pmatrix}\begin{pmatrix} B_1 & \\ & B_2 \end{pmatrix}\begin{pmatrix} L_{11}^{\mathrm{T}} & L_{21}^{\mathrm{T}} \\ & L_{22}^{\mathrm{T}} \end{pmatrix}.$$

4.4 Full Diagonal Modification of K

Suppose the approximate Hessian H^M of the merit function needs to be modified to get a positive-definite approximation. Rather than perturbing only the $(1,\ 1)$ block of H^M, consider a

perturbation that also effects the (2, 2) block of the form

$$H^M(\sigma) = H^M + \sigma T = \begin{pmatrix} H + 2J^{\mathrm{T}}D^{-1}J & J^{\mathrm{T}} \\ J & D \end{pmatrix} + \sigma \begin{pmatrix} M & 0 \\ 0 & N \end{pmatrix}.$$

To derive the corresponding perturbation to the KKT matrix, the perturbed Newton equations are premultiplied by the nonsingular matrix

$$U = \begin{pmatrix} I & -2J^{\mathrm{T}}D^{-1} \\ 0 & I \end{pmatrix},$$

which gives $UH^M(\sigma)\Delta v = -U\nabla M$. After some simplification, this reduces to

$$\begin{pmatrix} H + \sigma M & -J^{\mathrm{T}}(I + 2\sigma D^{-1}N) \\ J & D + \sigma N \end{pmatrix} \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = - \begin{pmatrix} \nabla f - J^{\mathrm{T}}y \\ D(y - \pi) \end{pmatrix}.$$

To symmetrize this system, let $\widehat{\Delta} y = -(I + 2\sigma D^{-1}N)\Delta y$, then an equivalent system is

$$\begin{pmatrix} H + \sigma M & J^{\mathrm{T}} \\ J & -(D + \sigma N)(I + 2\sigma D^{-1}N)^{-1} \end{pmatrix} \begin{pmatrix} \Delta x \\ \widehat{\Delta} y \end{pmatrix} = - \begin{pmatrix} \nabla f - J^{\mathrm{T}}y \\ D(y - \pi) \end{pmatrix}.$$

Thus, with $D(\sigma) = (D + \sigma N)(I + 2\sigma D^{-1}N)^{-1}$, the same Δv that solves the positive definite approximate Newton equations $H^M(\sigma)\Delta v = -\nabla M$ can be obtained by solving the system involving

$$K(\sigma) = \begin{pmatrix} H + \sigma M & J^{\mathrm{T}} \\ J & -D(\sigma) \end{pmatrix}.$$

This reduces exactly to the method of Wächter and Biegler [68] if $N \equiv 0$ and $M = I$.

The inertia relationships

$$\mathrm{In}(H^M(\sigma)) = \mathrm{In}(H + \sigma M + J^{\mathrm{T}} D(\sigma)^{-1} J) + (m,\ 0,\ 0)$$

$$\mathrm{In}(K(\sigma)) = \mathrm{In}(H + \sigma M + J^{\mathrm{T}} D(\sigma)^{-1} J) + (0,\ m,\ 0)$$

hold for all positive σ, so $H^M(\sigma)$ is positive definite if and only if $\mathrm{In}(K(\sigma)) = (n,\ m,\ 0)$. The question is whether or not perturbing D changes how sensitive $\mathrm{In}(K(\sigma))$ is to changes in σ. In the situation where N is a positive diagonal it holds that increasing σ actually decreases diagonal elements of $D(\sigma)$. Focus on a particular diagonal element of $d(\sigma) = [\, D(\sigma)\,]_{ii}$, then

$$d(\sigma) = [\, D\,]_{ii} \left(\frac{[\, D\,]_{ii} + \sigma [\, N\,]_{ii}}{[\, D\,]_{ii} + 2\sigma [\, N\,]_{ii}} \right).$$

The portion in parentheses approaches $\frac{1}{2}$ from above and so $d(\sigma)$ is strictly decreasing in σ. This means that for any $s > 0$ that the diagonal entries of $D(\sigma + s)^{-1}$ are strictly greater than those of $D(\sigma)^{-1}$. To study the sensitivity of $\mathrm{In}(K(\sigma))$ to changes in σ when D is perturbed, let's compare the case where $N = D$ with the case where $N = 0$ and consider the difference in the eigenvalues of $H + \sigma M + J^{\mathrm{T}} D^{-1} J$ and $H + \sigma M + J^{\mathrm{T}} D(\sigma)^{-1} J$ for increasing values of σ. Note that when $N = D$ one has

$$D(\sigma) = \left(\frac{1 + \sigma}{1 + 2\sigma} \right) D$$

and therefore

$$\begin{aligned} i_+(H + \sigma M + J^{\mathrm{T}} D(\sigma)^{-1} J) &= i_+ \left(H + \sigma M + J^{\mathrm{T}} D^{-1} J + \left(\frac{\sigma}{1 + \sigma} \right) J^{\mathrm{T}} D^{-1} J \right) \\ &\geq i_+(H + \sigma M + J^{\mathrm{T}} D^{-1} J) \end{aligned}$$

Thus, when σ is increased a difference of

$$\frac{\sigma + s}{1 + \sigma + s} J^{\mathrm{T}} D^{-1} J$$

results in the portion responsible for positive eigenvalues of $K(\sigma)$. This seems to indicate a smaller value of σ could be used to achieve the correct inertia, but also that it may be easier to "overshoot".

Chapter 5

Dynamic Convexification

5.1 Dynamic Convexification of a QP in Standard Form

Suppose we have a quadratic program in standard form

$$\begin{aligned} \underset{x\in\mathbb{R}^n}{\text{minimize}} \quad & \varphi(x) = g_k^{\mathrm{T}}(x - x_k) + \tfrac{1}{2}(x - x_k)^{\mathrm{T}} H_k (x - x_k) \\ \text{subject to} \quad & c_k + J_k(x - x_k) = 0, \quad x \geq 0, \end{aligned} \tag{5.1}$$

where g_k, H_k, c_k, J_k, and x_k are constants of appropriate dimension. This could, for example, represent a QP subproblem based at the k-th outer iteration of a SQP method, in which case x_k is the current iterate and the other constants are $g_k \equiv g(x_k)$, $H_k \equiv H(x_k, y_k)$, $J_k \equiv J(x_k)$ etc., i.e., the problem functions defined in Table 1.1 and their derivatives evaluated at (x_k, y_k). Nothing is assumed about H_k other than symmetry, hence (5.1) may be a nonconvex quadratic program.

In what follows, we will use the following notational conventions. The Lagrange multipliers

for the equality and bound constraints of the QP (5.1) will be denoted by $y \in \mathbb{R}^m$ and $z \in \mathbb{R}^n$ respectively. When convenient, the combined QP multipliers will be written $w = (y, z)$. The change in multipliers will be denoted $\Delta w_j = (q_j, r_j) = (y_{j+1} - y_j,\ z_{j+1} - z_j)$, or equivalently, $\Delta w_j = w_{j+1} - w_j$. Note that the active entries of w and Δw will always obey the slight abuse of notation

$$w_{\mathcal{A}} = \begin{pmatrix} y \\ z_{\mathcal{A}} \end{pmatrix} \qquad \text{and} \qquad \Delta w_{\mathcal{A}} = \begin{pmatrix} q \\ r_{\mathcal{A}} \end{pmatrix},$$

because the equality constraints are always active. For primal-dual problems in which y is a variable, the equality constraint multipliers will be denoted by π.

5.1.1 Non-binding active-set methods in standard form

A non-binding active-set method for quadratic programming is closely related to the simplex method for linear programming in the sense that the properties of Farkas' lemma are used to compute a sequence of special iterates. Farkas' lemma states that if x_k is not optimal then there exists a direction p emanating from x_k that is a feasible descent direction. In the context of linear programming, the special iterates are vertices, while in quadratic programming they are subspace minimizers.

The active-set methods introduced in Chapter 2 consist of two phases. The first phase, known as the *feasibility phase*, ignores the QP objective function while attempting to drive constraint violations to zero. If successful, the feasibility phase produces a feasible starting point x_0 along with a corresponding linearly independent subset of the active set known as the *working set*. The second phase, known as the *optimality phase*, takes the feasible x_0 and the working set as inputs, and retains feasibility while minimizing φ. During the optimality phase, the iterates have a special structure that will be useful to understand. A subsequence of the QP iterates are *standard subspace*

minimizers. Between any two of them there is a sequence of nonstandard subspace minimizers. If a non-optimal multiplier is found at a subspace minimizer, the corresponding variable is freed from it's bound and that constraint becomes inactive. However, the constraint is shifted implicitly so that it remains in the working set until its associated multiplier becomes zero. This sequence of points where the working set contains an inactive constraint constitutes a sequence of *nonstandard iterates*. Once the multiplier is driven to zero, the constraint is removed from the working set and the new point is necessarily a new standard subspace minimizer. Once a subspace minimizer with no non-optimal multipliers are found, the QP optimality conditions are satisfied.

To each active-free index partition there corresponds a permutation matrix $P = \begin{pmatrix} P_{\mathcal{F}} & P_{\mathcal{A}} \end{pmatrix}$ where the columns of $P_{\mathcal{F}}$ are unit vectors e_i for $i \in \mathcal{F}$, and an analogous definition holds for $P_{\mathcal{A}}$. It then holds that $P_{\mathcal{F}}^{\mathrm{T}} P_{\mathcal{F}} = I_{n_{\mathcal{F}}}$, $P_{\mathcal{A}}^{\mathrm{T}} P_{\mathcal{A}} = I_{n_{\mathcal{A}}}$, $P_{\mathcal{A}}^{\mathrm{T}} P_{\mathcal{F}} = 0$, and $P_{\mathcal{F}} P_{\mathcal{F}}^{\mathrm{T}} + P_{\mathcal{A}} P_{\mathcal{A}}^{\mathrm{T}} = I_n$, where $n_{\mathcal{F}} + n_{\mathcal{A}} = n$. During the optimality phase, the active set of constraints will include all of the equality constraints and some, possibly empty, subset of the simple bounds. This means the active constraint matrix has the form

$$G_{\mathcal{A}} = \begin{pmatrix} J_k \\ P_{\mathcal{A}}^{\mathrm{T}} \end{pmatrix}. \tag{5.2}$$

The concurrent convexification scheme described in Section 5.1.3 is concerned with the part of an active-set algorithm where a subspace stationary point has been found with a non-optimal multiplier. Suppose x_j is a subspace stationary point with respect to the current active-free partition and $w_j = (y_j, z_j)$ are the relevant Lagrange multipliers. Then, by definition (2.2.1) of a subspace stationary point,

$$\nabla\varphi(x_j) = G_{\mathcal{A}}^{\mathrm{T}} w_{\mathcal{A}} = J_k^{\mathrm{T}} y_j + P_{\mathcal{A}} z_{\mathcal{A}}, \tag{5.3}$$

where $z_{\mathcal{A}} \triangleq P_{\mathcal{A}}^{\mathrm{T}} z_j = [\, z_j \,]_{\mathcal{A}}$ are the bound-constraint multipliers corresponding to the active variables.

Let $\nu_s \in \{1, \ldots, n\}$ denote the index of an inequality constraint with a non-optimal multiplier, so that $[\, z_j \,]_{\nu_s} = [\, z_{\mathcal{A}} \,]_s < 0$. The search direction p is obtained by "moving off" the constraint with the non-optimal multiplier and keeping all other constraints in the working set fixed. The optimal such direction p_j is the solution of the equality constrained quadratic program

$$\begin{aligned} \underset{p \in \mathbb{R}^n}{\text{minimize}} \quad & \varphi(x_j + p) \\ \text{subject to} \quad & G_{\mathcal{A}} p = e_{m+s}. \end{aligned} \tag{5.4}$$

Any direction feasible for (5.4) must satisfy $J_k p = 0$ and $p_{\mathcal{A}} = e_s$ and thus for any $\alpha > 0$

$$c_k + J_k(x_j + \alpha p - x_k) = 0 \qquad \text{and} \qquad P_{\mathcal{A}}^{\mathrm{T}}(x_j + \alpha p) = \alpha e_s,$$

which shows that, along p_j, the equality constraints are satisfied and the ν_s-th inequality constraint becomes inactive. This confirms that a solution of (5.4) accomplishes the objective of moving off the targeted constraint while keeping other constraints in the working set satisfied.

The first-order necessary optimality conditions for a primal-dual solution $(p_j,\ y_{j+1}, z_{j+1})$ of (5.4) are given by

$$\begin{gathered} \nabla\varphi(x_j + p_j) = \nabla\varphi(x_j) + H_k p_j = J_k^{\mathrm{T}} y_{j+1} + P_{\mathcal{A}} P_{\mathcal{A}}^{\mathrm{T}} z_{j+1}, \\ G_{\mathcal{A}} p_j = e_{m+s}. \end{gathered} \tag{5.5}$$

By using the notation $\Delta w_j = (q_j,\ r_j) = (y_{j+1} - y_j,\ z_{j+1} - z_j)$ previously described, and by using the subspace stationary point property (5.3), the first set of equations in (5.5) reduces to

$H_k p_j - G_A^{\mathrm{T}} \Delta w_A = 0$. Combining this with the second set of equations produces the system

$$\begin{pmatrix} H_k & G_A^{\mathrm{T}} \\ G_A & 0 \end{pmatrix} \begin{pmatrix} p_j \\ -\Delta w_A \end{pmatrix} = \begin{pmatrix} 0 \\ e_{m+s} \end{pmatrix}. \tag{5.6}$$

In order to reduce this to a system involving just the free variables, extend the permutation P by I_{m+n_A} and symmetrically permute the KKT system as follows

$$\begin{aligned} \begin{pmatrix} P^{\mathrm{T}} & \\ & I_{m+n_A} \end{pmatrix} \begin{pmatrix} H_k & G_A^{\mathrm{T}} \\ G_A & 0 \end{pmatrix} \begin{pmatrix} P & \\ & I_{m+n_A} \end{pmatrix} &= \begin{pmatrix} P^{\mathrm{T}} H_k P & (G_A P)^{\mathrm{T}} \\ G_A P & 0 \end{pmatrix} \\ &= \begin{pmatrix} H_F & H_D & J_F^{\mathrm{T}} & I_{n_A} \\ H_D^{\mathrm{T}} & H_A & J_A^{\mathrm{T}} & 0 \\ J_F & J_A & 0 & 0 \\ 0 & I_{n_A} & 0 & 0 \end{pmatrix}, \end{aligned} \tag{5.7}$$

where H_F and H_A are the free and fixed rows and columns of H_k, respectively, and J_F and J_A denote the free and fixed columns of J_k, respectively. The quantity $H_D = P_F^{\mathrm{T}} H_k P_A$ represents the free rows of the fixed columns of H_k. It follows that a system equivalent to (5.6) is

$$\begin{pmatrix} H_F & H_D & J_F^{\mathrm{T}} & 0 \\ H_D^{\mathrm{T}} & H_A & J_A^{\mathrm{T}} & I_{n_A} \\ J_F & J_A & 0 & 0 \\ 0 & I_{n_A} & 0 & 0 \end{pmatrix} \begin{pmatrix} p_F \\ p_A \\ -q_j \\ -r_A \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ e_s \end{pmatrix},$$

with $p_F = P_F^{\mathrm{T}} p_j$, $p_A = P_A^{\mathrm{T}} p_j$. Note that q_j and r_j are not permuted. The fourth equation block gives $p_A = e_s$, which then allows the first and third blocks of equations to be written as

$$H_F p_F - J_F^{\mathrm{T}} q_j = -H_D e_s = -P_F^{\mathrm{T}} H_k e_{\nu_s} = -[\, H_k e_{\nu_s} \,]_F \quad \text{and} \quad J_F p_F = -J_A e_s = -J_k e_{\nu_s},$$

giving the reduced, free KKT system

$$\begin{pmatrix} H_{\mathcal{F}} & J_{\mathcal{F}}^{\mathrm{T}} \\ J_{\mathcal{F}} & 0 \end{pmatrix} \begin{pmatrix} p_{\mathcal{F}} \\ -q_j \end{pmatrix} = - \begin{pmatrix} [\, H_k e_{\nu_s} \,]_{\mathcal{F}} \\ J_k e_{\nu_s} \end{pmatrix} \tag{5.8}$$

from which $r_{\mathcal{A}} = [\, H_k p_j - J_k^{\mathrm{T}} q_j \,]_{\mathcal{A}}$ can be recovered.

5.1.2 Pre-convexification

In order to start the solution of the quadratic program, a subspace minimizer must first be located. As described in Definition 2.2.1, both stationarity and minimality must hold with respect to the active set at an initial point x_k. Pre-convexification is concerned with ensuring the latter requirement is satisfied, i.e., that the QP reduced Hessian $Z_{\mathcal{F}}^{\mathrm{T}} H Z_{\mathcal{F}}$ is positive definite, where the columns of $Z_{\mathcal{F}}$ form a basis for the null space of $J_{\mathcal{F}}$.

In what follows, it is required that the free rows of the Jacobian are linearly independent, so that $J_{\mathcal{F}}$ has full rank. Under this assumption, the convexification of the reduced Hessian can be done indirectly by ensuring that the free KKT matrix

$$K_{\mathcal{F}} \triangleq \begin{pmatrix} H_{\mathcal{F}} & J_{\mathcal{F}}^{\mathrm{T}} \\ J_{\mathcal{F}} & 0 \end{pmatrix}$$

has inertia $\mathrm{In}(K_{\mathcal{F}}) = (n_{\mathcal{F}}, m, 0)$. If this holds it is said that $K_{\mathcal{F}}$ is "second-order consistent" or that $\mathcal{F}(x_k)$ is a second-order consistent basis. The reason this can be done indirectly is that $\mathrm{In}(K_{\mathcal{F}}) = \mathrm{In}(Z^{\mathrm{T}} H_{\mathcal{F}} Z) + (m, m, 0)$ from which it follows that $K_{\mathcal{F}}$ being second-order consistent implies $Z^{\mathrm{T}} H_{\mathcal{F}} Z$ is positive definite. It should be emphasized that this inertia equation only holds when $J_{\mathcal{F}}$ is full rank.

If necessary, iterations are performed to find a subspace stationary point while retain-

ing second-order consistency. This is done by minimizing φ while holding the active constraints constant. At each of these iterations there are two possibilities:

(i) an unconstrained unit step is taken, or

(ii) a step is taken to a blocking constraint, which is added to the active set.

The unit step of case (i) must necessarily yield a constrained stationary point. The removal of a free variable in case (ii) will reduce the dimension of Z. If this procedure is repeated, then enough constraints become active to define a vertex, which is trivially a stationary point because then there must be a subset of n independent active constraints. As there are finitely many inactive constraints this procedure must terminate at a subspace stationary point in no more than m iterations. For details, see Gill and Wong [41].

Pre-convexification by modifying the Hessian

Suppose that the given initial point x_k defines a free KKT matrix that is not second-order consistent. There are several procedures available for pre-convexification that are based on the symmetric indefinite factorization of $K_{\scriptscriptstyle F}$. These procedures each produce a positive-semidefinite perturbation Δ to H such that

$$\text{In}\begin{pmatrix} H_{\scriptscriptstyle F}+\Delta_{\scriptscriptstyle F} & J_{\scriptscriptstyle F}^{\mathrm{T}} \\ J_{\scriptscriptstyle F} & 0 \end{pmatrix} = (n_{\scriptscriptstyle F}, m, 0).$$

Three methods considered here are:

1. the inertia-controlling symmetric indefinite factorization of Forsgren [27];

2. the method of Wächter and Biegler [68]; and

3. a two-stage symmetric indefinite factorization, see Section (4.1.1).

Each of these methods is guaranteed to produce a second-order consistent modification, but each method has its practical strengths and weaknesses. For example, the inertia-controlling factorization produces a diagonal Δ_F but can lead to significant fill-in of the factors due to the restricted pivot order needed to control the inertia. The method of Wächter and Biegler results in a diagonal Δ and sparse factors but may require several factorizations. The two-stage approach uses only two sparse "off-the-shelf" factorizations, but can result in a dense perturbation Δ. Of these methods, only the method of Wächter and Biegler is suitable for large sparse problems, but the computational cost of numerous factorizations can be considerable.

Temporary artificial constraints

The technique that will now be described is fundamentally different in that it obviates the need for pre-convexification. No modification Δ is computed and the QP Hessian is unchanged. Instead, second-order consistency is achieved by temporarily fixing a collection of variables at their current values.

Let $\mathcal{X} \subset \mathcal{F}(x_k)$ denote the index set of a collection of n_x free variables that will be temporarily fixed. Also, define $\widehat{\mathcal{A}}(x_k)$ and $\widehat{\mathcal{F}}(x_k)$ to be the indices of the free and active sets after the reassignment of variables in $\mathcal{X}$ from free to active, i.e.,

$$\widehat{\mathcal{A}}(x_k) = \mathcal{A}(x_k) \cup \mathcal{X} \qquad \text{and} \qquad \widehat{\mathcal{F}}(x_k) = \mathcal{F}(x_k) \setminus \mathcal{X}.$$

Similarly, the subscripts "X", "$\widehat{A}$", and "$\widehat{F}$" will refer to the entries of a variable or matrix with indices in the corresponding set.

Temporary artificial constraints have the form $[\, x - x_k \,]_X = 0$ which are, of course, active

at x_k by design. Assuming $K_{\mathcal{F}}$ is not second-order consistent, we want to investigate the inertia of the free KKT matrix of order $n_{\mathcal{F}} - n_{\mathcal{X}} + m$ that results from fixing $[\,x\,]_{\mathcal{X}}$, i.e.,

$$K_{\hat{\mathcal{F}}} = \begin{pmatrix} H_{\hat{\mathcal{F}}} & J_{\hat{\mathcal{F}}}^{\mathrm{T}} \\ J_{\hat{\mathcal{F}}} & 0 \end{pmatrix}.$$

This matrix may be related to $K_{\mathcal{F}}$ by defining a suitable permutation. If P_1 is a permutation matrix that moves indices in $\mathcal{X}$ to the trailing position then

$$P_1^{\mathrm{T}} K_{\mathcal{F}} P_1 = \begin{pmatrix} H_{\hat{\mathcal{F}}} & H_{\mathcal{O}} & J_{\hat{\mathcal{F}}}^{\mathrm{T}} \\ H_{\mathcal{O}}^{\mathrm{T}} & H_{\mathcal{X}} & J_{\mathcal{X}}^{\mathrm{T}} \\ J_{\hat{\mathcal{F}}} & J_{\mathcal{X}} & 0 \end{pmatrix}.$$

Let P_2 be the permutation that exchanges the trailing m rows and columns with those in positions $\{n_{\mathcal{F}} - n_{\mathcal{X}} + 1, \ldots, n_{\mathcal{F}}\}$. This gives

$$P_2^{\mathrm{T}} P_1^{\mathrm{T}} K_{\mathcal{F}} P_1 P_2 = P_2^{\mathrm{T}} \begin{pmatrix} H_{\hat{\mathcal{F}}} & H_{\mathcal{O}} & J_{\hat{\mathcal{F}}}^{\mathrm{T}} \\ H_{\mathcal{O}}^{\mathrm{T}} & H_{\mathcal{X}} & J_{\mathcal{X}}^{\mathrm{T}} \\ J_{\hat{\mathcal{F}}} & J_{\mathcal{X}} & 0 \end{pmatrix} P_2 = \begin{pmatrix} H_{\hat{\mathcal{F}}} & J_{\hat{\mathcal{F}}}^{\mathrm{T}} & H_{\mathcal{O}} \\ J_{\hat{\mathcal{F}}} & 0 & J_{\mathcal{X}} \\ H_{\mathcal{O}}^{\mathrm{T}} & J_{\mathcal{X}}^{\mathrm{T}} & H_{\mathcal{X}} \end{pmatrix}.$$

Combining these permutations as $P = P_1 P_2$ gives the expression

$$P^{\mathrm{T}} K_{\mathcal{F}} P = \begin{pmatrix} H_{\hat{\mathcal{F}}} & J_{\hat{\mathcal{F}}}^{\mathrm{T}} & H_{\mathcal{O}} \\ J_{\hat{\mathcal{F}}} & 0 & J_{\mathcal{X}} \\ H_{\mathcal{O}}^{\mathrm{T}} & J_{\mathcal{X}}^{\mathrm{T}} & H_{\mathcal{X}}. \end{pmatrix} = \begin{pmatrix} K_{\hat{\mathcal{F}}} & B \\ B^{\mathrm{T}} & H_{\mathcal{X}} \end{pmatrix}, \quad \text{with} \quad B = \begin{pmatrix} H_{\mathcal{O}} \\ J_{\mathcal{X}} \end{pmatrix}.$$

If $K_{\hat{\mathcal{F}}}$ is nonsingular, the inertia of $K_{\hat{\mathcal{F}}}$ can be deduced from that of $K_{\mathcal{F}}$ using the relation

$$\begin{pmatrix} K_{\hat{\mathcal{F}}} & B \\ B^{\mathrm{T}} & H_{\mathcal{X}} \end{pmatrix} = \begin{pmatrix} I & 0 \\ B^{\mathrm{T}} K_{\hat{\mathcal{F}}}^{-1} & I \end{pmatrix} \begin{pmatrix} K_{\hat{\mathcal{F}}} & 0 \\ 0 & H_{\mathcal{X}} - B^{\mathrm{T}} K_{\hat{\mathcal{F}}}^{-1} B \end{pmatrix} \begin{pmatrix} I & K_{\hat{\mathcal{F}}}^{-1} B \\ 0 & I \end{pmatrix}.$$

The application of Sylvester's Law of Inertia gives $\text{In}(K_{\mathcal{F}}) = \text{In}(K_{\hat{\mathcal{F}}}) + \text{In}(H_{\mathcal{X}} - B^{\mathrm{T}} K_{\hat{\mathcal{F}}}^{-1} B)$.

If the initial free KKT matrix is not second-order consistent, then its inertia can be written as $\text{In}(K_{\mathcal{F}}) = (n_{\mathcal{F}} - s,\ m + s, 0)$ for some positive integer s. The goal is to accumulate s negative eigenvalues in the Schur complement, so that

$$\begin{aligned}
\text{In}(K_{\hat{\mathcal{F}}}) &= \text{In}(K_{\mathcal{F}}) - \text{In}(H_{\mathcal{X}} - B^{\mathrm{T}} K_{\hat{\mathcal{F}}}^{-1} B) \\
&= (n_{\mathcal{F}} - s, m + s, 0) - (n_{\mathcal{X}} - s, s, 0) \\
&= (n_{\mathcal{F}} - n_{\mathcal{X}}, m, 0),
\end{aligned}$$

which is the correct inertia.

For this general case it is necessary to assume that the normal e_i^{T} of the artificial constraint is linearly independent of the rows of $G_{\mathcal{A}}$ for each $i \in \mathcal{X}$. This ensures that fixing x_i increases the rank of $G_{\mathcal{A}}$, thereby decreasing $\dim\big(\text{null}(G_{\mathcal{A}})\big)$. This ensures that if enough temporary constraints are added then x_k will become a non-degenerate vertex, which is trivially a subspace minimizer. As the starting dimension of $\text{null}(G_{\mathcal{A}})$ is finite, this process is guaranteed to terminate at a subspace minimizer, with the "worst case" scenario being that a temporary vertex must be defined. Once the correct inertia is observed, the quadratic program can be solved. All the multipliers corresponding to temporary constraints will be regarded as non-optimal, regardless of sign. Once an artificial multiplier has been driven to optimality, the artificial constraint is permanently released.

Note that two linear independence assumptions were required in this section; first that $J_{\mathcal{F}}$ has linearly independent rows, and second, that the rows of $G_{\mathcal{A}}$ are independent from those of the temporary artificial constraint Jacobian. It will be shown in Sections 5.2.2 and 5.3.1 that the constraint regularization employed by primal-dual SQP methods guarantees that both these

assumptions hold automatically.

Recursive inertia calculation

Rather than working with a Schur complement that increases in size each time a variable is artificially fixed, we will now present a way to compute the needed inertia recursively that involves only scalar complements. The real benefit of this approach is that it avoids having to compute the inertia of the $n_X \times n_X$ Schur complement matrix $H_X - B^{\mathrm{T}} K_{\hat{F}}^{-1} B$, which may be dense and increases in size with the temporarily fixed index set $\mathcal{X}$.

To illustrate the recursive relationship we shall use a subscript N to indicate the N-th state of this process. So, for example, $\mathcal{F}_N$ contains the indices of the free variables after N of them have been temporarily fixed. If we begin by fixing a single variable, we have

$$P^{\mathrm{T}} K_{F_0} P = \begin{pmatrix} H_{F_1} & J_{F_1}^{\mathrm{T}} & H_{D_1} \\ J_{F_1} & 0 & J_{X_1} \\ H_{D_1}^{\mathrm{T}} & J_{X_1}^{\mathrm{T}} & H_{X_1} \end{pmatrix} = \begin{pmatrix} K_{F_1} & B_1 \\ B_1^{\mathrm{T}} & H_{X_1} \end{pmatrix},$$

where K_{F_1} is bordered by a single column. The Schur complement of K_{F_1} is the scalar $S_1 = H_{X_1} - B_1^{\mathrm{T}} K_{F_1}^{-1} B_1$. As we have seen, we can write the inertia equation $\mathrm{In}(K_{F_0}) = \mathrm{In}(K_{F_1}) + \mathrm{In}(S_1)$. Note that the inertia of K_{F_1} is independent of the border enclosing it, and can be computed in the same way. In so doing, we get $\mathrm{In}(K_{F_1}) = \mathrm{In}(K_{F_2}) + \mathrm{In}(S_2)$, where S_2 is the scalar Schur complement $S_2 = H_{X_2} - B_2^{\mathrm{T}} K_{F_2}^{-1} B_2$. At the N-th iteration of this process we have

$$\mathrm{In}(K_{F_N}) = \mathrm{In}(K_F) - \sum_{i=1}^{N} \mathrm{In}(S_i).$$

The principal work required to calculate these inertia values is in computing each Schur

complement, which requires the solution of a system of the form $K_{F_N} u = B_N$. The solution is then used to compute $S_N = H_{X_N} - B_N^{\mathrm{T}} u$. At the start of the pre-convexification phase, an initial symmetric indefinite factorization of K_F is computed such that $\Pi^{\mathrm{T}} K_F \Pi = L D L^{\mathrm{T}}$. This factorization can be used to solve these systems efficiently, with only slight modification needed. Note that at each stage of pre-convexification, there exists a permutation matrix P such that K_{F_N} constitutes the first n_{F_N} rows and columns of $P^{\mathrm{T}} K_F P$, or simply $E^{\mathrm{T}} P^{\mathrm{T}} K_F P E$, where E is the first n_{F_N} columns of identity. Combining this fact with the symmetric indefinite factorization implies that

$$K_{F_N} = E^{\mathrm{T}} P^{\mathrm{T}} \Pi L D L^{\mathrm{T}} \Pi^{\mathrm{T}} P E.$$

This expression involves matrices that are permutations of columns of the identity matrix, so to simplify it, let $\bar{L}$ be the first n_{F_N} rows of the permuted rows of L, i.e., $P^{\mathrm{T}} \Pi L$, then we have the simple form $K_{F_N} = \bar{L} D \bar{L}^{\mathrm{T}}$. The solution of $K_{F_N} u = B_N$ is computed from

$$\bar{L} \xi = B_N$$

$$\bar{L}^{\mathrm{T}} u = D^{-1} \xi.$$

5.1.3 Concurrent convexification

In this section we assume without loss of generality that no pre-convexification is needed so that $\Delta = 0$ and $\nabla^2 \varphi = H_k$, since the following theory extends directly to the case $\nabla^2 \varphi = H_k + \Delta$ with nonzero Δ. Suppose that the equations (5.8) are solved for the free components of the primal-dual search direction, which facilitates the reconstruction of the full direction vector (p_j, q_j, r_j).

By design, p_j is a descent direction for the QP because

$$\nabla\varphi(x_j)^{\mathrm{T}} p_j = (G_{\mathcal{A}}^{\mathrm{T}} w_{\mathcal{A}})^{\mathrm{T}} p_j = w_{\mathcal{A}}^{\mathrm{T}} (G_{\mathcal{A}} p_j) = \begin{pmatrix} y_j & z_{\mathcal{A}} \end{pmatrix}^{\mathrm{T}} e_{m+s} = [\, z_{\mathcal{A}} \,]_s < 0.$$

If it happens that p_j is not a direction positive curvature for φ then

$$p_j^{\mathrm{T}} \nabla^2\varphi\, p_j = p_j^{\mathrm{T}} H_k p_j = p_j^{\mathrm{T}} (G_{\mathcal{A}}^{\mathrm{T}} \Delta w_{\mathcal{A}}) = e_{m+s}^{\mathrm{T}} \Delta w_{\mathcal{A}} = e_s^{\mathrm{T}} r_{\mathcal{A}} = [\, r_{\mathcal{A}} \,]_s \leq 0.$$

In this case, φ is unbounded below because

$$\nabla\varphi(x_j + \alpha p_j)^{\mathrm{T}} p_j = \nabla\varphi(x_j)^{\mathrm{T}} p_j + \alpha p_j^{\mathrm{T}} H_k p_j = [\, z_{\mathcal{A}} \,]_s + \alpha [\, r_{\mathcal{A}} \,]_s < 0$$

for all $\alpha > 0$.

To correct the curvature, some value of σ yet to be determined will be used to define the modified Hessian $H_k(\sigma) = H_k + \sigma e_{\nu_s} e_{\nu_s}^{\mathrm{T}}$. The reasoning for this choice is that the working set has the form (5.2), so the s-th inequality constraint normal is $(m+s)$-th row of the working set

$$G_{\mathcal{A}}^{\mathrm{T}} e_{m+s} = \begin{pmatrix} J_k^{\mathrm{T}} & P_{\mathcal{A}} \end{pmatrix} \begin{pmatrix} 0 \\ e_s \end{pmatrix} = P_{\mathcal{A}} e_s = e_{\nu_s}.$$

With this choice, the corresponding multipliers can be deduced such that $\nabla\varphi(x_j) = J_k^{\mathrm{T}} y_j + P_{\mathcal{A}} z_{\mathcal{A}}(\sigma)$,

so that x_j remains a subspace stationary point with the same y-values and modified reduced costs.

$$\begin{aligned} g_k + \big(H_k + \sigma e_{\nu_s} e_{\nu_s}^{\mathrm{T}}\big)(x_j - x_k) &= g_k + H_k(x_j - x_k) + \sigma[\, x_j - x_k \,]_{\nu_s} e_{\nu_s} \\ &= \nabla\varphi(x_j) - \sigma[\, x_k \,]_{\nu_s} e_{\nu_s} \\ &= J_k^{\mathrm{T}} y_j + P_{\!A} z_{\!A} - \sigma[\, x_k \,]_{\nu_s} P_{\!A} e_s \\ &= J_k^{\mathrm{T}} y_j + P_{\!A}(z_{\!A} - \sigma[\, x_k \,]_{\nu_s} e_s). \end{aligned}$$

It follows that $z_{\!A}(\sigma) = z_{\!A} - \sigma[x_k]_{\nu_s} e_s$ is the needed adjustment to the reduced costs. Notice that only the s-th element of $z_{\!A}$ (or ν_s-th element of z) requires correction.

The solution of (5.6) must also reflect the change. As H_k only appears in the first block of equations, the necessary change can be computed using the fact that $G_{\!A}^{\mathrm{T}} e_{m+s} = e_{\nu_s}$ as follows,

$$0 = H_k p_j + \sigma e_{\nu_s} e_{\nu_s}^{\mathrm{T}} p_j - G_{\!A}^{\mathrm{T}} \Delta w_{\!A}(\sigma) \tag{5.9}$$

$$= H_k p_j - G_{\!A}^{\mathrm{T}} \Delta w_{\!A}(\sigma) + \sigma G_{\!A}^{\mathrm{T}} e_{m+s}(e_s^{\mathrm{T}} p_{\!A}) \tag{5.10}$$

$$= H_k p_j - G_{\!A}^{\mathrm{T}} \big(\Delta w_{\!A}(\sigma) - \sigma e_{m+s}\big). \tag{5.11}$$

It follows that $\Delta w_{\!A} = \Delta w_{\!A}(\sigma) - \sigma e_{m+s}$ and therefore $\Delta w_{\!A}(\sigma) = \Delta w_{\!A} + \sigma e_{m+s}$. In terms of the separate change in multipliers we have $q_j(\sigma) = q_j$ and $r_{\!A}(\sigma) = r_{\!A} + \sigma e_s$.

The optimal step-length of the modified QP is

$$\alpha^*(\sigma) = -\frac{[\, z_{\!A}(\sigma) \,]_s}{[\, \Delta w_{\!A}(\sigma) \,]_s} = -\frac{[\, z_{\!A} \,]_s - \sigma[\, x_k \,]_{\nu_s}}{[\, \Delta w_{\!A} \,]_s + \sigma}. \tag{5.12}$$

The value of σ can then be chosen so that the resulting curvature $p_j^{\mathrm{T}} H_k(\sigma) p_j$ is sufficiently positive.

It is shown by Gill and Wong in [42] that the curvature $p_j^{\mathrm{T}} H_k p_j$ is non-decreasing during

each sequence of nonstandard iterates. This implies that concurrent convexification need only take place, if at all, at the beginning of each collection of iterates associated with a non-optimal multiplier. For consistency, we define $\sigma = 0$ for each direction along which the curvature was already sufficiently positive and no correction was made.

Let (s_1, σ_1), (s_2, σ_2), ... denote the indices of the selected non-optimal multipliers and the resulting value of σ determined by concurrent convexification. The rank-one corrective curvature matrices $\sigma_i e_{\nu_{s_i}} e_{\nu_{s_i}}^{\mathrm{T}}$ can be accumulated to form a positive semidefinite diagonal matrix

$$\Sigma = \sum_i \sigma_i e_{\nu_{s_i}} e_{\nu_{s_i}}^{\mathrm{T}} \tag{5.13}$$

Though the modification to H_k is implicit, the resulting sequence of iterates and computed quantities are identical to those produced by solving the "convexified" QP

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad g_k^{\mathrm{T}}(x - x_k) + \tfrac{1}{2}(x - x_k)^{\mathrm{T}}(H_k + \Sigma)(x - x_k). \tag{5.14}$$

Note that this does not mean that $H_k + \Sigma$ is necessarily positive definite or that the QP (5.14) is bounded. It does mean that the method finds a bounded solution of the modified subproblem.

5.1.4 Post-convexification for constraints in standard form

A standard form active-set method can be regarded as an all-inequality form method in which the general constraints are always active. This entails working with the constraints

$$\widetilde{c}(x) = \begin{pmatrix} c(x) \\ x \end{pmatrix} \geq 0, \quad \text{with} \quad \widetilde{J}(x) = \begin{pmatrix} J(x) \\ I_n \end{pmatrix},$$

and the corresponding Lagrangian function defined in terms of both the general and non-negativity constraints

$$L(x, y, z) = f(x) - y^{\mathrm{T}} c(x) - z^{\mathrm{T}} x = f(x) - w^{\mathrm{T}} \widetilde{c}(x).$$

The goal of a post-convexification strategy is to ensure the overall direction $p_k = \widehat{x} - x_k$ obtained by solving the QP subproblem satisfies descent direction requirements. Specifically, we must enforce that the direction produced by the QP subproblem is a descent direction for the Lagrangian evaluated with the optimal multipliers, i.e., that $\nabla L(x_k, \widehat{y}, \widehat{z})^{\mathrm{T}} p_k < 0$.

A QP subproblem that has been solved with either or both of the pre-convexification and concurrent convexification schemes will be referred to as *partially convexified*. An active-set method applied to a partially-convexified subproblem solves the closely related standard form quadratic program

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad & g_k^{\mathrm{T}}(x - x_k) + \tfrac{1}{2}(x - x_k)^{\mathrm{T}}(H_k + \Delta + \Sigma)(x - x_k) \\ \text{subject to} \quad & c_k + J_k(x - x_k) = 0, \quad x \geq 0, \end{aligned} \tag{5.15}$$

where Δ and Σ are symmetric positive semidefinite perturbations determined by partial convexification. As $\widehat{H} = H_k + \Delta + \Sigma$ may not be positive definite, there is no reason to expect the computed direction to be a descent direction for the Lagrangian function.

If $(\widehat{x},\ \widehat{w})$ are the primal-dual solution of the convexified problem (5.15), with the active-free index partition $\mathcal{A}(\widehat{x})$ and $\mathcal{F}(\widehat{x})$, then they must satisfy the second-order-consistent system

$$\begin{pmatrix} \widehat{H} & G_{\mathcal{A}}^{\mathrm{T}} \\ G_{\mathcal{A}} & 0 \end{pmatrix} \begin{pmatrix} p_k \\ -\widehat{w}_{\mathcal{A}} \end{pmatrix} = - \begin{pmatrix} g_k \\ \widetilde{c}_{\mathcal{A}} \end{pmatrix}. \tag{5.16}$$

Recalling that the working set matrix has the form (5.2), this is equivalent to

$$\begin{pmatrix} \widehat{H} & J_k^{\mathrm{T}} & P_{A} \\ J_k & 0 & 0 \\ P_{A}^{\mathrm{T}} & 0 & 0 \end{pmatrix} \begin{pmatrix} p_k \\ -\widehat{y} \\ -\widehat{z}_{A} \end{pmatrix} = - \begin{pmatrix} g_k \\ c_k \\ P_{A}^{\mathrm{T}} x_k \end{pmatrix}.$$

Therefore, the curvature resulting from partial convexification has the following form

$$p_k^{\mathrm{T}} \widehat{H} p_k = -p_k^{\mathrm{T}} (g_k - J_k^{\mathrm{T}} \widehat{y} - \widehat{z}) = -p_k^{\mathrm{T}} \nabla L(x_k, \widehat{y}, \widehat{z}).$$

To guarantee p_k is a descent direction for the Lagrangian function, we need only ensure that this curvature is positive.

In practice it is best to prevent the curvature from getting arbitrarily close to zero. Let $\lambda_{\min}$ be a positive preassigned scalar that controls the minimum allowable curvature. A symmetric positive semidefinite perturbation Γ is required that achieves

$$p_k^{\mathrm{T}} (\widehat{H} + \Gamma) p_k \geq \lambda_{\min} \|p_k\|^2. \tag{5.17}$$

Let λ be the potentially non-positive scalar that satisfies $p_k^{\mathrm{T}} \widehat{H} p_k = \lambda \|p_k\|^2$. If $\lambda \geq \lambda_{\min}$ then $\Gamma = 0$ satisfies (5.17), otherwise the form of the working set matrix (5.2) and the fact that $J_k p_k = -c_k$ and $P_{A}^{\mathrm{T}} p_k = -P_{A}^{\mathrm{T}} x_k$ motivates the choice of perturbation $\Gamma = \sigma G_{A}^{\mathrm{T}} G_{A}$ for which

$$p_k^{\mathrm{T}} (\widehat{H} + \Gamma) p_k = \lambda \|p_k\|^2 + \sigma(\|c_k\|^2 + \|[\, x_k \,]_{A}\|^2).$$

The minimum value of σ that satisfies (5.17) is

$$\sigma = (\lambda_{\min} - \lambda)\frac{\|p_k\|^2}{\|c_k\|^2 + \|[\,x_k\,]_{\mathcal{A}}\|^2}. \tag{5.18}$$

The suggested modification $\widehat{H}(\sigma) = \widehat{H} + \sigma G_{\mathcal{A}}^{\mathrm{T}} G_{\mathcal{A}} = \widehat{H} + \Gamma$ can be applied implicitly because

$$\begin{pmatrix} \widehat{H}(\sigma) & G_{\mathcal{A}}^{\mathrm{T}} \\ G_{\mathcal{A}} & 0 \end{pmatrix} \begin{pmatrix} p_k(\sigma) \\ -\widehat{w}_{\mathcal{A}}(\sigma) \end{pmatrix} = \begin{pmatrix} \widehat{H} & G_{\mathcal{A}}^{\mathrm{T}} \\ G_{\mathcal{A}} & 0 \end{pmatrix} \begin{pmatrix} p_k \\ -\widehat{w}_{\mathcal{A}} \end{pmatrix},$$

with $p_k(\sigma) = p_k$ and $\widehat{w}_{\mathcal{A}}(\sigma) = \widehat{w} - \sigma\widetilde{c}_{\mathcal{A}}$. In terms of the general and bound multipliers this gives $\widehat{y}(\sigma) = \widehat{y} - \sigma c_k$ and $\widehat{z}_{\mathcal{A}}(\sigma) = \widehat{z}_{\mathcal{A}} - \sigma[\,x_k\,]_{\mathcal{A}}$.

Preserving optimality

A major potential pitfall of post-convexification is that the adjustment to the nonnegativity constraint multipliers required to achieve convexification may corrupt their optimality. A primal-dual solution of the partially convexified QP subproblem (5.15) must satisfy necessary optimality conditions

$$\begin{aligned} g_k + \widehat{H}p_k &= J_k^{\mathrm{T}}\widehat{y} + \widehat{z}, \\ c_k + J_k p_k &= 0, \qquad\qquad \widehat{x} \geq 0, \\ \widehat{x} \cdot \widehat{z} &= 0, \qquad\qquad \widehat{z} \geq 0. \end{aligned} \tag{5.19}$$

Recall the post-convexification adjustments derived in Section 5.1.4 were $\widehat{y}(\sigma) = \widehat{y} - \sigma c_k$ and $\widehat{z}_{\mathcal{A}}(\sigma) = \widehat{z}_{\mathcal{A}} - \sigma[\,x_k\,]_{\mathcal{A}}$. There is no requirement on the sign of the equality constraint multipliers, so $\widehat{y}(\sigma) = \widehat{y} - \sigma c_k$ poses no problem. However, $\widehat{z} \geq 0$ is required in (5.19) and it is possible $\widehat{z}_{\mathcal{A}}(\sigma) = \widehat{z}_{\mathcal{A}} - \sigma[\,x_k\,]_{\mathcal{A}} < 0$. Assuming x_k was feasible, $x_k \geq 0$ so for $\sigma \geq 0$ it holds that $\widehat{z}(\sigma) \leq \widehat{z}$. If a large enough value of σ is required to achieve the convexification, the resulting multipliers may

no longer be optimal.

To remedy this, consider splitting the post-convexification into two parts corresponding to the general and bound constraints. To be precise, consider a convexification of the form

$$\widehat{H}(\Omega) = \widehat{H} + G_{A}^{\mathrm{T}} \Omega G_{A}, \tag{5.20}$$

where Ω is a positive semidefinite diagonal matrix of the form $\operatorname{diag}(\sigma_{J} I_m, \sigma_{A} I_{n_A})$. The resulting matrix has the specific form

$$\widehat{H}(\Omega) = \widehat{H} + \begin{pmatrix} J_k^{\mathrm{T}} & P_{A} \end{pmatrix} \begin{pmatrix} \sigma_{J} I_m & \\ & \sigma_{A} I_{n_A} \end{pmatrix} \begin{pmatrix} J_k \\ P_{A}^{\mathrm{T}} \end{pmatrix} = \widehat{H} + \sigma_{J} J_k^{\mathrm{T}} J_k + \sigma_{A} P_{A} P_{A}^{\mathrm{T}}.$$

The goal is to determine the values of σ_{J} and σ_{A} such that $(p_k, \widehat{y}(\sigma_{J}), \widehat{z}(\sigma_{A}))$ is a primal-dual descent direction for the Lagrangian function, while enforcing that $\widehat{z}(\sigma_{A})$ retains nonnegativity. Any potential values must satisfy the optimality of the post-convexified QP

$$\begin{aligned} g_k + \widehat{H}(\Omega) p_k &= J_k^{\mathrm{T}} \widehat{y}(\sigma_{J}) + \widehat{z}(\sigma_{A}), & & \\ c_k + J_k p_k &= 0, & \widehat{x} &\geq 0, \\ \widehat{x} \cdot \widehat{z}(\sigma_{A}) &= 0, & \widehat{z}(\sigma_{A}) &\geq 0. \end{aligned} \tag{5.21}$$

Consider modifications $\widehat{y}(\sigma_{J}) = \widehat{y} - \sigma_{J} c_k$ and $\widehat{z}(\sigma_{A}) = \widehat{z} + \sigma_{A} P_{A} [\, p_k \,]_{A}$, and observe that these forms

satisfy

$$\begin{aligned} g_k + \widehat{H}(\Omega)p_k &= \left(g_k + \widehat{H}p_k\right) + \sigma_{\scriptscriptstyle J} J_k^{\mathrm{T}} J_k p_k + \sigma_{\scriptscriptstyle A} P_{\scriptscriptstyle A} P_{\scriptscriptstyle A}^{\mathrm{T}} p_k \\ &= \left(J_k^{\mathrm{T}}\widehat{y} + \widehat{z}\right) - \sigma_{\scriptscriptstyle J} J_k^{\mathrm{T}} c_k + \sigma_{\scriptscriptstyle A} P_{\scriptscriptstyle A} P_{\scriptscriptstyle A}^{\mathrm{T}} p_k \\ &= J_k^{\mathrm{T}}(\widehat{y} - \sigma_{\scriptscriptstyle J} c_k) + \widehat{z} + \sigma_{\scriptscriptstyle A} P_{\scriptscriptstyle A} P_{\scriptscriptstyle A}^{\mathrm{T}} p_k \\ &= J_k^{\mathrm{T}}\widehat{y}(\sigma_{\scriptscriptstyle J}) + \widehat{z}(\sigma_{\scriptscriptstyle A}), \end{aligned}$$

which means that p_k remains a subspace stationary point of the QP with the modified Hessian $\widehat{H}(\Omega)$ and the corresponding multipliers $\widehat{y}(\sigma_{\scriptscriptstyle J})$ and $\widehat{z}_{\scriptscriptstyle A}(\sigma_{\scriptscriptstyle A})$.

In order to ensure nonnegativity of the simple bound multipliers, it is required that $[\,\widehat{z} + \sigma_{\scriptscriptstyle A} p_k\,]_i \geq 0$ for each $i \in \mathcal{A}(\widehat{x})$. If the active set at x_k and at $\widehat{x} = x_k + p_k$ are the same, then $[\,p_k\,]_i = 0$ for $i \in \mathcal{A}(\widehat{x})$ giving $\widehat{z}(\sigma_{\scriptscriptstyle A}) = \widehat{z} \geq 0$ for any $\sigma_{\scriptscriptstyle A}$. Otherwise, define

$$\sigma_{\scriptscriptstyle A}^{\max} = \min\left\{ -\frac{\widehat{z}_i}{p_i} \;:\; i \in \mathcal{A}(\widehat{x}) \setminus \mathcal{A}(x_k) \right\}. \tag{5.22}$$

This limit on $\sigma_{\scriptscriptstyle A}$ satisfies $\widehat{z}(\sigma_{\scriptscriptstyle A}) \geq 0$ for all $\sigma \leq \sigma_{\scriptscriptstyle A}^{\max}$. The desired value of $\sigma_{\scriptscriptstyle A}$ is then given by

$$\sigma_{\scriptscriptstyle A} = \min\left\{ \sigma_{\scriptscriptstyle A}^{\max},\; (\lambda_{\min} - \lambda)\frac{\|p_k\|^2}{\|[\,p_k\,]_{\scriptscriptstyle A}\|^2} \right\}. \tag{5.23}$$

If the threshold $\sigma_{\scriptscriptstyle A}^{\max}$ is not binding, then the prescribed value of $\sigma_{\scriptscriptstyle A}$ will achieve the entire post-convexification, leaving $\sigma_{\scriptscriptstyle J} = 0$.

With the value of $\sigma_{\scriptscriptstyle A}$ now fixed, the next step is to determine $\sigma_{\scriptscriptstyle J}$ that will give a descent direction. Note that the Lagrangian with $y = \widehat{y}(\sigma_{\scriptscriptstyle J})$ and $z = \widehat{z}(\sigma_{\scriptscriptstyle A})$ is a function of x with gradient

given by

$$\nabla L(x, \widehat{y}(\sigma_{J}), \widehat{z}(\sigma_{A}))\big|_{x=x_k} = \nabla f(x_k) - J(x_k)^{\mathrm{T}}\widehat{y}(\sigma_{J}) - \widehat{z}(\sigma_{A}) = g_k - J_k^{\mathrm{T}}\widehat{y}(\sigma_{J}) - \widehat{z}(\sigma_{A}). \tag{5.24}$$

It follows directly from (5.21) and (5.24) that

$$p_k^{\mathrm{T}}\widehat{H}(\Omega)p_k = -p_k^{\mathrm{T}}\nabla L(x_k, \widehat{y}(\sigma_{J}), \widehat{z}(\sigma_{A})),$$

so to guarantee a sufficiently negative directional derivative along p_k it suffices to set

$$\sigma_{J} = \frac{p_k^{\mathrm{T}}((\lambda_{\min} - \lambda)I - \sigma_{A}P_{A}P_{A}^{\mathrm{T}})p_k}{c_k^{\mathrm{T}}c_k}.$$

Indeed, this choice produces

$$p_k^{\mathrm{T}}\nabla L(x_k, \widehat{y}(\sigma_{J}), \widehat{z}(\sigma_{A})) = -\lambda_{\min}\|p_k\|^2.$$

5.2 Dynamic Convexification of Stabilized SQP Methods

The purpose of this chapter is to develop the theory of dynamic convexification for the stabilized QP subproblem as it was introduced in Chapter 2. The methods developed here apply to conventional stabilized SQP and do not require a merit function or any other reference to primal-dual SQP.

5.2.1 The stabilized subproblem – standard form

Recall the stabilized QP subproblem, repeated here, has the following form.

$$\underset{x\in\mathbb{R}^n,\,y\in\mathbb{R}^m}{\text{minimize}} \quad g_k^{\mathrm{T}}(x-x_k)+\tfrac{1}{2}(x-x_k)^{\mathrm{T}}H_k(x-x_k)+\tfrac{1}{2}\mu\|y\|^2 \tag{5.25}$$

$$\text{subject to} \quad c_k+J_k(x-x_k)+\mu(y-y_k)=0, \quad x\geq 0. \tag{5.26}$$

Define the following quantities

$$\begin{aligned} &v=(x,\,y), && P_{\scriptscriptstyle X}^{\mathrm{T}}=\begin{pmatrix} I_n & 0_{n\times m}\end{pmatrix},\\ &\widetilde{g}=\begin{pmatrix} g_k\\ \mu y_k\end{pmatrix}, && \widetilde{H}=\begin{pmatrix} H_k & \\ & \mu I_m\end{pmatrix},\\ &\widetilde{c}=c_k, && \widetilde{J}=\begin{pmatrix} J_k & \mu I_m\end{pmatrix}, \end{aligned} \tag{5.27}$$

and note that the stabilized QP (5.25) is equivalent to

$$\begin{aligned} &\underset{v\in\mathbb{R}^{n+m}}{\text{minimize}} \quad \widetilde{\varphi}(v)=\widetilde{g}^{\mathrm{T}}(v-v_k)+\tfrac{1}{2}(v-v_k)^{\mathrm{T}}\widetilde{H}(v-v_k)\\ &\text{subject to} \quad \widetilde{c}+\widetilde{J}(v-v_k)=0, \quad P_{\scriptscriptstyle X}^{\mathrm{T}}v\geq 0, \end{aligned} \tag{5.28}$$

which has the same form as the conventional standard-form QP subproblem (5.1), except that the simple bounds apply only to the original primal variables. The working-set matrix will be of dimension $(m+n_{\scriptscriptstyle A})\times(m+n)$ and have the form

$$\widetilde{G}_{\scriptscriptstyle A}=\begin{pmatrix}\widetilde{J}\\ P_{\scriptscriptstyle A}^{\mathrm{T}}P_{\scriptscriptstyle X}^{\mathrm{T}}\end{pmatrix}=\begin{pmatrix} J_k & \mu I_m\\ P_{\scriptscriptstyle A}^{\mathrm{T}} & 0_{n_{\scriptscriptstyle A}\times m}\end{pmatrix}. \tag{5.29}$$

5.2.2 Pre-convexification and regularization

As described in Section 5.1.2, a subspace minimizer must be located prior to solving the QP subproblem, and the first step in doing so is to convexify the reduced Hessian associated with the free variables. If the columns of a matrix Z are to be a basis for the null space of $[\,\widetilde{G}_{A}\,]_{F}$, then the following must hold

$$[\,\widetilde{G}_{A}\,]_{F}Z = \begin{pmatrix} J_{F} & \mu I_{m} \\ 0_{n_{A}\times n_{F}} & 0_{n_{A}\times m} \end{pmatrix} \begin{pmatrix} Z_{x} \\ Z_{y} \end{pmatrix} = 0.$$

The most natural such basis matrix is to take $Z_{x} = I_{n_{F}}$ and $Z_{y} = -J_{F}/\mu$, which gives the reduced Hessian for the standard form stabilized QP (5.28)

$$Z^{\mathrm{T}}\widetilde{H}_{F}Z = \begin{pmatrix} I_{n_{F}} & -\frac{1}{\mu}J_{F}^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} H_{F} & 0 \\ 0 & \mu I_{m} \end{pmatrix} \begin{pmatrix} I_{n_{F}} \\ -\frac{1}{\mu}J_{F} \end{pmatrix} = H_{F} + \frac{1}{\mu}J_{F}^{\mathrm{T}}J_{F},$$

where $\widetilde{H}_{F}$ is the free rows and columns of $\widetilde{H}$, which is defined in (5.27). The inertia relationship (4.1) implies that the reduced Hessian $Z^{\mathrm{T}}\widetilde{H}_{F}Z$ is positive definite if and only if

$$\mathrm{In}(K_{F}) = (n_{F},\ m,\ 0), \qquad \text{with} \qquad K_{F} \triangleq \begin{pmatrix} H_{F} & J_{F}^{\mathrm{T}} \\ J_{F} & -\mu I_{m} \end{pmatrix}.$$

This suggests that the reduced Hessian be made positive definite implicitly by modifying K_{F} to be second-order consistent. Note that the key identity used here, $\mathrm{In}(K_{F}) = \mathrm{In}(H_{F} + J_{F}^{\mathrm{T}}J_{F}/\mu) + (0,\ m,\ 0)$, is true for any J_{F} of appropriate dimension, regardless of rank. This is a crucial property of regularization, which stands in contrast to the generic standard form case, in which it was required to assume the rows of J_{F} were linearly independent. This is because without regularization the indirect link with the free KKT matrix relied on the equation $\mathrm{In}(K_{F}) = \mathrm{In}(Z^{\mathrm{T}}H_{k}Z) + (m,\ m,\ 0)$

which is only true for $J_{_F}$ with full row rank.

Pre-convexification methods

Each of the techniques described in Section 5.1.2 for determining a second-order consistent basis by Hessian modification extend directly to the regularized case. That is, they each compute a symmetric positive semidefinite perturbation Δ such that

$$\mathrm{In}\begin{pmatrix} H_{_F} + \Delta_{_F} & J_{_F}^{\mathrm{T}} \\ J_{_F} & -\mu I_m \end{pmatrix} = (n_{_F},\ m,\ 0),$$

and the relative strengths and weakness described there hold true here as well.

The alternative approach of imposing temporary constraints can also be extended for use in stabilized SQP methods. Using the notation of Section 5.1.2, if a collection of free variables indexed by a set $\mathcal{X}$ are temporarily fixed at their current values then $K_{_F}$ can be permuted such that

$$P^{\mathrm{T}} K_{_F} P = \begin{pmatrix} H_{_{\hat{F}}} & J_{_{\hat{F}}}^{\mathrm{T}} & H_{_O} \\ J_{_{\hat{F}}} & -\mu I_m & J_{_X} \\ H_{_O}^{\mathrm{T}} & J_{_X}^{\mathrm{T}} & H_{_X} \end{pmatrix} = \begin{pmatrix} K_{_{\hat{F}}} & B \\ B^{\mathrm{T}} & H_{_X} \end{pmatrix}.$$

Assuming $K_{_{\hat{F}}}$ is nonsingular, the inertia of $K_{_{\hat{F}}}$ can be deduced from $\mathrm{In}(K_{_F}) = \mathrm{In}(K_{_{\hat{F}}}) + \mathrm{In}(H_{_X} - B^{\mathrm{T}} K_{_{\hat{F}}}^{-1} B)$. Note also that the same recursive inertia calculation

$$\mathrm{In}(K_{_{F_N}}) = \mathrm{In}(K_{_F}) - \sum_{i=1}^{N} \mathrm{In}(S_i), \quad \text{where} \quad K_{_{F_N}} = \begin{pmatrix} H_{_{F_N}} & J_{_{F_N}}^{\mathrm{T}} \\ J_{_{F_N}} & -\mu I_m \end{pmatrix},$$

and where $S_i = H_{_{X_i}} - B_i^{\mathrm{T}} K_{_{F_i}}^{-1} B_i$, can be used to determine which indices to fix. In the unregularized scenario, it was necessary to assume that the constraint normal e_i^{T} of each temporarily fixed variable

x_i was linearly independent from the rows of G_{A}. Due to the regularization and the special properties of bound constraints, for each $i \in \mathcal{X} \subset \mathcal{F}(x_k)$ it necessarily holds that the constraint normal e_i^{T} is linearly independent the rows of $\widetilde{G}_{A}$. It is therefore guaranteed that temporarily fixing a free variable will increase the rank of $\widetilde{G}_{A}$ thereby decreasing the dimension of the null space. This algorithm therefore must produce a final temporarily fixed index set $\mathcal{X}$ such that

$$\operatorname{In} \begin{pmatrix} H_{\hat{F}} & J_{\hat{F}}^{\mathrm{T}} \\ J_{\hat{F}} & -\mu I_m \end{pmatrix} = (n_{F} - n_{X}, m, 0),$$

which shows that $\widehat{\mathcal{F}}(x_k) = \mathcal{F}(x_k) \setminus \mathcal{X}$ is a second-order consistent basis suitable for initializing the stabilized QP subproblem.

Updating the temporary constraints

If the temporary constraint index set from the previous iteration is used to initialize $\mathcal{X}_0$ for the current iteration, it can then be expanded or contracted as needed, and the effect on the inertia of doing so can be computed using an extension of the Schur complement method of Section 5.1.2. Let $\mathcal{F}_0 = \mathcal{F} \setminus \mathcal{X}_0$ and suppose K_{F_0} is second-order consistent. It may be the case that releasing one or more of the indices in $\mathcal{X}_0$ will still yield a second-order consistent basis, while reducing the number of artificial constraints. Suppose $i \in \mathcal{X}_0$ is the candidate index considered for being freed, and define $H_{O} = P_{F}^{\mathrm{T}} H e_i$ and $J_i = J e_i$ so that

$$K_{F_1} = \begin{pmatrix} H_{ii} & H_{O}^{\mathrm{T}} & J_i^{\mathrm{T}} \\ H_{O} & H_{F} & J_{F}^{\mathrm{T}} \\ J_i & J_{F} & \mu I \end{pmatrix} \triangleq \begin{pmatrix} c & b^{\mathrm{T}} \\ b & K_{\hat{F}} \end{pmatrix} \sim \begin{pmatrix} K_{\hat{F}} & \\ & c - b^{\mathrm{T}} K_{\hat{F}}^{-1} b \end{pmatrix}.$$

It follows that if $c - b^{\mathrm{T}} K_{\hat{F}}^{-1} b > 0$ then $\mathrm{In}(K_{F_1}) = (n_{F_0} + 1, m, 0)$ and that temporary constraint can be released, giving $\mathcal{X}_1 = \mathcal{X}_0 \setminus \{i\}$ and $\mathcal{F}_1 = \mathcal{F}_0 \cup \{i\}$. This process can be repeated either for a pre-determined number of times, or until each temporarily fixed index has been considered for removal.

If the fixed set from the previous iteration defines K_{F_0} that is not second-order consistent, it can be added to using the recursive inertia calculation technique described in Section 5.1.2 rather than starting from $\mathcal{X}_0 = \emptyset$.

5.2.3 Concurrent convexification of a stabilized QP subproblem

Suppose $(v_j,\, w_j)$ is a subspace stationary point with respect to a working set of active constraints, where $v_j = (x_j,\ y_j)$ and $w_A = (\pi_j,\ z_A)$ are the multipliers π_j for equality constraints and $z_A \equiv [\, z_j \,]_A$ for the active bound constraints. The primal-dual direction will be obtained by solving the QP

$$\begin{aligned} \underset{d \in \mathbb{R}^{n+m}}{\text{minimize}} \quad & \widetilde{\varphi}(v_j + d) \\ \text{subject to} \quad & \widetilde{G}_A d = e_{m+s}. \end{aligned} \tag{5.30}$$

The optimality conditions for (5.30) are

$$\nabla\widetilde{\varphi}(v_j + d_j) = \nabla\widetilde{\varphi}(v_j) + \widetilde{H} d_j = \widetilde{G}_A^{\mathrm{T}} \widehat{w}_A = \widetilde{G}_A^{\mathrm{T}} (w_A + \Delta w_A) \tag{5.31}$$

$$\widetilde{G}_A d_j = e_{m+s}. \tag{5.32}$$

The subspace stationarity property gives $\nabla\widetilde{\varphi}(v_j) = \widetilde{G}_A^{\mathrm{T}} w_A$ therefore

$$\begin{pmatrix} \widetilde{H} & \widetilde{G}_A^{\mathrm{T}} \\ \widetilde{G}_A & 0 \end{pmatrix} \begin{pmatrix} d_j \\ -\Delta w_A \end{pmatrix} = \begin{pmatrix} 0 \\ e_{m+s} \end{pmatrix}.$$

If the permutation matrix P that defines the active-free partition of the original primal variables is expanded to $\widetilde{P} = \mathrm{diag}(P,\, I_m)$ then

$$\widetilde{G}_{A}\widetilde{P} = \begin{pmatrix} J_{F} & J_{A} & \mu I_m \\ 0_{n_A \times n_F} & I_{n_A} & 0_{n_A \times m} \end{pmatrix} \quad \text{and} \quad \widetilde{P}^{\mathrm{T}}\widetilde{H}\widetilde{P} = \begin{pmatrix} H_{F} & H_{D} & \\ H_{D}^{\mathrm{T}} & H_{A} & \\ & & \mu I_m \end{pmatrix},$$

where quantities with the subscripts J_{F}, J_{A}, H_{F}, H_{D}, and H_{A} are defined as in (5.7). Applying the expanded permutation symmetrically to the KKT system yields

$$\begin{pmatrix} H_{F} & H_{D} & 0 & J_{F}^{\mathrm{T}} & 0 \\ H_{D}^{\mathrm{T}} & H_{A} & 0 & J_{A}^{\mathrm{T}} & I \\ 0 & 0 & \mu I & \mu I & 0 \\ J_{F} & J_{A} & \mu I & 0 & 0 \\ 0 & I & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} p_{F} \\ p_{A} \\ q_j \\ -\Delta w_{F} \\ -\Delta w_{A} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ e_s \end{pmatrix}. \tag{5.33}$$

The last block of equations gives $p_{A} = e_s$ and so this reduces to the free variable stabilized system

$$\begin{pmatrix} H_{F} & J_{F}^{\mathrm{T}} \\ J_{F} & -\mu I_m \end{pmatrix} \begin{pmatrix} p_{F} \\ -\Delta w_{F} \end{pmatrix} = - \begin{pmatrix} [\, H_{\nu_s} \,]_{F} \\ J_{\nu_s} \end{pmatrix}, \tag{5.34}$$

from which one can recover $\Delta w_{A} = [\, H_k p_j - J_k^{\mathrm{T}} \Delta w_{F} \,]_{A}$.

Suppose at a subspace stationary point $(v_j,\, w_j)$ with a non-optimal multiplier $[\, z_{A} \,]_s < 0$, that the stabilized, reduced free-variable system (5.34) is solved and the full $(d_j,\, \Delta w_j)$ are reconstructed from the solution. The directional derivative and curvature of $\widetilde{\varphi}$ along d_j are given by

$$\nabla\widetilde{\varphi}(v_j)^{\mathrm{T}} d_j = (\widetilde{G}_{A}^{\mathrm{T}} w_{A})^{\mathrm{T}} d_j = w_{A}^{\mathrm{T}} e_{m+s} = \begin{pmatrix} \pi_j^{\mathrm{T}} & z_{A}^{\mathrm{T}} \end{pmatrix} \begin{pmatrix} 0 \\ e_s \end{pmatrix} = [\, z_{A} \,]_s < 0,$$

and by

$$d_j^{\mathrm{T}} \widetilde{H} d_j = d_j^{\mathrm{T}} (\widetilde{G}_{A}^{\mathrm{T}} \Delta w_{A}) = e_{m+s}^{\mathrm{T}} \Delta w_{A} = [\, r_{A} \,]_s,$$

respectively. If the curvature is sufficiently positive then no modification is required. Otherwise, consider the perturbation

$$\widetilde{H}(\sigma) = \widetilde{H} + \sigma e_{\nu_s} e_{\nu_s}^{\mathrm{T}} = \begin{pmatrix} H_k(\sigma) & \\ & \mu I \end{pmatrix}.$$

It should be emphasized here that e_{ν_s} is an $(n+m)$-vector now, whereas before it was an n-vector. If $\widetilde{H}$ is perturbed as suggested, the corresponding multipliers $z_{A}(\sigma)$ must satisfy $\widetilde{g} + \widetilde{H}(\sigma)(v_j - v_k) = \widetilde{G}_{A}^{\mathrm{T}} w_{A}(\sigma)$, i.e.,

$$\begin{pmatrix} g_k + H_k(\sigma)(x_j - x_k) \\ \mu y_j \end{pmatrix} = \begin{pmatrix} J_k^{\mathrm{T}} \pi(\sigma) + P_{A} z_{A}(\sigma) \\ \mu \pi(\sigma) \end{pmatrix}. \tag{5.35}$$

The second block of equations already holds with $\pi(\sigma) = \pi_j$ because $(v_j,\, w_j)$ is a subspace stationary point and therefore $\pi_j = y_j$. The first block of equations can be rearranged to give

$$P_{A} z_{A}(\sigma) = (g_k + H_k(x_j - x_k) - J_k^{\mathrm{T}} \pi_j) + \sigma e_{\nu_s} e_{\nu_s}^{\mathrm{T}} (x_j - x_k) = P_{A} z_{A} + \sigma [\, x_j - x_k \,]_{\nu_s} P_{A} e_s,$$

which implies that $z_{A}(\sigma) = z_{A} - [\, x_k \,]_{\nu_s} e_s$ precisely as in the generic case. Similarly, just as in (5.9), the perturbed change in dual variables has the form

$$\Delta w_j(\sigma) = \Delta w_j + \sigma e_{m+\nu_s},$$

where $e_{m+\nu_s}$ is an $(m+n)$-vector. Equivalently, we could write $\Delta w_{A}(\sigma) = \Delta w_{A} + \sigma e_{m+s}$ or simply

$q(\sigma) = 0$ and $r_A(\sigma) = r_A + \sigma e_s$.

5.2.4 Stabilized post-convexification

Consider the function $\widetilde{f}(x, y) = f(x) + \frac{1}{2}\mu y^{\mathrm{T}} y$ subject to the shifted constraints $\widetilde{c}(x, y) = c(x) + \mu(y - y_k) = 0$ and $x \geq 0$. Notice that these problem functions satisfy

$$\nabla \widetilde{f}(x_k, y_k) = \begin{pmatrix} g_k \\ \mu y_k \end{pmatrix} = \widetilde{g} \tag{5.36}$$

$$\widetilde{c}(x_k, y_k) = c_k = \widetilde{c} \tag{5.37}$$

$$\nabla^2 \widetilde{f}(x_k, y_k) - \sum_{i=1}^{m} \pi_i \nabla^2 \widetilde{c}_i(x_k, y_k) = \begin{pmatrix} \nabla^2 f(x_k) - \sum_{i=1}^{m} \pi_i \nabla^2 c_i(x_k) & \\ & \mu I \end{pmatrix} = \begin{pmatrix} H_k & \\ & \mu I \end{pmatrix} = \widetilde{H}, \tag{5.38}$$

which are the quantities appearing in (5.27). If we form the Lagrangian of this problem we get

$$\widetilde{L}(x, y, \pi, z) = \widetilde{f}(x, y) - \pi^{\mathrm{T}} \widetilde{c}(x, y) - z^{\mathrm{T}} x. \tag{5.39}$$

Therefore, with $\pi = \widehat{\pi}$ and $z = \widehat{z}$ the gradient of the Lagrangian is

$$\nabla \widetilde{L}(x, y, \widehat{\pi}, \widehat{z}) = \begin{pmatrix} \nabla f(x) - J(x)^{\mathrm{T}} \widehat{\pi} - \widehat{z} \\ \mu(y - \widehat{\pi}) \end{pmatrix}.$$

This shows that the stabilized QP subproblem is equivalent to minimizing a two-norm regularization of the objective function subject to shifted constraints.

A solution $d_k = (p_k,\ q_k)$ of the partially convexified stabilized QP subproblem will be the

solution of a system analogous to (5.16), i.e.,

$$\begin{pmatrix} \widehat{H} & 0 & J_k^{\mathrm{T}} & P_{\mathcal{A}} \\ 0 & \mu I & \mu I & 0 \\ J_k & \mu I & 0 & 0 \\ P_{\mathcal{A}}^{\mathrm{T}} & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} p_k \\ q_k \\ -\widehat{\pi} \\ -\widehat{z}_{\mathcal{A}} \end{pmatrix} = - \begin{pmatrix} g_k \\ \mu y_k \\ c_k \\ [\, x_k \,]_{\mathcal{A}} \end{pmatrix},$$

where $\widehat{H} = H_k + \Delta + \Sigma$ is the result of partial convexification. Rearranging the first two blocks of equations yields

$$\begin{pmatrix} \widehat{H} & 0 \\ 0 & \mu I \end{pmatrix} \begin{pmatrix} p_k \\ q_k \end{pmatrix} = - \begin{pmatrix} g_k \\ \mu y_k \end{pmatrix} + \begin{pmatrix} J_k^{\mathrm{T}} \widehat{\pi} + \widehat{z} \\ \mu \widehat{\pi} \end{pmatrix} = -\nabla \widetilde{L}(x_k, y_k, \widehat{\pi}, \widehat{z})$$

To ensure that the primal-dual solution $d_k = \begin{pmatrix} p_k & q_k \end{pmatrix}$ is a descent direction for the regularized Lagrangian, we require $\widehat{\pi}(\sigma)$, $\widehat{z}_{\mathcal{A}}(\sigma)$, and Γ such that

$$-\nabla \widetilde{L}(x_k, y_k, \widehat{\pi}(\sigma), \widehat{z}_{\mathcal{A}}(\sigma))^{\mathrm{T}} d_k = d_k^{\mathrm{T}} (\check{H} + \Gamma) d_k \geq \lambda_{\min} \|d_k\|^2,$$

with $\check{H} = \operatorname{diag}(\widehat{H},\ \mu I)$. If a perturbation of the form $\Gamma = \sigma G_{\mathcal{A}}^{\mathrm{T}} G_{\mathcal{A}}$ is used then

$$\check{H} + \Gamma = \begin{pmatrix} \widehat{H} & \\ & \mu I \end{pmatrix} + \sigma \begin{pmatrix} J_k^{\mathrm{T}} J_k + P_{\mathcal{A}} P_{\mathcal{A}}^{\mathrm{T}} & \mu J_k^{\mathrm{T}} \\ \mu J_k & \mu^2 I \end{pmatrix}.$$

Note that $d_k^{\mathrm{T}} \Gamma d_k = \sigma \|G_{\mathcal{A}} d_k\|^2 = \|c_k\|^2 + \|[\, x_k \,]_{\mathcal{A}}\|^2$, so the value of σ that solves

$$d_k^{\mathrm{T}} (\check{H} + \sigma G_{\mathcal{A}}^{\mathrm{T}} G_{\mathcal{A}}) d_k = \lambda_{\min} \|d_k\|^2,$$

with λ defined by $d_k^{\mathrm{T}} \check{H} d_k = \lambda d_k^{\mathrm{T}} d_k$, can be written as follows

$$\sigma = (\lambda_{\min} - \lambda) \frac{\|d_k\|^2}{\|c_k\|^2 + \|[\, x_k \,]_{\mathcal{A}}\|^2}.$$

The convexification can be applied implicitly as well by modifying the resulting multipliers according to

$$\begin{pmatrix} \widehat{\pi}(\sigma) \\ \widehat{z}_{\mathcal{A}}(\sigma) \end{pmatrix} = \begin{pmatrix} \widehat{\pi} - \sigma c_k \\ \widehat{z}_{\mathcal{A}} - \sigma [\, x_k \,]_{\mathcal{A}} \end{pmatrix}$$

It should be emphasized that applying these modifications implicitly is of critical importance in the large-scale case because each of the post-convexification modifications so far discussed involve the matrix $J_k^{\mathrm{T}} J_k$, which does not retain the sparsity of J_k.

Preserving optimality in the stabilized setting

Suppose that pre-convexification and/or concurrent convexification have been performed during solution of the stabilized QP subproblem (5.25) giving $\widehat{H} = H_k + \Delta + \Sigma$ and

$$\check{H} = \begin{pmatrix} \widehat{H} & \\ & \mu I \end{pmatrix}.$$

The post-convexification proposed in the previous section had the general form

$$\widehat{w}_{\mathcal{A}}(\sigma) = \widehat{w}_{\mathcal{A}} + \sigma \widetilde{G}_{\mathcal{A}} d_k.$$

We propose the generalization of this to

$$\widehat{w}_{\mathcal{A}}(\Omega) = \widehat{w}_{\mathcal{A}} + \Omega \widetilde{G}_{\mathcal{A}} d_k,$$

where Ω is a positive semidefinite diagonal matrix. Observe that this form solves

$$\check{H}d_k - \widetilde{G}_{A}^{\mathrm{T}}\widehat{w}_{A} = (\check{H} + \widetilde{G}_{A}^{\mathrm{T}}\Omega\widetilde{G}_{A})d_k - \widetilde{G}_{A}^{\mathrm{T}}\widehat{w}_{A}(\Omega).$$

For example, letting $\Omega = \operatorname{diag}(\sigma_{J}I,\ \sigma_{A}I)$ allows for separate control of the general and bound constraint multipliers. With this choice,

$$\widehat{w}_{A}(\Omega) = \begin{pmatrix} \widehat{\pi}(\sigma_{J}) \\ \widehat{z}_{A}(\sigma_{A}) \end{pmatrix} = \begin{pmatrix} \widehat{\pi} \\ \widehat{z}_{A} \end{pmatrix} - \begin{pmatrix} \sigma_{J}c_k \\ \sigma_{A}[\,x_k\,]_{A} \end{pmatrix} = \widehat{w}_{A} + \Omega\widetilde{G}_{A}d_k.$$

It follows that if $(d_k,\ \widehat{w})$ is a solution of the partially convexified analogue of the stabilized QP subproblem (5.25), then the same d_k is a solution of the post-convexified problem with modified Hessian $\check{H}(\Omega)$ and multipliers $\widehat{\pi}(\sigma_{J})$ and $\widehat{z}(\sigma_{A})$. Moreover, $\widehat{z}(\sigma_{A}) \geq 0$ and d_k is a descent direction for the Lagrangian function formed with the modified optimal multipliers.

Theorem 5.2.1 (Stabilized Post-convexification)**.** *Let Δ and Σ be positive semidefinite perturbations resulting from pre-convexification and concurrent convexification respectively, and let $\widehat{H} = H_k + \Delta + \Sigma$. Suppose $(\widehat{v},\ \widehat{\pi},\ \widehat{z})$ is a primal-dual solution of the partially convexified stabilized QP subproblem and $\widetilde{G}_{A}$ the working set matrix of active constraints at the solution. Define the positive semidefinite diagonal matrix $\Omega = \operatorname{diag}(\sigma_{J}I,\ \sigma_{A}I)$, where*

$$\sigma_{A} = \min\left\{\sigma_{A}^{\max},\ (\lambda_{\min} - \lambda)\frac{\|p_k\|^2}{\|[\,p_k\,]_{A}\|^2}\right\} \qquad \textit{and} \qquad \sigma_{J} = \frac{d_k^{\mathrm{T}}((\lambda_{\min} - \lambda)I - \sigma_{A}P_{A}P_{A}^{\mathrm{T}})d_k}{\widetilde{c}^{\mathrm{T}}\widetilde{c}},$$

with $\lambda_{\min} > 0$ given and λ defined by $d_k^{\mathrm{T}}\check{H}d_k = \lambda\|d_k\|^2$, and with

$$\sigma_{A}^{\max} = \min\left\{-\frac{[\,\widehat{z}\,]_i}{[\,p_k\,]_i} \ :\ i \in \mathcal{A}(\widehat{x}) \setminus \mathcal{A}(x_k)\right\}.$$

Then the post-convexification defined by $\check{H}(\Omega) = \check{H} + \widetilde{G}_{A}^{\mathrm{T}} \Omega \widetilde{G}_{A}$ *has the following properties.*

1. *The post-convexification can be applied implicitly via the adjustments*

$$\widehat{\pi}(\sigma_{J}) = \widehat{\pi} - \sigma_{J} c_k \;\; \textit{and} \;\; \widehat{z}_{A}(\sigma_{A}) = \widehat{z}_{A} - \sigma_{A}[\, x_k \,]_{A}.$$

2. *The inequality constraint multipliers remain optimal:* $\widehat{z}_{A}(\sigma_{A}) \geq 0$.

3. $(\widehat{v},\ \widehat{\pi}(\sigma_{J}),\ \widehat{z}(\sigma_{A}))$ *solves the stabilized QP with modified Hessian* $\check{H}(\Omega)$

$$\begin{aligned} \underset{v \in \mathbb{R}^{n+m}}{\text{minimize}} \quad & \widetilde{\varphi}(v) = \widetilde{g}^{\mathrm{T}}(v - v_k) + \tfrac{1}{2}(v - v_k)^{\mathrm{T}} \check{H}(\Omega)(v - v_k) \\ \text{subject to} \quad & \widetilde{c} + \widetilde{J}(v - v_k) = 0, \quad P_{A}^{\mathrm{T}} v \geq 0. \end{aligned} \tag{5.40}$$

4. *The resulting curvature along* d_k *is positive. Specifically,* $d_k^{\mathrm{T}} \check{H}(\Omega) d_k = \lambda_{\min} d_k^{\mathrm{T}} d_k > 0$.

5. d_k *is a descent direction for the Lagrangian function evaluated at* $(v,\ \pi,\ z) = (v_k,\ \widehat{\pi}(\sigma_{J}),\ \widehat{z}(\sigma_{A}))$.

Proof. For (1), note that the suggest modification can be written as $\widehat{w}_{A}(\sigma) = \widehat{w}_{A} + \Omega \widetilde{G}_{A} d_k$, therefore

$$\check{H}(\Omega) d_k - \widetilde{G}_{A}^{\mathrm{T}} \widehat{w}_{A}(\Omega) = (\check{H} + \widetilde{G}_{A}^{\mathrm{T}} \Omega \widetilde{G}_{A}) d_k - \widetilde{G}_{A}^{\mathrm{T}} (\widehat{w}_{A} + \Omega \widetilde{G}_{A} d_k) = \check{H} d_k - \widetilde{G}_{A}^{\mathrm{T}} \widehat{w}_{A}.$$

It follows that

$$\begin{pmatrix} \check{H}(\Omega) & \widetilde{G}_{A}^{\mathrm{T}} \\ \widetilde{G}_{A} & 0 \end{pmatrix} \begin{pmatrix} d_k \\ -\widehat{w}_{A}(\Omega) \end{pmatrix} = - \begin{pmatrix} \widetilde{g} \\ \widetilde{c}_{A} \end{pmatrix}. \tag{5.41}$$

Therefore, computing $\widehat{\pi}(\sigma_{J})$ and $\widehat{z}(\sigma_{A})$ is equivalent to solving the KKT system with the post-convexified Hessian.

For (2), let $i \in \mathcal{A}(\widehat{x})$. If also $i \in \mathcal{A}(x_k)$ then $[\, p_k \,]_{A} = 0$ and $\widehat{z}(\sigma_{A}) = \widehat{z} \geq 0$. Otherwise, consider indices $i \in \mathcal{A}(\widehat{x}) \setminus \mathcal{A}(x_k)$. By definition, $\sigma_{A}^{\max} \leq \widehat{z}_i / [\, x_k \,]_i$ therefore $\widehat{z}_i \geq \sigma_{A}^{\max}$ and consequently

$[\,\widehat{z} - \sigma_{A}^{\max} x_k\,]_i \geq 0$. As $\sigma_{A} \leq \sigma_{A}^{\max}$, it follows

$$\widehat{z}_{A}(\sigma_{A}) = \widehat{z}_{A} - \sigma_{A}[\,x_k\,]_{A} \geq \widehat{z}_{A} - \sigma_{A}^{\max}[\,x_k\,]_{A} \geq 0.$$

Assertion (3) will be shown by demonstrating the optimality conditions for the post-convexified QP are satisfied. These optimality conditions are

$$\begin{aligned} \widetilde{g} + \check{H}(\Omega) d_k &= \widetilde{J}^{\mathrm{T}} \widehat{\pi}(\sigma_{J}) + \widehat{z}(\sigma_{A}), \\ \widetilde{c} + \widetilde{J} d_k &= 0, \qquad & \widehat{x} &\geq 0, \\ \widehat{x} \cdot \widehat{z}(\sigma_{A}) &= 0, \qquad & \widehat{z}(\sigma_{A}) &\geq 0. \end{aligned} \tag{5.42}$$

The stationarity and feasibility conditions are shown to hold by (5.41) and the assumption that $\widehat{v}$ is feasible. Part (2) shows $\widehat{z}(\sigma_{A}) \geq 0$, and as $\widehat{x} \cdot \widehat{z}(\sigma_{A}) = \widehat{x} \cdot \widehat{z} = 0$, the optimality conditions are met.

For assertion (4), direct computation shows

$$\begin{aligned} d_k^{\mathrm{T}} \check{H}(\Omega) d_k &= \lambda \|d_k\|^2 + d_k^{\mathrm{T}} \widetilde{G}_{A}^{\mathrm{T}} \Omega \widetilde{G}_{A} d_k \\ &= \lambda \|d_k\|^2 + \sigma_{J} \widetilde{c}^{\mathrm{T}} \widetilde{c} + \sigma_{A} x_k^{\mathrm{T}} P_{A} P_{A}^{\mathrm{T}} x_k \\ &= \lambda \|d_k\|^2 + d_k^{\mathrm{T}} ((\lambda_{\min} - \lambda) I - \sigma_{A} P_{A} P_{A}^{\mathrm{T}}) d_k + \sigma_{A} x_k^{\mathrm{T}} P_{A} P_{A}^{\mathrm{T}} x_k \\ &= \lambda \|d_k\|^2 + d_k^{\mathrm{T}} (\lambda_{\min} - \lambda) I d_k = \lambda_{\min} \|d_k\|^2. \end{aligned}$$

Lastly for (5), the Lagrangian function in question is

$$\widetilde{L}(x, y, \pi, z) = \widetilde{f}(x, y) - \pi^{\mathrm{T}} \widetilde{c}(x, y) - z^{\mathrm{T}} x.$$

Therefore, again using (5.41), with $\pi = \widehat{\pi}(\sigma_{J})$ and $z = \widehat{z}(\sigma_{A})$ the gradient of the Lagrangian can be

written

$$\nabla\widetilde{L}(v_k, \widehat{\pi}(\sigma_{\scriptscriptstyle J}), \widehat{z}(\sigma_{\scriptscriptstyle A})) = \begin{pmatrix} g_k - J_k^{\mathrm{T}}\widehat{\pi}(\sigma_{\scriptscriptstyle J}) - \widehat{z}(\sigma_{\scriptscriptstyle A}) \\ \mu(y_k - \widehat{\pi}(\sigma_{\scriptscriptstyle J})) \end{pmatrix} = \widetilde{g} - \widetilde{G}_{\scriptscriptstyle A}^{\mathrm{T}}\widehat{w}(\Omega) = -\check{H}(\Omega)d_k.$$

This shows that the direction derivative along d_k is

$$\nabla\widetilde{L}(v_k, \widehat{\pi}(\sigma_{\scriptscriptstyle J}), \widehat{z}(\sigma_{\scriptscriptstyle A}))^{\mathrm{T}} d_k = -d_k^{\mathrm{T}}\check{H}(\Omega)d_k = -\lambda_{\min}\|d_k\|^2 < 0,$$

with the last equality being from part (4). It follows d_k is a descent direction. □

Relating back to stabilized SQP

It is helpful to understand how the proposed modification affects the original stabilized QP.

Theorem 5.2.2. *Define the modified quadratic objective*

$$\widetilde{\varphi}_\Omega(x, y) = g_k^{\mathrm{T}}(x - x_k) + \tfrac{1}{2}(x - x_k)^{\mathrm{T}}\widehat{H}(\Omega)(x - x_k) + \mu(y - y_k)^{\mathrm{T}}\big(y_k + \sigma_{\scriptscriptstyle J}J_k(x - x_k)\big) + \tfrac{1}{2}\mu(1 + \sigma_{\scriptscriptstyle J}\mu)\|y - y_k\|^2.$$

The post-convexified, stabilized QP

$$\begin{aligned} &\underset{x\in\mathbb{R}^n, y\in\mathbb{R}^m}{\text{minimize}} \quad \widetilde{\varphi}_\Omega(x, y) \\ &\text{subject to} \quad c_k + J_k(x - x_k) + \mu(y - y_k) = 0, \quad x \geq 0, \end{aligned} \tag{5.43}$$

has the same solution as the generic standard form post-convexified QP (5.40) *as described in* Theorem 5.2.1.

Proof. A primal dual solution $(\widehat{x},\ \widehat{y},\ \widehat{w}(\Omega))$ must satisfy the stationarity requirement

$$\nabla\widetilde{\varphi}_{\Omega}(\widehat{x},\widehat{y}) = \begin{pmatrix} J_k^{\mathrm{T}} & P_{\!A} \\ \mu I & 0 \end{pmatrix} \begin{pmatrix} \widehat{\pi}(\sigma_{\!J}) \\ \widehat{z}_{\!A}(\sigma_{\!A}) \end{pmatrix} = \widetilde{G}_{\!A}^{\mathrm{T}} \widehat{w}_{\!A}(\Omega).$$

Taking the gradient of the modified quadratic shows this is equivalent to the requirements

$$g_k + \widehat{H}(\Omega)(\widehat{x} - x_k) + \sigma_{\!J}\mu J_k^{\mathrm{T}}(\widehat{y} - y_k) = J_k^{\mathrm{T}}\widehat{\pi}(\sigma_{\!J}) + \widehat{z}_{\!A}(\sigma_{\!A}), \ \text{ and}$$

$$\mu(1 + \sigma_{\!J}\mu)(\widehat{y} - y_k) + \mu(y_k + \sigma_{\!J} J_k(\widehat{x} - x_k)) = \mu\widehat{\pi}(\sigma_{\!J}),$$

which can be expressed as a system of equations

$$\begin{pmatrix} \widehat{H}(\Omega) & \sigma_{\!J}\mu J_k^{\mathrm{T}} \\ \sigma_{\!J}\mu J_k & \mu(1 + \sigma_{\!J}\mu)I \end{pmatrix} \begin{pmatrix} p_k \\ q_k \end{pmatrix} = - \begin{pmatrix} g_k \\ \mu y_k \end{pmatrix} + \begin{pmatrix} J_k^{\mathrm{T}} & P_{\!A} \\ \mu I & 0 \end{pmatrix} \begin{pmatrix} \widehat{\pi}(\sigma_{\!J}) \\ \widehat{z}_{\!A}(\sigma_{\!A}) \end{pmatrix}. \tag{5.44}$$

The feasibility requirement that $(\widehat{x},\ \widehat{y})$ satisfy $c_k + J_k(\widehat{x} - x_k) + \mu(\widehat{y} - y_k) = 0$ along with the identity $P_{\!A}^{\mathrm{T}}(\widehat{x} - x_k) = -[\,x_k\,]_{\!A}$ yields the complementary linear system

$$\begin{pmatrix} J_k & \mu I \\ P_{\!A}^{\mathrm{T}} & 0 \end{pmatrix} \begin{pmatrix} p_k \\ q_k \end{pmatrix} = - \begin{pmatrix} c_k \\ [\,x_k\,]_{\!A} \end{pmatrix}. \tag{5.45}$$

Now collect the equations (5.44) and (5.45) to obtain

$$\begin{pmatrix} \widehat{H}(\Omega) & \sigma_{\!J}\mu J_k^{\mathrm{T}} & J_k^{\mathrm{T}} & P_{\!A} \\ \sigma_{\!J}\mu J_k & \mu(1 + \sigma_{\!J}\mu)I & \mu I & 0 \\ J_k & \mu I & 0 & 0 \\ P_{\!A}^{\mathrm{T}} & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} p_k \\ q_k \\ -\widehat{\pi}(\sigma_{\!J}) \\ -\widehat{z}_{\!A}(\sigma_{\!A}) \end{pmatrix} = - \begin{pmatrix} g_k \\ \mu y_k \\ c_k \\ [\,x_k\,]_{\!A} \end{pmatrix},$$

which is identical to the block system (5.41) representing optimality conditions for the post-

convexified generic QP (5.40). □

5.3 Primal-Dual SQP methods with Dynamic Convexification

In Chapter 3, it was shown that the stabilized QP (5.25) and a certain bound-constrained QP subproblem (3.9) have the same solution. This bound-constrained subproblem involves a quadratic model of a primal-dual merit function and has the form

$$\begin{aligned} \underset{v\in\mathbb{R}^{n+m}}{\text{minimize}} \quad & \nabla M(v_k)^{\mathrm{T}}(v-v_k) + \tfrac{1}{2}(v-v_k)^{\mathrm{T}} H_k^M (v-v_k) \\ \text{subject to} \quad & P_X^{\mathrm{T}} v \geq 0, \end{aligned} \tag{5.46}$$

where $P_X^{\mathrm{T}} = \begin{pmatrix} I_n & 0_{n\times m} \end{pmatrix}$ so that $P_X^{\mathrm{T}} v = x$. Note that as $P_X^{\mathrm{T}} v = x \geq 0$, the dual variables are always free. Define

$$\Pi_{\scriptscriptstyle F} = \begin{pmatrix} P_{\scriptscriptstyle F} & 0 \\ 0 & I_m \end{pmatrix}, \; \Pi_{\scriptscriptstyle A} = \begin{pmatrix} P_{\scriptscriptstyle A} \\ 0 \end{pmatrix}, \text{ and } \Pi = \begin{pmatrix} \Pi_{\scriptscriptstyle F} & \Pi_{\scriptscriptstyle A} \end{pmatrix}, \tag{5.47}$$

where the matrices $P_{\scriptscriptstyle F}$ and $P_{\scriptscriptstyle A}$ are defined as before, taking columns of identity corresponding to indices in the free and active sets. It follows Π is a permutation matrix of dimension $n+m$, and that

$$\Pi^{\mathrm{T}} v = \begin{pmatrix} P_{\scriptscriptstyle F}^{\mathrm{T}} & 0 \\ 0 & I \\ P_{\scriptscriptstyle A}^{\mathrm{T}} & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} P_{\scriptscriptstyle F}^{\mathrm{T}} x \\ y \\ P_{\scriptscriptstyle A}^{\mathrm{T}} x \end{pmatrix} = \begin{pmatrix} x_{\scriptscriptstyle F} \\ y \\ x_{\scriptscriptstyle A} \end{pmatrix} = \begin{pmatrix} v_{\scriptscriptstyle F} \\ v_{\scriptscriptstyle A} \end{pmatrix}.$$

The working set matrix is therefore given by $G_{\scriptscriptstyle A} = \Pi_{\scriptscriptstyle A}^{\mathrm{T}} = \begin{pmatrix} P_{\scriptscriptstyle A}^{\mathrm{T}} & 0 \end{pmatrix}$. We will now comment on how the bound-constrained formulation affects pre-convexification and the observations made in Section 5.2.2.

5.3.1 Pre-convexification of the bound-constrained subproblem

The free columns of G_A are $G_A \Pi_F = \Pi_A^{\mathrm{T}} \Pi_F = 0_{n_A \times (n_F+m)}$, therefore the columns of $Z = I_{n_F+m}$ are a basis for the null space of $[\,G_A\,]_F$. This means the matrix that must be made convex is the reduced Hessian $Z^{\mathrm{T}} H_F^M Z = H_F^M$, which can be done implicitly by modifying the free KKT matrix

$$K_F = \begin{pmatrix} H_F & J_F^{\mathrm{T}} \\ J_F & -\mu I \end{pmatrix}.$$

This is based on the identity $\mathrm{In}(H_F^M) = \mathrm{In}(K_F) + (m, -m, 0)$ that follows from (4.1) and is true regardless of $\mathrm{rank}(J_F)$. The theory developed in Section 5.2.2 applies here with only slight modification. The needed observation is that the rows of $G_A = \begin{pmatrix} P_A^{\mathrm{T}} & 0 \end{pmatrix}$ are necessarily independent from the constraint normal e_i^{T} of any free variable x_i, which implies that both the method of pre-convexification by temporary constraint imposition and the iterations to find a subspace stationary point are guaranteed to terminate successfully regardless of the rank of the Jacobian.

5.3.2 Concurrent convexification of the bound-constrained QP

When $y^E = y_k$, the bound-constrained QP (5.46) is equivalent to the stabilized QP, which is in standard form, so it is reasonable to expect that the same perturbation derived in those cases should work as a concurrent convexification method for the bound-constrained problem. Next, we clarify this relationship and derive the same perturbation starting from (5.46).

Suppose an active-set method is being applied to solve this problem, and that a active-free partition is defined at a subspace stationary point (v_j, z_j). The application of Π symmetrically to

the KKT search direction system analogous to (5.6) yields

$$\begin{pmatrix} H_{F} + \frac{2}{\mu} J_{F}^{\mathrm{T}} J_{F} & J_{F}^{\mathrm{T}} & H_{D} + \frac{2}{\mu} J_{F}^{\mathrm{T}} J_{A} & 0 \\ J_{F} & \mu I & J_{A} & 0 \\ H_{D}^{\mathrm{T}} + \frac{2}{\mu} J_{A}^{\mathrm{T}} J_{F} & J_{A}^{\mathrm{T}} & H_{A} + \frac{2}{\mu} J_{A}^{\mathrm{T}} J_{A} & I \\ 0 & 0 & I & 0 \end{pmatrix} \begin{pmatrix} p_{F} \\ q_{j} \\ p_{A} \\ -r_{A} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ e_{s} \end{pmatrix}.$$

This system reduces to the following doubly-augmented system involving only the free variables

$$\begin{pmatrix} H_{F} + \frac{2}{\mu} J_{F}^{\mathrm{T}} J_{F} & J_{F}^{\mathrm{T}} \\ J_{F} & \mu I \end{pmatrix} \begin{pmatrix} p_{F} \\ q_{j} \end{pmatrix} = - \begin{pmatrix} \left[\left(H_{k} + \frac{2}{\mu} J_{k}^{\mathrm{T}} J_{k} \right)_{\nu_{s}} \right]_{F} \\ J_{\nu_{s}} \end{pmatrix}, \tag{5.48}$$

from which the full directions can be recovered using the identities

$$p_{A} = e_{s} \;\text{ and }\; r_{A} = P_{A}^{\mathrm{T}} \left(\left(H_{k} + \frac{2}{\mu} J_{k}^{\mathrm{T}} J_{k} \right) p_{j} + J_{k}^{\mathrm{T}} q_{j} \right).$$

Suppose that the primal-dual direction is computed from the reduced, doubly-augmented system (5.48), yielding (d_{j}, r_{j}) that satisfy

$$\begin{pmatrix} H_{k}^{M} & G_{A}^{\mathrm{T}} \\ G_{A} & 0 \end{pmatrix} \begin{pmatrix} d_{j} \\ -r_{A} \end{pmatrix} = \begin{pmatrix} 0 \\ e_{s} \end{pmatrix},$$

and that the curvature along d_{j} is not positive

$$d_{j}^{\mathrm{T}} H_{k}^{M} d_{j} = d_{j}^{\mathrm{T}} (G_{A}^{\mathrm{T}} r_{A}) = e_{s}^{\mathrm{T}} r_{A} = [\, r_{A} \,]_{s} \leq 0.$$

Consider the perturbation $H^M(\sigma) = H^M + \sigma e_{\nu_s} e_{\nu_s}^{\mathrm{T}}$ and note that this is can be written as

$$H_k^M(\sigma) = H_k^M + \sigma e_{\nu_s} e_{\nu_s}^{\mathrm{T}} = \begin{pmatrix} H_k(\sigma) + \frac{2}{\mu} J_k^{\mathrm{T}} J_k & J_k^{\mathrm{T}} \\ J_k & \mu I \end{pmatrix}.$$

Therefore, the same perturbation is being applied. The next steps are to verify that this perturbation produces a subspace stationary point and then compute the adjusted dual quantities

$$z_j(\sigma) = z_j - \sigma[\, x_k \,]_{\nu_s} e_{\nu_s} \qquad \text{and} \qquad r_j(\sigma) = r_j + \sigma e_{\nu_s}.$$

As (v_j, z_j) is a subspace stationary point it holds that $\nabla M_k + H_k^M(v_j - v_k) = G_A^{\mathrm{T}} z_A$, therefore

$$\begin{aligned} \nabla M_k + H_k^M(\sigma)(v_j - v_k) &= \nabla M_k + (H_k^M + \sigma e_{\nu_s} e_{\nu_s}^{\mathrm{T}})(v_j - v_k) \\ &= \big(\nabla M_k + H_k^M(v_j - v_k)\big) + \sigma[\, v_j - v_k \,]_{\nu_s} e_{\nu_s} \\ &= G_A^{\mathrm{T}} z_A + \sigma[\, v_j - v_k \,]_{\nu_s} G_A^{\mathrm{T}} e_{\nu_s} \\ &= G_A^{\mathrm{T}}(z_A - \sigma[\, x_k \,]_{\nu_s} e_s) = G_A^{\mathrm{T}} z_A(\sigma), \end{aligned}$$

thus v_j remains a subspace stationary point for $\sigma \geq 0$. As before,

$$\begin{pmatrix} H_k^M(\sigma) & G_A^{\mathrm{T}} \\ G_A & 0 \end{pmatrix} \begin{pmatrix} d_j \\ -r_A(\sigma) \end{pmatrix} = \begin{pmatrix} 0 \\ e_s \end{pmatrix}.$$

The modified optimal step can now be computed as

$$\alpha^*(\sigma) = -\frac{[\, z_A(\sigma) \,]_s}{[\, r_A(\sigma) \,]_s} = -\frac{[\, z_A \,]_s - \sigma[\, x_k \,]_{\nu_s}}{[\, r_A \,]_s + \sigma}.$$

It is worth noting how these modifications affect the reduced system in the free variables. If the same symmetric permutation is applied to the system

$$\begin{pmatrix} H_k^M(\sigma) & G_A^{\mathrm{T}} \\ G_A & 0 \end{pmatrix} \begin{pmatrix} d_j \\ -r_A(\sigma) \end{pmatrix} = \begin{pmatrix} 0 \\ e_s \end{pmatrix},$$

only the third block of equations defining p_A is affected by the convexification. This is because

$$\Pi_F^{\mathrm{T}} e_{\nu_s} = \Pi_F^{\mathrm{T}} \Pi_A e_s = 0 = e_{\nu_s}^{\mathrm{T}} \Pi_F,$$

the perturbed, permuted system is

$$\begin{pmatrix} H_F + \frac{2}{\mu} J_F^{\mathrm{T}} J_F & J_F^{\mathrm{T}} & H_D + \frac{2}{\mu} J_F^{\mathrm{T}} J_A & 0 \\ J_F & \mu I & J_A & 0 \\ H_D^{\mathrm{T}} + \frac{2}{\mu} J_A^{\mathrm{T}} J_F & J_A^{\mathrm{T}} & H_A + \sigma e_s e_s^{\mathrm{T}} + \frac{2}{\mu} J_A^{\mathrm{T}} J_A & I \\ 0 & 0 & I & 0 \end{pmatrix} \begin{pmatrix} p_F \\ q_j \\ p_A \\ -r_A(\sigma) \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ e_s \end{pmatrix}.$$

The third block of equations can be rearranged to give

$$r_A(\sigma) - \sigma e_s = r_A,$$

which confirms that the value derived for $r_A(\sigma)$ implicitly achieves the correct perturbation. This also shows that, if necessary, one could recover

$$H_F(\sigma) = H_F, \;\; H_D(\sigma) = H_D, \;\; \text{and} \;\; H_A(\sigma) = H_A + \sigma e_s e_s^{\mathrm{T}}.$$

Selecting the convexification scale

The choice of σ during concurrent convexification must satisfy certain requirements. There are also some preferences to consider when the requirements leave freedom in the choice of σ. First of all, the convexification modifies the quadratic program being solved and so represents a departure from the truth. For this reason we want σ as small as possible, provided our other criteria are met. For the following discussion, let $\lambda_{\min}$, $d_{\max}$, and τ_D be positive preassigned tolerances controlling minimum curvature, maximum step norm, and dual feasibility (minimum value of an optimal multiplier), respectively.

The modified curvature must satisfy $d_j^{\mathrm{T}} H_k^M(\sigma) d_j \geq \lambda_{\min}$. Recall that

$$e_{\nu_s}^{\mathrm{T}} d_j = e_s^{\mathrm{T}} P_A^{\mathrm{T}} p_j = e_s^{\mathrm{T}} [\, p_j \,]_A = e_s^{\mathrm{T}} e_s = 1,$$

therefore $d_j^{\mathrm{T}}(H_k^M + \sigma e_{\nu_s} e_{\nu_s}^{\mathrm{T}}) d_j = d_j^{\mathrm{T}} H_k^M d_j + \sigma = [\, r_A \,]_s + \sigma$. This provides a lower bound $\sigma_{\min} = \lambda_{\min} - [\, r_A \,]_s$ such that the resulting curvature is sufficiently positive for all $\sigma \geq \sigma_{\min}$. The remaining discussion depends critically on the sign of $[\, x_k \,]_{\nu_s}$ because it essentially determines whether the convexification increases or decreases the nonoptimal multiplier.

- **Case: $[\, \boldsymbol{x_k} \,]_{\boldsymbol{\nu_s}} < 0$.** In this case the multiplier $[\, z_A(\sigma) \,]_s = [\, z_A \,]_s - \sigma [\, x_k \,]_{\nu_s}$ is *increasing*. Consequently, the multiplier can be made optimal by taking $\sigma \geq \sigma_O$, where

$$\sigma_O = \frac{[\, z_A \,]_s - \tau_D}{[\, x_k \,]_{\nu_s}}. \tag{5.49}$$

  In this situation there is no need to step along p_j because another multiplier can be selected and a new direction computed.

- **Case:** $[\,\boldsymbol{x_k}\,]_{\boldsymbol{\nu_s}} \geq \mathbf{0}$. In this case the multiplier $[\,z_A(\sigma)\,]_s < 0$ is negative and *decreasing*, so we cannot drive it to optimality by convexification alone and will have to step along p_j. The resulting change in primal variables must satisfy

$$\|x_{j+1} - x_j\| \leq d_{\max}, \tag{5.50}$$

which is equivalent to an upper bound on the step $\alpha(\sigma) \leq d_{\max}/\|p_j\| = \alpha_{\max}$. For all $\sigma \geq \sigma_{\min}$ we have

$$\frac{d}{d\sigma}\alpha(\sigma) = \frac{[\,z_A\,]_s + [\,r_A\,]_s[\,x_k\,]_{\nu_s}}{([\,r_A\,]_s + \sigma)^2} < 0.$$

This shows the step size is decreasing so we can make σ large enough to get the step size within tolerance, provided $\lim_{\sigma\to\infty}\alpha(\sigma) = [\,x_k\,]_{\nu_s} < \alpha_{\max}$. The value of σ that achieves this bound is

$$\sigma_D = -\frac{[\,z_A\,]_s + [\,r_A\,]_s\alpha_{\max}}{\alpha_{\max} - [\,x_k\,]_{\nu_s}}.$$

In summary, the choice of σ can be expressed as

$$\sigma = \begin{cases} \max(\sigma_{\min}, \sigma_O), & [\,x_k\,]_{\nu_s} < 0 \\ \max(\sigma_{\min}, \sigma_D), & [\,x_k\,]_{\nu_s} \geq 0. \end{cases}$$

5.3.3 Post-convexification of the bound-constrained QP

This section will explore post-convexification of the bound-constrained QP and the limitations of doing so. The way that primal-dual SQP methods deal with equality constraints by incorporating them in the merit function results in subproblems subject only to simple bounds. Though advantageous in other respects, this makes the optimality-preserving post-convexification

strategy described in (5.2.4) difficult to apply because there are no equality constraint multipliers. The reason for the difference between this case and the stabilized derivation is that the equivalence of the bound-constrained and stabilized QP subproblems relies on the primal "y-variables" being identical to the equality constraint multipliers, or "π-values". This equality comes straight out of the stabilized QP optimality conditions. However, the optimality preserving post-convexification strategy modifies $\widehat{\pi} \mapsto \widehat{\pi}(\sigma_{\scriptscriptstyle J})$ but leaves $\widehat{y}$ unmodified, invalidating the link between bound-constrained and stabilized QP subproblems.

We will now investigate several approaches that have been considered for deriving an optimality-preserving strategy that can be applied *implicitly*, and discuss their limitations. The approaches we will consider for post-convexification are

1. The direct derivation - This results in a shift in the inequality multipliers, which may become negative.

2. Starting with the post-convexified stabilized QP of Theorem 5.2.2 and deriving the equivalent bound-constrained QP using a nonsingular transformation as in [40].

3. Allowing the modification to affect y_k or $y^{\scriptscriptstyle E}$ (or both) with which the the merit function is constructed.

The direct approach

Recall that the bound-constrained SQP method is based on the augmented Lagrangian merit function (3.2)

$$M(x, y\,; y^{\scriptscriptstyle E}, \mu) = f(x) - c(x)^{\mathrm{T}} y^{\scriptscriptstyle E} + \frac{1}{2\mu}\|c(x)\|^2 + \frac{1}{2\mu}\|c(x) + \mu(y - y^{\scriptscriptstyle E})\|^2,$$

and solves the QP subproblem (3.9). A primal-dual solution $d_k = (p_k, q_k)$ of the partially convexified analogue of (3.9) must satisfy

$$\begin{pmatrix} \widehat{H}^{M} & G_{A}^{\mathrm{T}} \\ G_{A} & 0 \end{pmatrix} \begin{pmatrix} d_k \\ -\widehat{z}_{A} \end{pmatrix} = - \begin{pmatrix} \nabla M_k \\ P_{A}^{\mathrm{T}} x_k \end{pmatrix}, \tag{5.51}$$

where the working set matrix in this case has the form

$$G_{A} = P_{A}^{\mathrm{T}} P_{X}^{\mathrm{T}} = \begin{pmatrix} P_{A}^{\mathrm{T}} & 0 \end{pmatrix}.$$

If the merit function is regarded as an objective function subject to the equality constraint $[\, x \,]_{A} = 0$, i.e., $G_{A} v = 0$, then the corresponding Lagrangian is

$$L(v, z_{A}) = M(v) - z_{A}^{\mathrm{T}} G_{A} v. \tag{5.52}$$

The goal is to ensure the quantity d_k defines a descent direction for this Lagrangian, that is,

$$\nabla L(v_k, \widehat{z}_{A})^{\mathrm{T}} d_k < 0.$$

From the equations (5.51) it follows $\widehat{H}^{M} d_k = -\nabla M_k + G_{A}^{\mathrm{T}} \widehat{z}_{A} = -\nabla L(v_k \,; \widehat{z}_{A})$, so

$$d_k^{\mathrm{T}} \widehat{H}^{M} d_k = -\nabla L(v_k, \widehat{z}_{A})^{\mathrm{T}} d_k,$$

which means all that is needed to ensure a descent direction is to enforce the curvature to be sufficiently positive, i.e.,

$$\widehat{H}^{M}(\sigma) = \widehat{H}^{M} + \sigma G_{A}^{\mathrm{T}} G_{A} = \begin{pmatrix} \widehat{H} + \sigma P_{A} P_{A}^{\mathrm{T}} + \frac{2}{\mu} J_{k}^{\mathrm{T}} J_{k} & J_{k}^{\mathrm{T}} \\ J_{k} & \mu I \end{pmatrix},$$

where

$$\sigma = (\lambda_{\min} - \lambda) \frac{\|d_{k}\|^{2}}{\|[\, x_{k} \,]_{A}\|^{2}}.$$

Comparing the systems

$$\begin{pmatrix} \widehat{H}^{M} + \sigma G_{A}^{\mathrm{T}} G_{A} & G_{A}^{\mathrm{T}} \\ G_{A} & 0 \end{pmatrix} \begin{pmatrix} d_{k}(\sigma) \\ -\widehat{z}_{A}(\sigma) \end{pmatrix} = \begin{pmatrix} \widehat{H}^{M} & G_{A}^{\mathrm{T}} \\ G_{A} & 0 \end{pmatrix} \begin{pmatrix} d_{k} \\ -\widehat{z}_{A} \end{pmatrix}$$

shows that $d_{k}(\sigma) = d_{k}$ and $\widehat{z}_{A}(\sigma) = \widehat{z}_{A} - \sigma [\, x_{k} \,]_{A}$ implicitly applies the convexification.

This approach is relatively straightforward and can be applied implicitly. The major drawback is that there is no guarantee that $\widehat{z}_{A}(\sigma)$ will remain nonnegative if the active set changes during solution of the QP subproblem.

Post-convexification via the equivalence of stabilized QP

Although post-convexifying makes it so that the primal y-variables and the dual π-values no longer agree, breaking the direct link between the stabilized problem and the bound-constrained problem, the identity $\widehat{\pi}(\sigma_{J}) = \widehat{y} + \sigma_{J}(J_{k} p_{k} + \mu q_{k})$ may still be used to eliminate $\widehat{\pi}(\sigma_{J})$ from the post-convexified optimality conditions of the stabilized QP studied in Theorem 5.2.2.

Recall the stationarity condition for the post-convexified stabilized QP

$$g_k + \widehat{H}(\Omega)(\widehat{x} - x_k) + \sigma_{J}\mu J_k^{\mathrm{T}}(\widehat{y} - y_k) = J_k^{\mathrm{T}}\widehat{\pi}(\sigma_{J}) + \widehat{z}_{A}(\sigma_{A}), \text{ and}$$

$$\mu(1 + \sigma_{J}\mu)(\widehat{y} - y_k) + \mu(y_k + \sigma_{J}J_k(\widehat{x} - x_k)) = \mu\widehat{\pi}(\sigma_{J}),$$

where $\widehat{H}(\Omega) = \widehat{H} + \sigma_{J}J_k^{\mathrm{T}}J_k + \sigma_{A}P_{A}P_{A}^{\mathrm{T}}$ is the $(1,\ 1)$-block of $\check{H}(\Omega)$. The second equality reduces to $\widehat{\pi}(\sigma_{J}) = \widehat{y} + \sigma_{J}(J_k p_k + \mu q_k)$, which can be used to eliminate $\widehat{\pi}(\sigma_{J})$ in the first equality, giving

$$g_k + \widehat{H}(\Omega)p_k + \sigma_{J}\mu J_k^{\mathrm{T}}q_k = J_k^{\mathrm{T}}(\widehat{y} + \sigma_{J}(J_k p_k + \mu q_k)) + \widehat{z}_{A}(\sigma_{A}).$$

Notice that $\sigma_{J}\mu J_k^{\mathrm{T}}q_k$ and $\sigma_{J}J_k^{\mathrm{T}}J_k p_k$ both appear on both sides of the equality, therefore

$$g_k + (\widehat{H} + \sigma_{A}P_{A}P_{A}^{\mathrm{T}})p_k = J_k^{\mathrm{T}}\widehat{y} + \widehat{z}_{A}(\sigma_{A}).$$

This has reduced back down to the modified $\widehat{H}^{M}(\sigma) = \widehat{H}^{M} + \sigma G_{A}^{\mathrm{T}}G_{A}$ with $\sigma = \sigma_{A}$, effectively removing the ability to convexify implicitly.

Post-convexification by modifying the merit function

Rather than using the stabilized SQP method as a back-door to the bound-constrained method, we will now explore applying convexification implicitly by modifying the merit function. The idea is to realize the convexification by modifying the multipliers y_k or y^{E} appearing in the merit function, instead of the change in multipliers q_k. In what follows, we will assume none of the post-convexification is done by shifting the bound constraint multipliers, and focus solely on change y_k and y^{E}.

Let $(\widehat{v},\ \widehat{z})$ be the primal-dual solution of the partially convexified QP subproblem

$$\begin{aligned} \underset{v\in\mathbb{R}^{n+m}}{\text{minimize}} \quad & \nabla M(v_k)^{\mathrm{T}}(v-v_k)+\tfrac{1}{2}(v-v_k)^{\mathrm{T}}\widehat{H}_k^{M}(v-v_k) \\ \text{subject to} \quad & P_{A}^{\mathrm{T}}v\geq 0, \end{aligned} \tag{5.53}$$

where $\widehat{H}^{M}=H^{M}+P_{X}(\Delta+\Sigma)P_{X}^{\mathrm{T}}$ is the partially convexified approximate merit Hessian. Then $(d_k,\ \widehat{z}_{A})$ must satisfy the stationarity requirement

$$\widehat{H}^{M}d_k-G_{A}^{\mathrm{T}}\widehat{z}_{A}=-\nabla M_k \tag{5.54}$$

The merit function $M(x,y)\equiv M(x,y\,;y^{E},\mu)$ has the property that for any $u\in\mathbb{R}^{m}$

$$\nabla M(x,y+u)=\nabla M(x,y)+\widetilde{J}^{\mathrm{T}}u. \tag{5.55}$$

In order to determine a value of u that will allow the convexification to be implicit, consider making a positive semidefinite modification $\widehat{H}^{M}+\Gamma$ where $\Gamma=\sigma_{J}\widetilde{J}^{\mathrm{T}}\widetilde{J}$. The perturbed system is then

$$(\widehat{H}^{M}+\Gamma)d_k-G_{A}^{\mathrm{T}}\widehat{z}_{A}=-\nabla M(x_k,y_k+u). \tag{5.56}$$

Using (5.54) and (5.55) shows this system is equivalent to $\Gamma d_k=-\widetilde{J}^{\mathrm{T}}u$, and when the desired form of Γ is substituted, it becomes

$$\widetilde{J}^{\mathrm{T}}\big(\sigma_{J}\widetilde{J}d_k+u\big)=0,$$

which is satisfied by the quantity $u=-\sigma_{J}\widetilde{J}d_k$. Therefore, given the solution of the system (5.54), the solution of the perturbed system (5.56) is obtained without needing to solve it again. All that

is needed is to compute the shift $y(\sigma_J) = y_k - \sigma_J \widetilde{J} d_k$ and the modified merit function $M(x, y(\sigma_J)) = M(x, y - \sigma_J \widetilde{J} d_k)$.

As before, we want a descent direction for the augmented Lagrangian function with the modified multipliers, which now has the form

$$L(x, y(\sigma_J), \widehat{z}_A) = M(x, y(\sigma_J)) - \widehat{z}_A^{\mathrm{T}} G_A^{\mathrm{T}} \begin{pmatrix} x \\ y(\sigma_J) \end{pmatrix}$$

The identity (5.56) implies that the directional derivative of along d_k is

$$d_k^{\mathrm{T}} \nabla L(x_k, y_k(\sigma_J), \widehat{z}_A) = -d_k^{\mathrm{T}} (\widehat{H}^M + \Gamma) d_k.$$

As Γ is positive semidefinite, σ_J can be computed to achieve

$$d_k^{\mathrm{T}} (\widehat{H}^M + \Gamma) d_k = \lambda_{\min} \|d_k\|^2.$$

It turns out that the needed shift to y_k is the same as the modification to $\widehat{\pi}$ derived for the stabilized QP in Theorems 5.2.1 and 5.2.2 given by $\widehat{\pi}(\sigma_J) = \widehat{\pi} + \sigma_J (J_k p_k + \mu q_k) = \widehat{\pi} + \sigma_J \widetilde{J} d_k$. This indicates that this approach is successful in reproducing the convexification that was developed for stabilized SQP, but with the caveat that monotonicity of the merit function must be safeguarded.

Shifting y_k and y^E

It may also be reasonable to shift the multiplier estimate y^E by the same amount as y_k. The gradient of the merit function has the property that

$$\nabla M(x, y+u\,; y^E+u) = \nabla M(x,y) - \begin{pmatrix} J^{\mathrm{T}}u \\ 0 \end{pmatrix},$$

which implies that if both y_k and y^E are shifted then the equivalent of (5.56) is

$$(\widehat{H}^{M} + \Gamma)d_k - G_{A}^{\mathrm{T}}\widehat{z}_{A} = -\nabla M(x_k, y_k+u\,; y^E+u) = -\big(\nabla M(x_k, y_k\,; y^E) - P_{X}J_k^{\mathrm{T}}u\big). \tag{5.57}$$

This suggests a perturbation of the form $\Gamma = \sigma_{J}P_{X}J^{\mathrm{T}}JP_{X}^{\mathrm{T}}$, along with the corresponding shift to the multipliers given by $u = \sigma_{J}J_kp_k$. These forms satisfy

$$(\widehat{H}^{M} + \Gamma)d_k - G_{A}^{\mathrm{T}}\widehat{z}_{A} = -\nabla M(x_k, y_k+u\,; y^E+u).$$

This formulation has the added advantage that the norm of the perturbation will always be less than if $\sigma_{J}\widetilde{J}^{\mathrm{T}}\widetilde{J}$ is used. Moreover, only the first n rows and columns of $\widehat{H}^{M}$ are affected.

Modifying y^E only

If the multiplier estimate y^E is shifted by a vector u but y_k is not changed, the resulting merit function gradient satisfies

$$\nabla M(v_k\,; y^E+u, \mu) = \nabla M(v_k\,; y^E, \mu) - \begin{pmatrix} 2J^{\mathrm{T}} \\ \mu I \end{pmatrix} u.$$

To enable a *implicit* post-convexification, a symmetric positive semidefinite modification Γ to $\widehat{H}^{M}$ is sought such that

$$(\widehat{H}^{M} + \Gamma)d_k - G_{A}^{\mathrm{T}}\widehat{z}_{A} = -\nabla M(v_k\,; y^{E} + u) = -\nabla M(v_k) + \begin{pmatrix} 2J_k^{\mathrm{T}} \\ \mu I \end{pmatrix} u. \tag{5.58}$$

It is readily verified that the quantities

$$\Gamma = \sigma_{J}\bar{J}^{\mathrm{T}}\bar{J} \;\text{ with }\; \bar{J} \triangleq \sqrt{\frac{\mu}{2}}\begin{pmatrix} J_k & \frac{2}{\mu}I \end{pmatrix}, \qquad \text{and} \qquad u = \frac{\sigma_{J}}{\mu}\left(J_k p + \frac{\mu}{2}q\right)$$

satisfy (5.58). As in the preceding approaches, σ_{J} can be chosen so that $d_k^{\mathrm{T}}(\widehat{H}^{M} + \Gamma)d_k$ is positive enough to produce a direction of sufficient decrease.

Safeguarding dynamic convexification

The primary concern when doing post-convexification by modifying the multipliers is that it can create non-monotonicity in the merit function, interfering with the flexible line search used to guarantee global convergence. Though preliminary numerical results suggest this is relatively uncommon, the implicit post-convexification can only be computed if the shift to y_k or y^{E} decreases the merit function. In the event that an increase results, we must compute a full convexification and re-solve the QP instead.

Suppose post-convexification is required, and it produces non-monotonicity in the merit function. If the method of pre-convexification used is one of the Hessian modification options, much of the full convexification may already by done. This is a result of the fact that if K_{F} is second-order

consistent, then for sufficiently large $\sigma_A > 0$ it holds that

$$\text{In} \begin{pmatrix} H_k + \Delta + \sigma_A P_A P_A^{\mathrm{T}} & J_k^{\mathrm{T}} \\ J_k & -\mu I \end{pmatrix} = (n, m, 0), \tag{5.59}$$

(see Gill and Robinson [40] for more details). Essentially, the convexification can be completed simply by modifying diagonal entries of H_k with indices in the active set. However, if the method of temporary constraints is used then $\Delta = 0$ and the active diagonal modification must be done with respect to $\widehat{\mathcal{A}}(x_k) = \mathcal{A}(x_k) \cup \mathcal{X}$, where $\mathcal{X}$ is the set of temporarily fixed indices defined in Section 5.1.2. The reason is that $\mathcal{F}(x_k)$ is generally not a second-order consistent basis while $\widehat{\mathcal{F}}(x_k)$ is, which is required for (5.59) to be applicable.

Even if the concurrent convexification modifications are taken into consideration, it may still be that (5.59) defined with $H_k + \Sigma + \sigma_A P_A P_A^{\mathrm{T}}$ may never be second-order consistent for any σ_A. First, unless strict complementarity holds, concurrent convexification is not guaranteed to modify all diagonal entries with indices in $\mathcal{X}$. The active-set method will only drive z_{ν_s} for $\nu_s \in \mathcal{X}$ to optimality if z_{ν_s} is nonzero, where $z = g_k + H_k(x - x_k) - J_k^{\mathrm{T}} y$. Though z_{ν_s} can be nonzero, it is not required to be. Second, even if z_{ν_s} is non-optimal, concurrent convexification is designed to only consider curvature of the QP objective along the specific directions computed by the active-set method. There may be other directions of negative curvature such that $H_F + \frac{1}{\mu} J_F^{\mathrm{T}} J_F$ is indefinite, or equivalently, K_F is not second-order consistent.

Chapter 6

A Dynamically-Convexified Primal-Dual SQP Algorithm

In this section we will focus on the bound constrained primal-dual formulation of the second-derivative SQP method with special attention given to presentation of the algorithm and analysis of convergence.

6.1 Formal Algorithm Statement

The main algorithm Algorithm 3 will make repeated use of the active-set algorithm Algorithm 2 for solving the stabilized QP subproblem (3.1) repeated here

$$\begin{aligned} \underset{x,y}{\text{minimize}} \quad & g_k^{\mathrm{T}}(x - x_k) + \tfrac{1}{2}(x - x_k)^{\mathrm{T}} \widehat{H}_k (x - x_k) + \tfrac{1}{2}\mu_k \|y\|^2 \\ \text{subject to} \quad & c_k + J_k(x - x_k) + \mu_k(y - y_k) = 0, \quad x \geq 0. \end{aligned} \tag{6.1}$$

Algorithm 2 Stabilized QP subproblem with concurrent convexification.

Input $(x_k,\ y_k)$; Choose (x, y) such that $x \geq 0$;
Compute $\mathcal{A} = \mathcal{A}(x)$ and $\mathcal{F} = \mathcal{F}(x)$;
Set $z = g + \widehat{H}(x - x_k) - J^{\mathrm{T}} y$;
repeat
    Select index $\nu_s \in \mathcal{A}(x)$ of a nonoptimal multiplier;
    repeat
        Solve $\begin{pmatrix} \widehat{H}_{F} & J_{F}^{\mathrm{T}} \\ J_{F} & -\mu I \end{pmatrix} \begin{pmatrix} p_{F} \\ q \end{pmatrix} = - \begin{pmatrix} [\,\widehat{H}_{F}\,]_{\nu_s} \\ J_{\nu_s} \end{pmatrix}$; $p_{A} = e_s$;
        $r_{A} = [\,\widehat{H}p - J^{\mathrm{T}} q\,]_{A}$; $\lambda = [\,r\,]_{\nu_s}$;
        if $\lambda < \lambda_{\min}$ **then** [Concurrent convexification]
            Compute σ according to (5.3.2);
            $\widehat{H} \leftarrow \widehat{H} + \sigma e_{\nu_s} e_{\nu_s}^{\mathrm{T}}$; $z \leftarrow z - \sigma [\,x_k\,]_{\nu_s} e_{\nu_s}$; $r \leftarrow r + \sigma e_{\nu_s}$;
        end if
        $\alpha_{\mathrm{opt}} = -\dfrac{[\,z\,]_{\nu_s}}{[\,r\,]_{\nu_s}}$;
        $t = \underset{i \in \mathcal{F}(x), p_i < 0}{\operatorname{argmin}} \left\{ -\dfrac{x_i}{p_i} \right\}$; $\alpha_{\max} = -\dfrac{x_t}{p_t}$;
        $\alpha = \min(\alpha_{\max},\ \alpha_{\mathrm{opt}})$;
        $x \leftarrow x + \alpha p$; $y \leftarrow y + \alpha q$; $z \leftarrow z + \alpha r$;
        if $\alpha_{\mathrm{opt}} \geq \alpha_{\max}$ **then** [t becomes active]
            $\mathcal{A} \leftarrow \mathcal{A} \cup \{\, t \,\}$; $\mathcal{F} \leftarrow \mathcal{F} \setminus \{\, t \,\}$;
        end if
    until $[\,z\,]_{\nu_s} \geq 0$
    $\mathcal{A} \leftarrow \mathcal{A} \setminus \{\, \nu_s \,\}$; $\mathcal{F} \leftarrow \mathcal{F} \cup \{\, \nu_s \,\}$;
until $\min z \geq \tau_{D}$
return $(\widehat{x},\ \widehat{y},\ \widehat{z}) = (x,\ y,\ z)$;

Algorithm 3 `dcpdSQP`: Primal-dual SQP method with dynamic convexification.

Input $v_0 = (x_0,\ y_0)$; $k \leftarrow 0$;
Evaluate f, g, c, J, and H at (x_k, y_k);
while $k \leq k_{\max}$ and $\|r_{\text{opt}}(v_k)\| \leq \tau_P$ **do**
    Compute Δ such that $[\,H^M + \Delta\,]_F$ is positive definite; [pre-convexification]
    Solve the stabilized QP subproblem for $(\widehat{x}, \widehat{y}, \widehat{z})$ and Σ using Algorithm (2);
    $d_k = (\widehat{x} - x_k,\ \widehat{y} - y_k) = (p_k,\ q_k)$;
    $\lambda = d_k^{\mathrm{T}} H^M_{\Delta\Sigma} d_k / \|d_k\|^2$;
    if $\lambda < \lambda_{\min}$ **then** [post-convexification]
        Compute Γ such that $d_k^{\mathrm{T}} H^M_{\Delta\Sigma\Gamma} d_k \geq \lambda_{\min} \|d_k\|^2$;
        Compute $y_k(\sigma)$ and $y^E(\sigma)$ according to (5.3.3);
        if $M(x_k, y_k(\sigma)\,; y^E(\sigma), \mu) \leq M(x_k, y_k\,; y^E, \mu)$ **then**
            $y_k \leftarrow y_k(\sigma)$; $y^E \leftarrow y^E(\sigma)$;
        else
            Update Δ so that $H^M + \Delta$ is positive definite; $\Sigma = 0$; $\Gamma = 0$;
            Solve the convex QP (3.9) for d_k;
        end if
    end if
    Execute flexible line search for α_k satisfying (3.15) and (3.16);
    Update $(x_{k+1},\ y_{k+1}) = (x_k,\ y_k) + \alpha_k (p_k,\ q_k)$;
    Evaluate f, g, c, J, and H at (x_{k+1}, y_{k+1});
    $(\phi_V^{\max}, \phi_O^{\max}, y^E_{k+1}, \tau_{k+1}) = \texttt{pseudo-filter}(x_{k+1}, y_{k+1}, \phi_V^{\max}, \phi_O^{\max}, y^E_k, \tau_k)$;
    Update μ_k^R and μ_k according to (3.21) and (3.23)
    $k \leftarrow k + 1$;
end while
return $(x^*,\ y^*,\ z^*) = (x_k,\ y_k,\ g(x_k) - J(x_k)^{\mathrm{T}} y_k)$;

Algorithm 4 `pseudo-filter`: Pseudo-filter parameter update.

Input $x_{k+1}, y_{k+1}, \phi_V^{\max}, \phi_O^{\max}, y_k^E, \tau_k$;
if $\phi_V(x_{k+1}, y_{k+1}) \leq \frac{1}{2}\phi_V^{\max}$ **then** [V-iterate]
    $\phi_V^{\max} = \frac{1}{2}\phi_V^{\max}$;
    $y_{k+1}^E = y_{k+1}$;
    $\tau_{k+1} = \tau_k$;
else if $\phi_O(x_{k+1}, y_{k+1}) \leq \frac{1}{2}\phi_O^{\max}$ **then** [O-iterate]
    $\phi_O^{\max} = \frac{1}{2}\phi_O^{\max}$;
    $y_{k+1}^E = y_{k+1}$;
    $\tau_{k+1} = \tau_k$;
else if v_{k+1} satisfies (3.19) **then** [M-iterate]
    $y_{k+1}^E = \max(-y_{\max}e, \min(y_{k+1}, y_{\max}e))$;
    $\tau_{k+1} = \frac{1}{2}\tau_k$;
else [F-iterate]
    $y_{k+1}^E = y_k^E$;
    $\tau_{k+1} = \tau_k$;
end if
return $(\phi_V^{\max}, \phi_O^{\max}, y_{k+1}^E, \tau_{k+1})$;

6.2 Convergence

The convergence of Algorithm 3 is discussed under the following assumptions.

Assumption 6.2.1. *Each $\widehat{H}_k = H_k + \Delta + \Sigma + \Gamma$ is computed using dynamic convexification.*

Assumption 6.2.2. *The functions f and c are twice continuously differentiable.*

Assumption 6.2.3. *The sequence $\{x_k\}_{k\geq 0}$ is contained in a compact set.*

In the "worst" case, i.e., when all iterates are eventually M-iterates or F-iterates, Algorithm 3 emulates a *primal-dual* augmented Lagrangian method [12, 13, 62]. Consequently, it is

possible that y_k^E and μ_k^R will remain fixed over a sequence of iterations, although this has been uncommon in our preliminary numerical results. The following result concerns this situation.

Theorem 6.2.1. *Let Assumptions* 6.2.1–6.2.3 *hold. If there exists an integer* $\widehat{k}$ *such that* $\mu_k^R \equiv \mu^R > 0$ *and* k *is an F-iterate for all* $k \geq \widehat{k}$*, then the following hold for the search directions* $d_k = (\widehat{x}_k - x_k, \widehat{y}_k - y_k)$*, where* $(\widehat{x}_k, \widehat{y}_k)$ *is the solution of subproblem* (3.9)*;*

(i) $\{d_k\}_{k \geq \widehat{k}}$ *are uniformly bounded;*

(ii) $\{d_k\}_{k \geq \widehat{k}}$ *are bounded away from zero; and*

(iii) *there exists a constant* $\epsilon > 0$ *such that*

$$\nabla M(v_k\,; y_k^E, \mu_k^R)^{\mathrm{T}} d_k \leq -\epsilon \;\; \textit{for all } k \geq \widehat{k}.$$

Proof. The assumptions of this theorem imply that

$$\tau_k \equiv \tau > 0, \;\; \mu_k^R = \mu^R, \;\; \text{and} \;\; y_k^E = y^E \;\; \text{for all } k \geq \widehat{k}. \tag{6.2}$$

First we prove part (i). From Assumption 6.2.1 we know concurrent convexification is used during solution of each QP subproblem, and consequently that the change in primal variables at each inner iteration satisfies $\|p_j\| = \|x_{j+1} - x_j\| \leq \tau_D$ as shown in (5.50). The total number of QP steps per iteration can be bounded by the same constant N, and therefore the sequence $\|p_k\| \leq N\tau_D$ so that $\{p_k\}$ is a uniformly bounded sequence.

The change in multipliers q_j is computed from the system (5.34), repeated here

$$\begin{pmatrix} (H + \Delta + \Sigma_j)_{F} & J_{F}^{\mathrm{T}} \\ J_{F} & -\mu_k^R I_m \end{pmatrix} \begin{pmatrix} p_{F} \\ -q_j \end{pmatrix} = - \begin{pmatrix} [\, H_{\nu_s}\,]_{F} \\ J_{\nu_s} \end{pmatrix},$$

where Σ_j is the partial sum of (5.13) for $i \leq j$, therefore

$$q_j = \frac{1}{\mu_k^R}(J_F p_F - J_{\nu_s}).$$

Uniform boundedness of $\{q_k\}_{k \geq \widehat{k}}$ now follows from (6.2), Assumptions 6.2.2 and 6.2.3, and the boundedness of $\{p_k\}$. This completes the proof of part (i).

Part (ii) is established by showing that $\{\|d_k\|\}_{k \geq \widehat{k}}$ is bounded away from zero. If this were not the case, there would exist a subsequence $\mathcal{S}_1 \subseteq \{k : k \geq \widehat{k}\}$ such that $\lim_{k \in \mathcal{S}_1} d_k = 0$, where $d_k = (\widehat{x}_k - x_k, \widehat{y}_k - y_k)$ and $(\widehat{x}_k, \widehat{y}_k)$ is a solution of problem (3.9). From Assumptions 6.2.1-6.2.3 we have

$$H_{\Delta\Sigma}^M = \begin{pmatrix} H_k + \Delta + \Sigma + \frac{2}{\mu_k^R} J_k^{\mathrm{T}} J_k & J_k^{\mathrm{T}} \\ J_k & \mu_k^R I \end{pmatrix},$$

and that $\{(H_{\Delta\Sigma}^M)_k\}_{k \in \mathcal{S}_1}$ is uniformly bounded. It follows that d_k satisfies

$$\begin{pmatrix} \widehat{z}_k \\ 0 \end{pmatrix} = H_{\Delta\Sigma}^M d_k + \nabla M(v_k\,; y^E, \mu^R) \quad \text{and} \quad 0 = \min(\widehat{x}_k, \widehat{z}_k),$$

for all $k \in \mathcal{S}_1$. It may then be inferred from Assumptions 6.2.1–6.2.3, and the definitions (6.2) of τ_k, μ_k^R and y_k^E that for $k \in \mathcal{S}_1$ sufficiently large, the iterate v_k satisfies the definition (3.19) of an M-iterate, and as a consequence, μ_k^R will be decreased. This contradicts the assumption that $\mu_k^R \equiv \mu^R$ for all $k \geq \widehat{k}$. It follows that $\{\|d_k\|\}_{k \geq \widehat{k}}$ is bounded away from zero and part (ii) holds.

The proof of part (iii) is immediate when post-convexification is used, in which case we have

$$-\nabla M(x_k, y_k + w\,; y^E + w, \mu^R)^{\mathrm{T}} d_k = d_k^{\mathrm{T}} \widehat{H}^M d_k \geq \lambda_{\min} \|d_k\|^2.$$

As $\{d_k\}_{k \geq \widehat{k}}$ is bounded away from zero by part (ii), part (iii) follows. Otherwise, assume that there

exists a subsequence $\mathcal{S}_2$ of $\{k : k \geq \widehat{k}\}$ such that

$$\lim_{k \in \mathcal{S}_2} \nabla M(v_k\,; y^{E}, \mu^{R})^{\mathrm{T}} d_k = 0, \tag{6.3}$$

where we have used (6.2) and d_k is defined as above. As the vector $v_k = (x_k, y_k)$ is feasible for the convex problem (3.9), and $(\widehat{x}_k, \widehat{y}_k)$ is the solution of problem (3.9) in Algorithm 3, it must hold that

$$\begin{aligned}
-\nabla M(v_k\,; y^{E}, \mu^{R})^{\mathrm{T}} d_k &\geq \tfrac{1}{2} d_k^{\mathrm{T}} B(v_k\,; \mu^{R}) d_k \\
&= \tfrac{1}{2} d_k^{\mathrm{T}} L_k^{-T} L_k^{\mathrm{T}} B(v_k\,; \mu^{R}) L_k L_k^{-1} d_k \\
&= \tfrac{1}{2} d_k^{\mathrm{T}} L_k^{-T} \begin{pmatrix} \widehat{H}_k + \frac{1}{\mu^{R}} J_k^{\mathrm{T}} J_k & 0 \\ 0 & \nu \mu^{R} \end{pmatrix} L_k^{-1} d_k,
\end{aligned}$$

where L_k denotes the nonsingular matrix

$$L_k = \begin{pmatrix} I & 0 \\ -\frac{1}{\mu^{R}} J_k & I \end{pmatrix}, \quad \text{with} \quad L_k^{-1} d_k = \begin{pmatrix} p_k \\ q_k + \frac{1}{\mu^{R}} J_k p_k \end{pmatrix},$$

with $p_k = \widehat{x}_k - x_k$ and $q_k = \widehat{y}_k - y_k$. Assumption 6.2.1 yields

$$\begin{aligned}
-\nabla M(v_k\,; y^{E}, \mu^{R})^{\mathrm{T}} d_k &\geq \tfrac{1}{2} p_k^{\mathrm{T}} \Big(\widehat{H}_k + \frac{1}{\mu^{R}} J_k^{\mathrm{T}} J_k \Big) p_k + \tfrac{1}{2} \nu \mu^{R} \| q_k + (1/\mu^{R}) J_k p_k \|^2 \\
&\geq \lambda_{\min} \|p_k\|^2 + \tfrac{1}{2} \nu \mu^{R} \| q_k + (1/\mu^{R}) J_k p_k \|^2,
\end{aligned}$$

for some $\lambda_{\min} > 0$. Combining this inequality with (6.3) gives the limit

$$\lim_{k \in \mathcal{S}_2} p_k = \lim_{k \in \mathcal{S}_2} \left(q_k + \frac{1}{\mu^R} J_k p_k \right) = 0,$$

in which case $\lim_{k \in \mathcal{S}_2} q_k = 0$ follows from Assumptions 6.2.2 and 6.2.3. This contradicts the result of part (ii) and so part (iii) must hold. □

The following theorem states the main convergence result for Algorithm 3.

Theorem 6.2.2. *Let Assumptions* 6.2.1–6.2.3 *hold. If v_k denotes the kth iterate generated by Algorithm* 3*, then either:*

(i) *Algorithm* 3 *terminates with an approximate primal-dual first-order solution v_k satisfying $\|r_{\text{opt}}(v_k)\| \leq \tau_{\text{opt}}$, where r_{opt} is defined by* (3.22)*; or*

(ii) *there exists a subsequence $\mathcal{S}$ such that $\lim_{k \in \mathcal{S}} \mu_k^R = 0$, $\{y_k^E\}_{k \in \mathcal{S}}$ is bounded, $\lim_{k \in \mathcal{S}} \tau_k = 0$, and for each $k \in \mathcal{S}$ the vector v_{k+1} is an approximate first-order solution of* (3.6) *with the choice $y^E = y_k^E$ and $\mu = \mu_k^R$ that satisfies* (3.19)*.*

Proof. If there exists a subsequence of $\{\|r_{\text{opt}}(v_k)\|\}_{k \geq 0}$ that converges to zero, then clearly case (i) holds. Therefore, for the remainder of the proof, it is assumed that the sequence $\{\|r_{\text{opt}}(v_k)\|\}_{k \geq 0}$ is bounded away from zero.

From the definitions of a V-iterate and O-iterate, the functions ϕ_V and ϕ_O, and the update strategies for $\phi_V^{\max}$ and $\phi_O^{\max}$, we conclude that the number of V-iterates and O-iterates must be finite. We claim that there must be an infinite number of M-iterates. To prove this, assume to the contrary that the number of M-iterates is finite, so that all iterates are F-iterates for k sufficiently large. It follows from the form of the update to μ_k^R (3.21) and the assumption made in this case,

that eventually μ_k^R remains constant. In this case, the update to μ_k given by (3.23) implies that eventually, μ_k also remains constant. These arguments imply the existence of an integer $\widehat{k}$ such that

$$\mu_k^R \equiv \mu^R \leq \mu \equiv \mu_k, \;\; y_k^E \equiv y^E, \;\; \tau_k \equiv \tau > 0, \;\; \text{and } k \text{ is an F-iterate for all } k \geq \widehat{k}.$$

It follows from (3.23) that

$$M(v_{k+1}\,; y^E, \mu) \leq M(v_k\,; y^E, \mu) + \min(\alpha_{\min}, \alpha_k)\eta_S \delta_k \;\; \text{for all } k \geq \widehat{k}, \tag{6.4}$$

where δ_k is defined by (3.16). Moreover, parts (ii) and (iii) of Theorem 6.2.1 ensure that $\{\delta_k\}_{k\geq\widehat{k}}$ is a negative sequence bounded away from zero. In addition, it must hold that $\{\alpha_k\}_{k\geq\widehat{k}}$ is bounded away from zero. To see this, note that parts (i) and (iii) of Theorem 6.2.1 and Assumption 6.2.2 ensure that $\{\alpha_k\}_{k\geq\widehat{k}}$ is bounded away from zero if a conventional Armijo line search is used, i.e., if $\mu_k^F = \mu^R$ and $\delta_k = d_k^{\mathrm{T}} \nabla M(v_k\,; y^E, \mu^R)$ in (3.15). However, the computed value of α_k can be no smaller because the definition of δ_k is less restrictive, and the use of a flexible line search makes the acceptance of a step more likely. Combining these results with (6.4), yields

$$M(v_{k+1}\,; y^E, \mu) \leq M(v_k\,; y^E, \mu) - \kappa \;\; \text{for all } k \geq \widehat{k} \text{ and some } \kappa > 0,$$

so that $\lim_{k\to\infty} M(v_k\,; y^E, \mu) = -\infty$. However, Assumptions 6.2.2 and 6.2.3 ensure that this is not possible. This contradiction implies that there must exist infinitely many M-iterations, and *every* iterate is an M-iterate or F-iterate for k sufficiently large. Part (ii) now follows from (3.21) and the properties of the updates to τ_k and y_k^E used for M-iterates and F-iterates in Algorithm 3. $\square$

6.3 Numerical Results

Results were obtained in order to measure the relative performance of `pdSQP` and `dcpdSQP` on a collection of optimization problems from the CUTEst benchmarking suite. The runs were done using MATLAB version R2023b on an iMac Pro with a 3.0 GHz Intel Xeon W processor and 128 GB of 800 MHz DDR4 RAM running macOS, version 14.4.1 (64 bit). Results were obtained for six subsets of problems from the CUTEst test collection. The subsets consisted of 139 bound constrained (BC) problems with a general nonlinear objective and upper and lower bounds on the variables; 262 feasible-point (FP) problems with no objective, general linear and nonlinear constraints and bounds on the variables; 126 problems formulated by Hock and Schittkowski ([50]) (HS); 212 linearly constrained (LC) problems with a general nonlinear objective, general linear constraints and bounds on the variables; 386 nonlinearly constrained (NC) problems with a general nonlinear objective, general linear and nonlinear constraints and bounds on the variables; and 173 unconstrained (UC) problems with a general nonlinear objective and no constraints. In total, these subsets contain 1172 test problems.

The BC, FP, HS, LC, NC and UC subsets were selected based on the number of variables and general constraints. In particular, a problem was chosen if the associated KKT system was of the order of 3000 or less. The same criterion was used to set the dimension of those problems for which the problem size can be specified. Exact second derivatives were used for all the runs.

6.3.1 Implementation

Both `pdSQP` and `dcpdSQP` were implemented in MATLAB version R2023b. The difference between the two is in the convexification strategy. The base algorithm `pdSQP` does a full convexification using the method of Wächter and Biegler [68], while the dynamically convexified `dcpdSQP`

uses Algorithms 3 and 2. Both MATLAB implementations were initialized with identical parameter values that were chosen based on the empirical performance on the entire collection of problems. A summary of the values is given in Table 6.1. The initial primal-dual estimate (x_0, y_0) was based on the default initial values supplied by CUTEst.

There are three scenarios that are considered to represent the successful solution of a problem. The first two of these scenarios correspond to the two outcomes presented in Theorem 6.2.2, while the third is the recognition of an unbounded problem. The first scenario is convergence to a first-order solution of (NP), characterized by

$$\|r_{\text{opt}}(x,y)\| \leq \tau_{\text{opt}} \qquad \text{where} \qquad r_{\text{opt}}(x,y) = \begin{pmatrix} c(x) \\ \min(x, g(x) - J(x)^{\mathrm{T}} y) \end{pmatrix}. \tag{6.5}$$

The second is convergence to an infeasible stationary point (x, y), where

$$\begin{aligned} &(x,y) \ \text{ satisfies } \ (3.19), \ \text{ and} \\ &\|r_{\text{inf}}(x,y)\| \leq \tau_{\text{inf}} \qquad \text{where} \qquad r_{\text{inf}}(x,y) = \min(x, \ \max(0, \ J(x)^{\mathrm{T}} c(x))). \end{aligned} \tag{6.6}$$

Lastly, the problem (NP) is declared unbounded if

$$\begin{aligned} &f(x) \leq f_{\text{unb}}, \ \text{ and} \\ &\|r_{P}(x)\|_{\infty} \leq \tau_{P} \qquad \text{where} \qquad r_{P}(x) = \begin{pmatrix} c(x) \\ \min(0, x) \end{pmatrix}. \end{aligned} \tag{6.7}$$

The iterates were terminated at the first point satisfying either (6.5), (6.6), or (6.7).

Table 6.1: Control parameters for Algorithms `pdSQP` and `dcpdSQP`.

Parameter	Description	Value
$y_{\max}$	Maximum allowed y^E (3.20)	`1.0e+6`
μ_0^R	Initial regularization parameter for Algorithm (3)	`1.0e-6`
μ_0^L	Initial flexible line-search penalty parameter for Algorithm (3)	`1.0`
$\mu_{\min}^L$	Minimum allowed μ^L	`1.0e-14`
$\mu_{\min}^R$	Minimum allowed μ^R	`1.0e-14`
$d_{\max}$	Maximum allowed $\|d\|$ (5.50)	`1.0e+2`
τ_{opt}	Optimality tolerance (6.5)	`1.0e-4`
τ_{inf}	Infeasible stationary point tolerance (6.6)	`1.0e-5`
τ_P	Primal feasibility tolerance (6.7)	`1.0e-4`
τ_D	Dual feasibility tolerance (5.49)	`1.0e-6`
$\lambda_{\min}$	Minimum allowed positive eigenvalue	`1.0e-8`
η_S	Line-search backtracking sufficient decrease (3.15)	`1.0e-3`
η_D	Line-search factor in backtracking acceptance test (3.16)	`1.0e-3`
γ_C	Line-search backtracking contraction factor	`0.5`
f_{unb}	Unbounded objective (6.7)	`-1.0e+9`
$k_{\max}$	Iteration limit for Algorithm (3)	`750`

6.3.2 Performance profiles

The relative performance of the solvers is summarized using performance profiles (in $\log_2$ scale), which were proposed by Dolan and Moré [17]. Let $\mathcal{P}$ denote a set of problems used for a given numerical experiment. For each method s we define the function $\rho_s : [0, r_M] \mapsto \mathbb{R}^+$ such that

$$\rho_s(\tau) = \frac{1}{n_{\mathcal{P}}} \left| \left\{ p \in \mathcal{P} : \log_2(r_{p,s}) \leq \tau \right\} \right|,$$

where $n_{\mathcal{P}}$ is the number of problems in the test set and $r_{p,s}$ denotes the ratio of the performance metric (for example, the total number of function evaluations) needed to solve problem p with

method s and the least value of the performance metric needed to solve problem p. If method s failed for problem p, then $r_{p,s}$ is set to be twice of the maximal ratio. The parameter r_M is the maximum value of $\log_2(r_{p,s})$.

Note that for Figures 6.1–6.7 and Tables 6.2–6.7 `dcpdSQP` uses the method of Wächter and Biegler [68] for pre-convexification. In Figures 6.8–6.14, the different pre-convexification schemes are compared, in which case `dcpdSQP` is labeled `dcpdSQP-WB` to emphasize the pre-convexification method used. The method labeled `dcpdSQP-ic` uses the inertia-controlling symmetric indefinite factorization of Forsgren [27], and `dcpdSQP-2stage` does the two-stage factorization presented in Section 4.1.1. As always, `pdSQP` performs a full convexification. It should also be noted that the performance of `pdSQP` and `dcpdSQP` are *identical* on unconstrained problems because all variables are free and therefore pre-convexification is equivalent to full convexification. This is the reason that the UC set is not included in the combined ALL set, and that there are no figures or profiles comparing `pdSQP` and `dcpdSQP` on the UC problem set. However, the different methods of pre-convexification do differ on unconstrained problems, which can be seen in Figure 6.14, where it is also confirmed that `pdSQP` and `dcpdSQP` perform identically.

It is important to keep in mind that the purpose of dynamic convexification is to improve computation efficiency of convexification in a way that is scalable to large-scale, sparse problems. As `pdSQP` and `dcpdSQP` are based on the same primal-dual merit function SQP algorithm, the hope is that dynamic convexification will not adversely affect the performance while significantly reducing the cost associated with computing convex approximations. It is for this reason the chosen performance metrics are function evaluations, iterations, and factorizations. In particular, function evaluations and iterations measure the performance of a given method while the number of factorizations required is a measure of efficiency directly related to the convexification method

used.

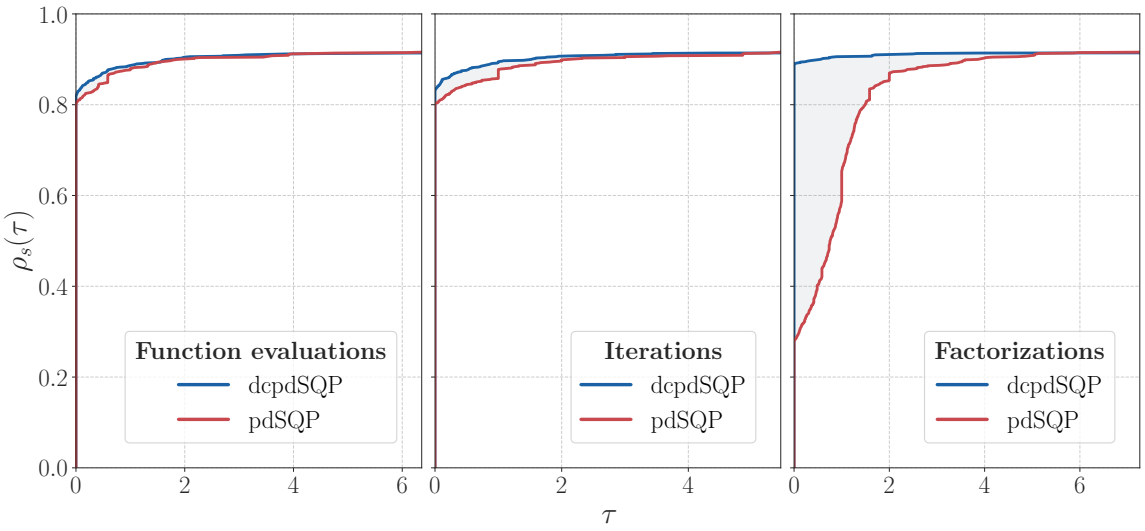

Figure 6.1: Performance profiles comparing function evaluations, iterations, and factorizations used by the algorithms `dcpdSQP` and `pdSQP` when applied to 999 problems from the combined (ALL) CUTEst test set. The (ALL) set is the union of the (BC), (FP), (HS), (LC), and (NC) test sets.

It is apparent from Figure 6.1 that, overall, the performance of `dcpdSQP` is comparable to `pdSQP` in terms of function evaluations and iterations. Profiles for these metrics on the FP, NC, and HS sets are nearly indistinguishable (Figures 6.3, 6.7, and 6.5), while a modest advantage from dynamic convexification is shown on the BC and LC sets (Figures 6.2 and 6.6). This advantage may be due to the fact that dynamic convexification usually computes a smaller norm perturbation compared to a full convexification, so that the Hessian used approximates the true Hessian of the Lagrangian function more faithfully. The most striking feature of these numerical results, however, is seen in the factorization profiles. The near-horizontal curve for `dcpdSQP` indicates that the proportion of problems for which `dcpdSQP` used the fewest factorizations is nearly the same as the total number of problems solved. In other words `dcpdSQP` used fewer factorizations on almost every problem it solved successfully. Together, these profiles demonstrate that dynamic convexification

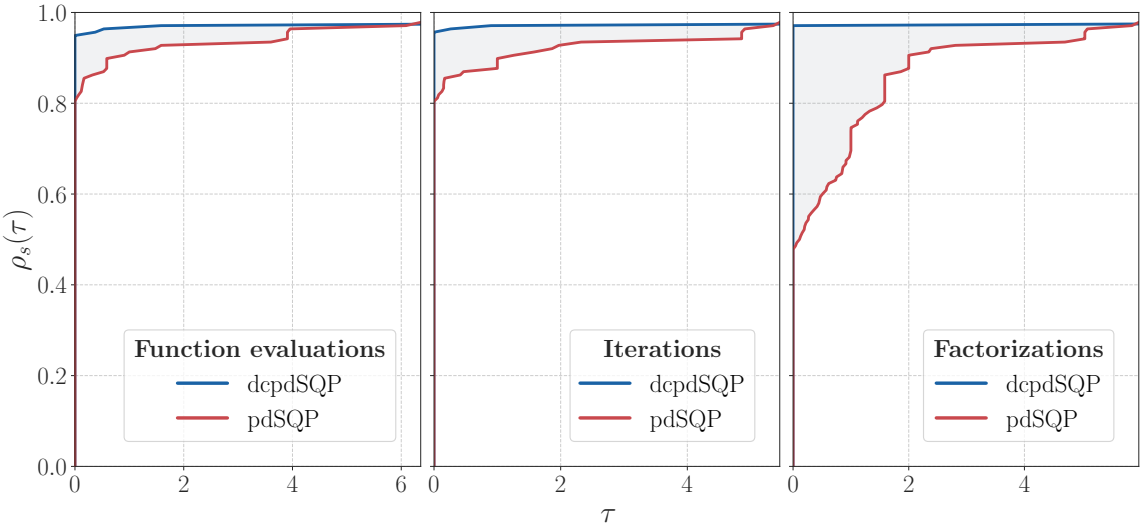

Figure 6.2: Performance profiles comparing function evaluations, iterations, and factorizations used by the algorithms `dcpdSQP` and `pdSQP` when applied to 139 bound constrained (BC) problems from the CUTEst test set.

achieves its intended purpose of dramatically reducing the computational cost of convexification while matching or improving overall performance.

For the feasible-point problems, which have no objective function, it is generally beneficial to use an artificial objective function $f(x) = \frac{1}{2}\|x\|^2$. This objective regularizes the problem in the sense that most feasible-point problems have infinitely many solutions, and the inclusion of $f(x)$ targets a specific one. The profiles in Figure 6.4 are made with no objective function, while the profiles in Figure 6.3, as well as the data in Table 6.4, were generated using the suggested artificial objective function.

Tables 6.2–6.7 give details of the outcomes for `pdSQP` and `dcpdSQP` on each problem set. The UC set is not included because the methods perform identically on an unconstrained function. Also note that although almost all the problems in the HS set are included in the NC set, these problems are not duplicated in the combined ALL problem set. Each sub-table lists, for both solvers,

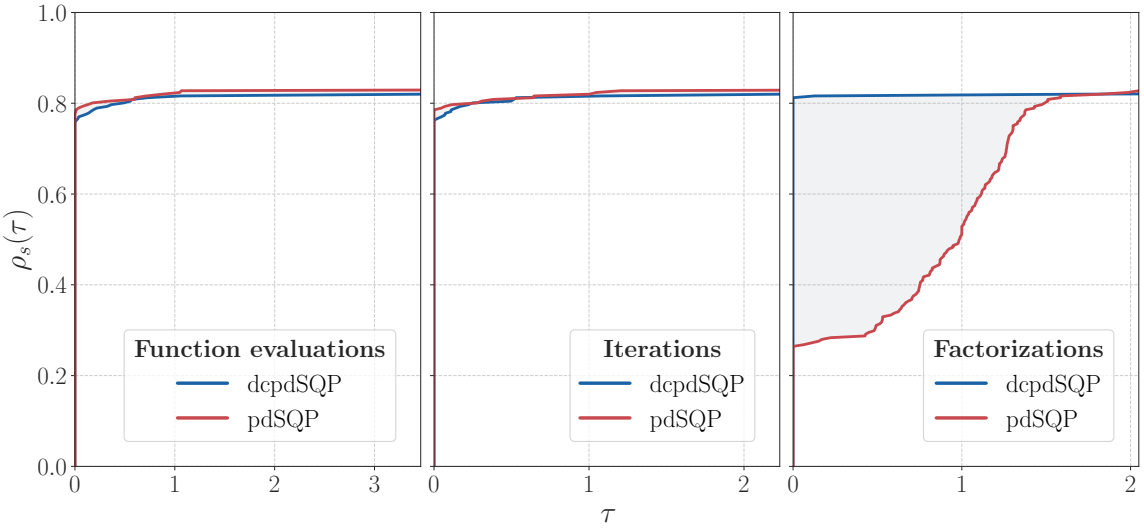

Figure 6.3: Performance profiles comparing function evaluations, iterations, and factorizations used by the algorithms `dcpdSQP` and `pdSQP` when applied to 262 feasible-point (FP) problems *with an artificial objective function* from the CUTEst test set.

the number of problems in the given problem set for which each outcome was achieved. Of the runs that fail, `dcpdSQP` is unable to convexify the QP problems more often, and `pdSQP` terminates more often with the iteration limit exceeded and line-search failure. Of the runs that succeed, `dcpdSQP` finds more local minimizers and `pdSQP` finds more infeasible stationary points. Overall, Table 6.2 supports the claim that the performance with dynamic convexification is comparable to or better than that of the base algorithm.

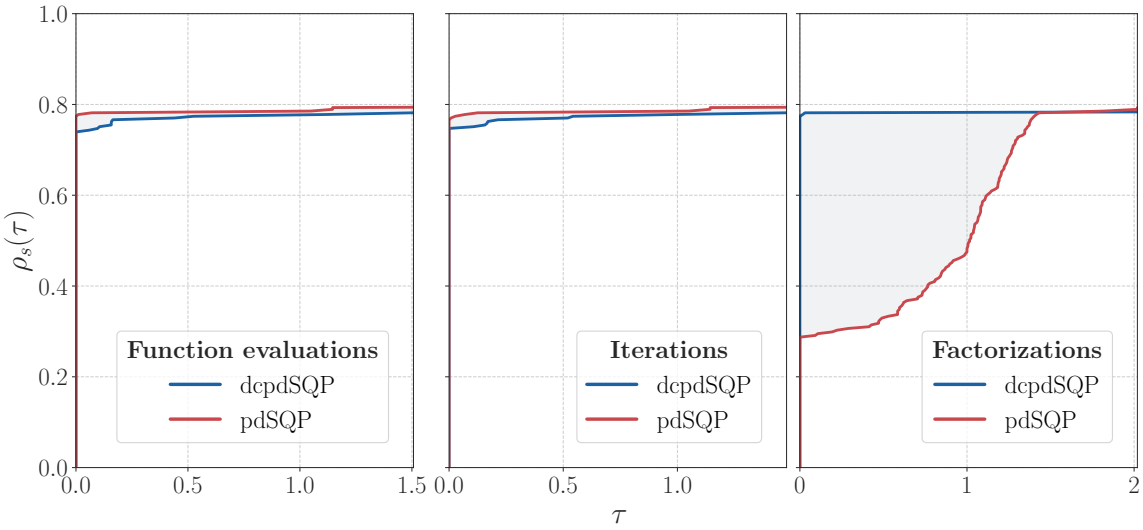

Figure 6.4: Performance profiles comparing function evaluations, iterations, and factorizations used by the algorithms `dcpdSQP` and `pdSQP` when applied to 262 feasible-point (FP) problems *with no objective function* from the CUTEst test set.

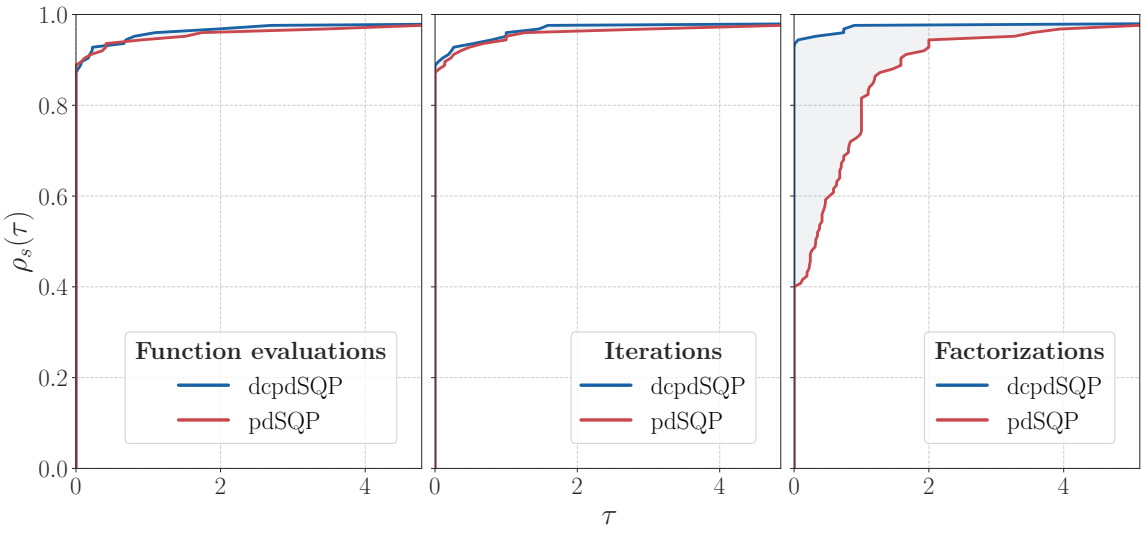

Figure 6.5: Performance profiles comparing function evaluations, iterations, and factorizations used by the algorithms `dcpdSQP` and `pdSQP` when applied to 126 Hock-Shittkowski (HS) problems from the CUTEst test set.

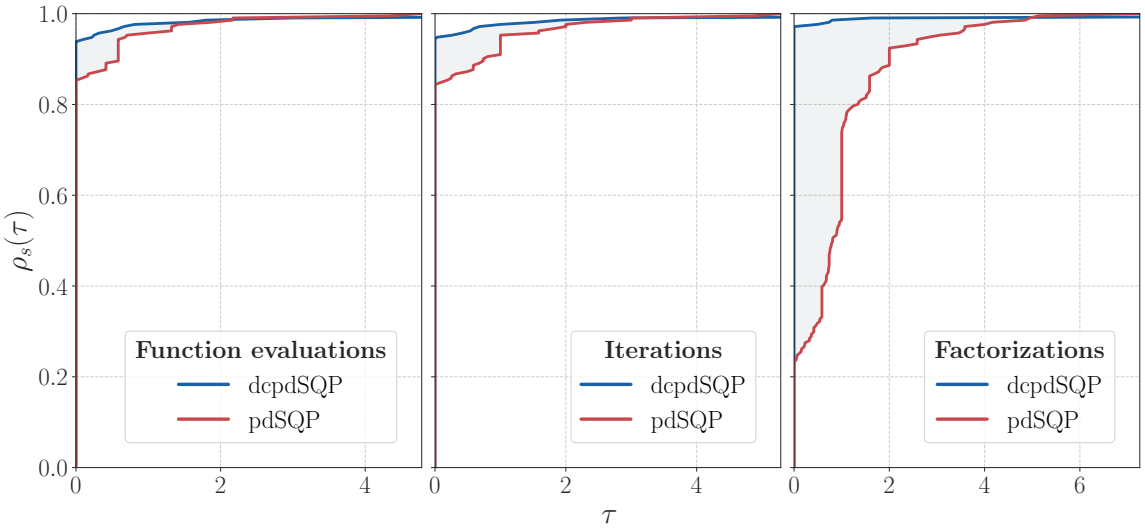

Figure 6.6: Performance profiles comparing function evaluations, iterations, and factorizations used by the algorithms `dcpdSQP` and `pdSQP` when applied to 212 linearly constrained (LC) problems from the CUTEst test set.

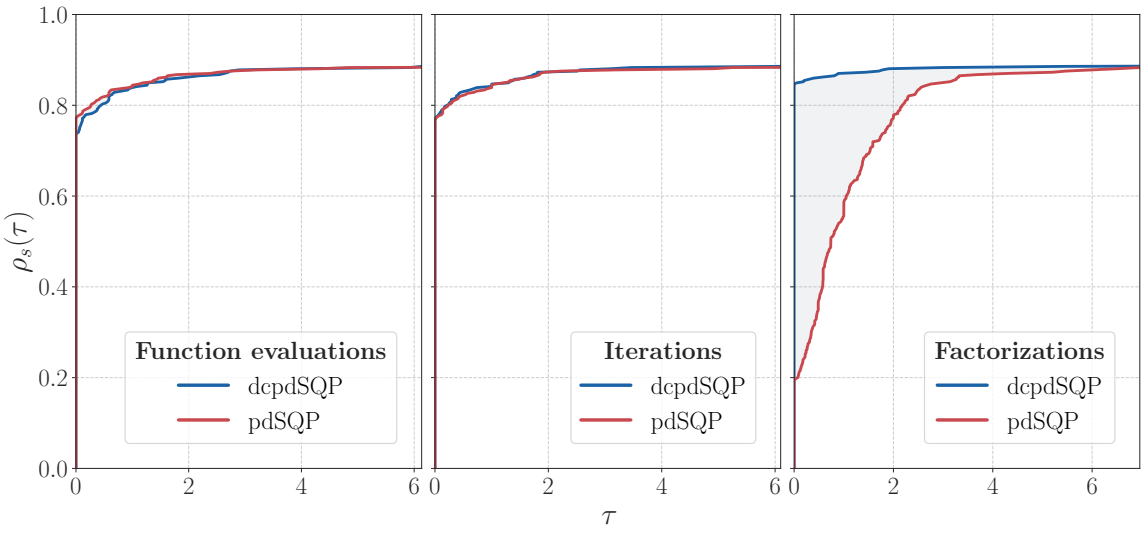

Figure 6.7: Performance profiles comparing function evaluations, iterations, and factorizations used by the algorithms `dcpdSQP` and `pdSQP` when applied to 386 nonlinearly constrained (NC) problems from the CUTEst test set.

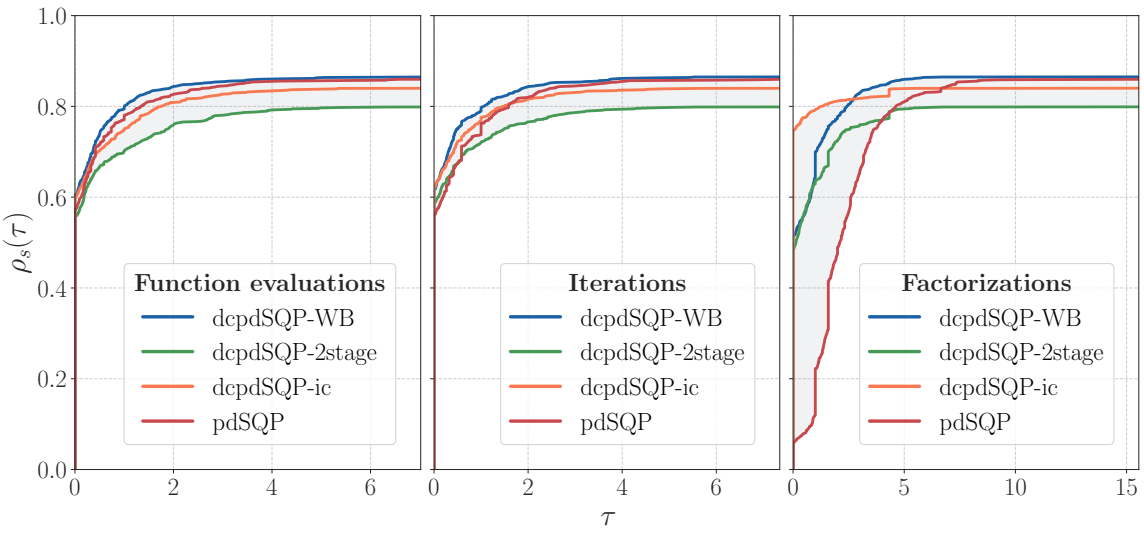

Figure 6.8: Performance profiles comparing pre-convexification methods in `dcpdSQP` with `pdSQP` when applied to 999 problems from the combined (ALL) CUTEst test set.

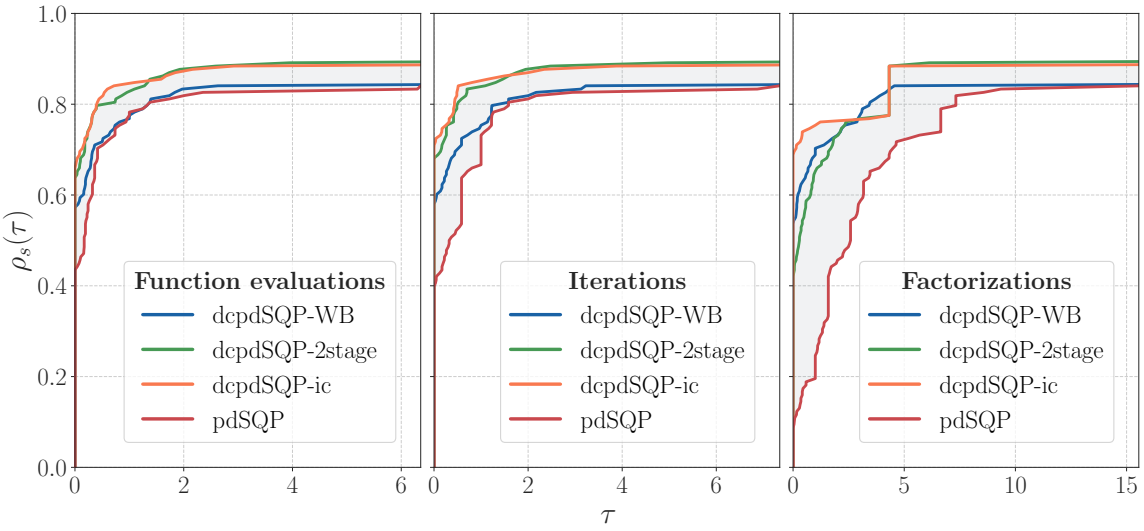

Figure 6.9: Performance profiles comparing pre-convexification methods in `dcpdSQP` with `pdSQP` when applied to 139 bound constrained (BC) problems from the CUTEst test set.

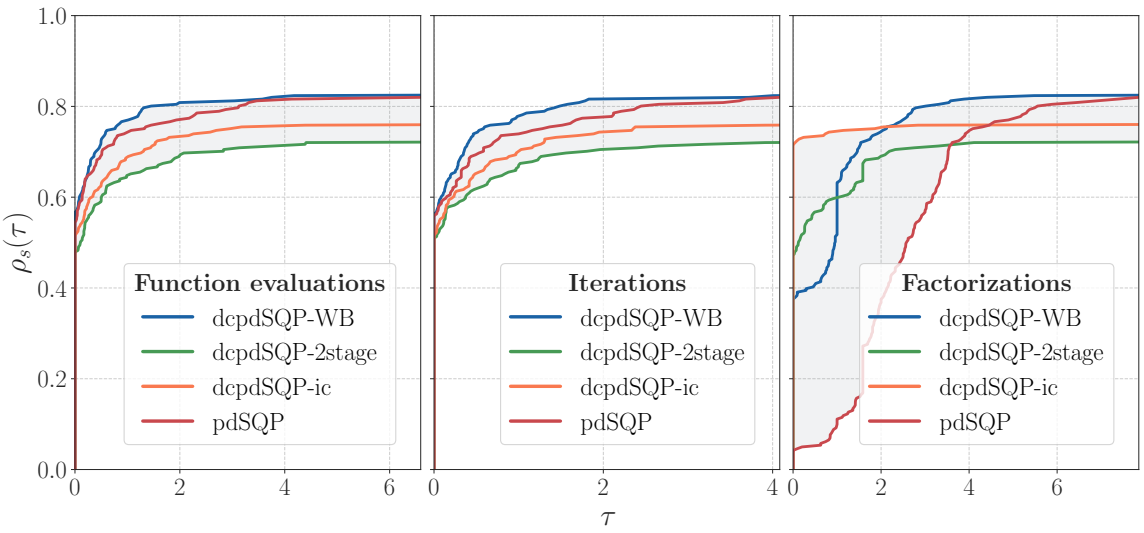

Figure 6.10: Performance profiles comparing pre-convexification methods in `dcpdSQP` with `pdSQP` when applied to 262 feasible-point (FP) problems from the CUTEst test set.

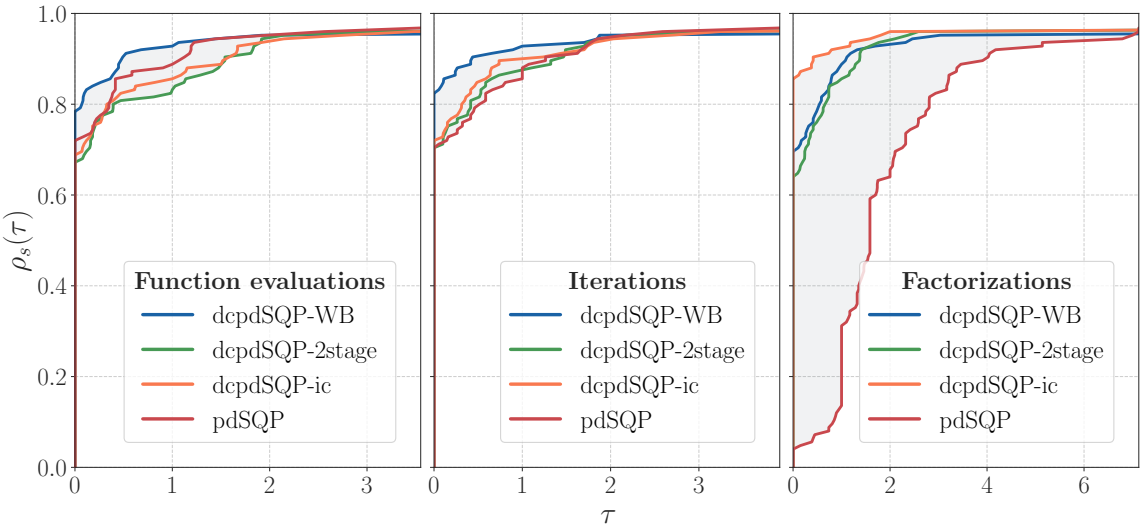

Figure 6.11: Performance profiles comparing pre-convexification methods in `dcpdSQP` with `pdSQP` when applied to 126 Hock-Shittkowski (HS) problems from the CUTEst test set.

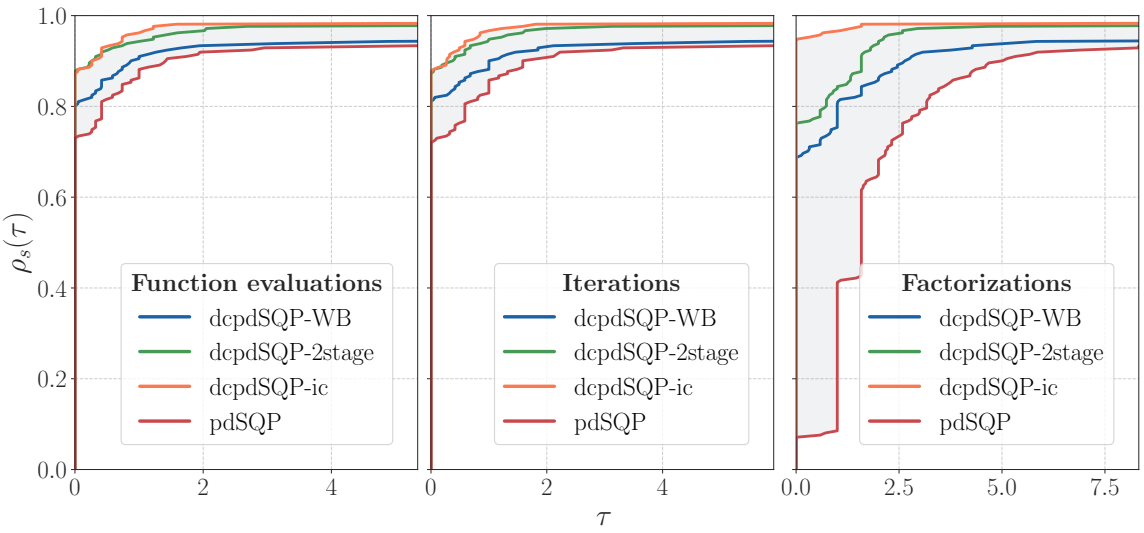

Figure 6.12: Performance profiles comparing pre-convexification methods in `dcpdSQP` with `pdSQP` when applied to 212 linearly constrained (LC) problems from the CUTEst test set.

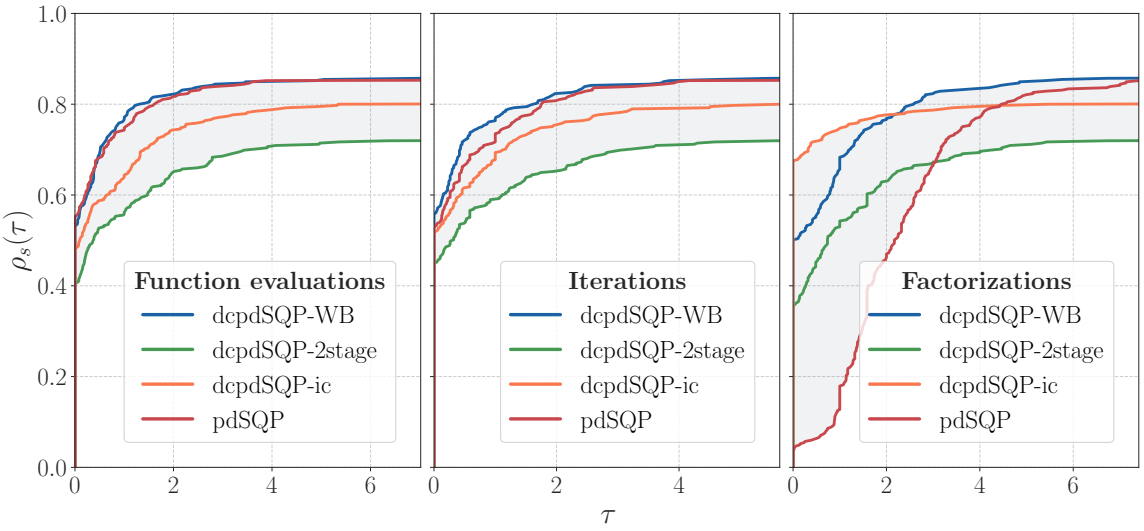

Figure 6.13: Performance profiles comparing pre-convexification methods in `dcpdSQP` with `pdSQP` when applied to 386 nonlinearly constrained (NC) problems from the CUTEst test set.

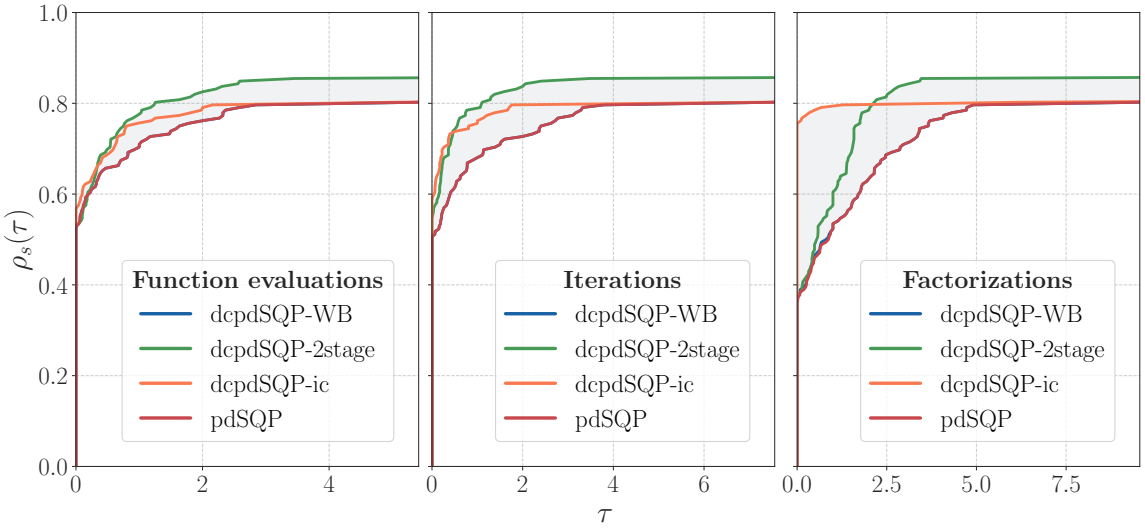

Figure 6.14: Performance profiles comparing pre-convexification methods in `dcpdSQP` with `pdSQP` when applied to 173 unconstrained (UC) problems from the CUTEst test set.

Table 6.2: Problem set (ALL) outcome counts.

Outcome	pdSQP	dcpdSQP
Optimal	784	792
Infeasible stationary point	126	115
Near optimal	10	13
Iteration limit	56	58
Line-search failure	15	5
QP convexification failure	3	10
Unbounded problem	5	6

Table 6.3: Problem set (BC) outcome counts.

Outcome	pdSQP	dcpdSQP
Optimal	136	135
Infeasible stationary point	-	-
Near optimal	1	1
Iteration limit	2	2
Line-search failure	-	-
QP convexification failure	-	1
Unbounded problem	-	-

Table 6.4: Problem set (FP) outcome counts.

Outcome	pdSQP	dcpdSQP
Optimal	113	113
Infeasible stationary point	104	102
Near optimal	4	3
Iteration limit	30	38
Line-search failure	10	4
QP convexification failure	1	2
Unbounded problem	-	-

Table 6.5: Problem set (HS) outcome counts.

Outcome	pdSQP	dcpdSQP
Optimal	122	122
Infeasible stationary point	1	1
Near optimal	1	1
Iteration limit	1	1
Line-search failure	1	1
QP convexification failure	-	-
Unbounded problem	-	-

Table 6.6: Problem set (LC) outcome counts.

Outcome	pdSQP	dcpdSQP
Optimal	207	207
Infeasible stationary point	4	2
Near optimal	-	-
Iteration limit	-	-
Line-search failure	-	-
QP convexification failure	-	2
Unbounded problem	1	1

Table 6.7: Problem set (NC) outcome counts.

Outcome	pdSQP	dcpdSQP
Optimal	328	337
Infeasible stationary point	18	11
Near optimal	5	9
Iteration limit	24	17
Line-search failure	5	1
QP convexification failure	2	6
Unbounded problem	4	5

Bibliography

[1] Patrick R. Amestoy, Iain S. Duff, Jean-Yves L'Excellent, and Jacko Koster. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM J. Matrix Anal. Appl.*, 23(1):15–41 (electronic), 2001.

[2] Mihai Anitescu. A superlinearly convergent sequential quadratically constrained quadratic programming algorithm for degenerate nonlinear programming. *SIAM J. Optim.*, 12(4):949–978, 2002.

[3] Cleve Ashcraft and Roger Grimes. SPOOLES: an object-oriented sparse matrix library. In *Proceedings of the Ninth SIAM Conference on Parallel Processing for Scientific Computing 1999 (San Antonio, TX)*, page 10, Philadelphia, PA, 1999. SIAM.

[4] Jonathan M. Borwein. Necessary and sufficient conditions for quadratic minimality. *Numer. Funct. Anal. and Optimiz.*, 5:127–140, 1982.

[5] James R. Bunch. Partial pivoting strategies for symmetric matrices. *SIAM J. Numer. Anal.*, 11:521–528, 1974.

[6] James R. Bunch and Linda Kaufman. Some stable methods for calculating inertia and solving symmetric linear systems. *Math. Comp.*, 31:163–179, 1977.

[7] James R. Bunch and Beresford N. Parlett. Direct methods for solving symmetric indefinite systems of linear equations. *SIAM J. Numer. Anal.*, 8:639–655, 1971.

[8] Richard Byrd, Jorge Nocedal, Richard Waltz, and Yuchen Wu. On the use of piecewise linear models in nonlinear programming. *Math. Program.*, pages 1–36, 2010. 10.1007/s10107-011-0492-9.

[9] Thomas F. Coleman and Alex Pothen. The null space problem I. Complexity. *SIAM J. on Algebraic and Discrete Methods*, 7:527–537, 1986.

[10] Thomas F. Coleman and Danny C. Sorensen. A note on the computation of an orthogonal basis for the null space of a matrix. *Math. Program.*, 29:234–242, 1984.

[11] Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. Global convergence of a class of trust region algorithms for optimization with simple bounds. *SIAM J. Numer. Anal.*, 25:433–460, 1988.

[12] Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. A comprehensive description of LANCELOT. Technical Report 91/10, Département de Mathématique, Facultés Universitaires de Namur, 1991.

[13] Andrew R. Conn, Nicholas I. M. Gould, and Philippe L. Toint. A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds. *SIAM J. Numer. Anal.*, 28:545–572, 1991.

[14] Luis B. Contesse. Une caractérisation complète des minima locaux en programmation quadratique. *Numer. Math.*, 34:315–332, 1980.

[15] Richard W. Cottle, G. J. Habetler, and C. E. Lemke. On classes of copositive matrices. *Linear Algebra Appl.*, 3:295–310, 1970.

[16] Frank E. Curtis and Jorge Nocedal. Flexible penalty functions for nonlinear constrained optimization. *IMA J. Numer. Anal.*, 28(4):749–769, 2008.

[17] Elizabeth D. Dolan and Jorge J. Moré. Benchmarking optimization software with performance profiles. *Math. Program.*, 91(2, Ser. A):201–213, 2002.

[18] Iain S. Duff. MA57—a code for the solution of sparse symmetric definite and indefinite systems. *ACM Trans. Math. Software*, 30(2):118–144, 2004.

[19] Iain S. Duff and John K. Reid. MA27: a set of Fortran subroutines for solving sparse symmetric sets of linear equations. Technical Report R-10533, Computer Science and Systems Division, AERE Harwell, Oxford, England, 1982.

[20] Damián Fernández and Mikhail Solodov. Stabilized sequential quadratic programming for optimization and a stabilized Newton-type method for variational problems. *Math. Program. Ser. A*, 125:47–73, 2010.

[21] Andreas Fischer. Modified Wilson's method for nonlinear programs with nonunique multipliers. *Math. Oper. Res.*, 24(3):699–727, 1999.

[22] Roger Fletcher. A general quadratic programming algorithm. *J. Inst. Math. Applics.*, 7:76–91, 1971.

[23] Roger Fletcher. An ℓ_1 penalty method for nonlinear constraints. In Paul T. Boggs, Richard H. Byrd, and Robert B. Schnabel, editors, *Numerical Optimization 1984*, pages 26–40, Philadelphia, 1985. SIAM.

[24] Roger Fletcher and Sven Leyffer. User manual for filterSQP. Technical Report NA/181, Dept. of Mathematics, University of Dundee, Scotland, 1998.

[25] Roger Fletcher and Sven Leyffer. Nonlinear programming without a penalty function. *Math. Program.*, 91(2, Ser. A):239–269, 2002.

[26] Roger Fletcher, Sven Leyffer, and Philippe L. Toint. On the global convergence of a filter-SQP algorithm. *SIAM J. Optim.*, 13(1):44–59 (electronic), 2002.

[27] Anders Forsgren. Inertia-controlling factorizations for optimization algorithms. *Appl. Numer. Math.*, 43:91–107, 2002.

[28] Anders Forsgren and Philip E. Gill. Primal-dual interior methods for nonconvex nonlinear programming. *SIAM J. Optim.*, 8:1132–1152, 1998.

[29] Anders Forsgren, Philip E. Gill, and Walter Murray. On the identification of local minimizers in inertia-controlling methods for quadratic programming. *SIAM J. Matrix Anal. Appl.*, 12:730–746, 1991.

[30] Anders Forsgren and Walter Murray. Newton methods for large-scale linear equality-constrained minimization. *SIAM J. Matrix Anal. Appl.*, 14:560–587, 1993.

[31] John R. Gilbert and Michael T. Heath. Computing a sparse basis for the null space. Report TR86-730, Department of Computer Science, Cornell University, 1986.

[32] Philip E. Gill, Vyacheslav Kungurtsev, and Daniel P. Robinson. A stabilized SQP method: Global convergence. *IMA J. Numer. Anal.*, 37(1):407–443, 05 2017.

[33] Philip E. Gill, Vyacheslav Kungurtsev, and Daniel P. Robinson. A stabilized SQP method: superlinear convergence. *Mathematical Programming*, 163(1):369–410, 2017.

[34] Philip E. Gill and Walter Murray. Newton-type methods for unconstrained and linearly constrained optimization. *Math. Program.*, 7:311–350, 1974.

[35] Philip E. Gill, Walter Murray, and Michael A. Saunders. SNOPT: An SQP algorithm for large-scale constrained optimization. *SIAM Rev.*, 47:99–131, 2005.

[36] Philip E. Gill, Walter Murray, Michael A. Saunders, Gilbert (Pete) W. Stewart, and Margaret H. Wright. Properties of a representation of a basis for the null space. *Math. Programming*, 33(2):172–186, 1985.

[37] Philip E. Gill, Walter Murray, Michael A. Saunders, and Margaret H. Wright. Shifted barrier methods for linear programming. Report SOL 87-9, Department of Operations Research, Stanford University, Stanford, CA, 1987.

[38] Philip E. Gill, Walter Murray, Michael A. Saunders, and Margaret H. Wright. Inertia-controlling methods for general quadratic programming. *SIAM Rev.*, 33(1):1–36, 1991.

[39] Philip E. Gill and Daniel P. Robinson. A primal-dual augmented Lagrangian. *Comput. Optim. Appl.*, 51:1–25, Jan 2012.

[40] Philip E. Gill and Daniel P. Robinson. A globally convergent stabilized SQP method. *SIAM J. Optim.*, 23(4):1983–2010, 2013.

[41] Philip E. Gill and Elizabeth Wong. Sequential quadratic programming methods. In Jon Lee and Sven Leyffer, editors, *Mixed Integer Nonlinear Programming*, volume 154 of *The IMA Volumes in Mathematics and its Applications*, pages 147–224. Springer New York, 2012.

[42] Philip E. Gill and Elizabeth Wong. Methods for convex and general quadratic programming. Center for Computational Mathematics Report CCoM 13-1, University of California, San Diego, La Jolla, CA, 2013.

[43] Philip E. Gill and Elizabeth Wong. Methods for convex and general quadratic programming. *Math. Program. Comput.*, 7:71–112, 2015.

[44] Philip E. Gill and Margaret H. Wright. *Computational Optimization: Nonlinear Programming.* Cambridge University Press, New York, NY, USA, 2024. To be published in 2024.

[45] Nicholas I. M. Gould. On practical conditions for the existence and uniqueness of solutions to the general equality quadratic programming problem. *Math. Program.*, 32:90–99, 1985.

[46] Nicholas I. M. Gould. An algorithm for large-scale quadratic programming. *IMA J. Numer. Anal.*, 11(3):299–324, 1991.

[47] Nicholas I. M. Gould and Daniel P. Robinson. A second derivative SQP method: Global convergence. *SIAM J. Optim.*, 20(4):2023–2048, 2010.

[48] John Greenstadt. On the relative efficiencies of gradient methods. *Math. Comput.*, 21:360–367, 1967.

[49] William W. Hager. Stabilized sequential quadratic programming. *Comput. Optim. Appl.*, 12(1-3):253–273, 1999. Computational optimization—a tribute to Olvi Mangasarian, Part I.

[50] W. Hock and K. Schittkowski. *Test Examples for Nonlinear Programming Codes.* Lecture Notes in Econom. Math. Syst. 187. Springer-Verlag, Berlin, 1981.

[51] Vyacheslav Kungurtsev. *Second-Derivative Sequential Quadratic Programming Methods for Nonlinear Optimization.* PhD thesis, Department of Mathematics, University of California San Diego, La Jolla, CA, 2013.

[52] Dong-Hui Li and Liqun Qi. A stabilized SQP method via linear equations. Technical Report AMR00/5, School of Mathematics, University of New South Wales, Sydney, 2000.

[53] Antal Majthay. Optimality conditions for quadratic programming. *Math. Programming*, 1:359–365, 1971.

[54] Olvi L. Mangasarian and Stanley Fromovitz. The Fritz John necessary optimality conditions in the presence of equality and inequality constraints. *J. Math. Anal. Appl.*, 17:37–47, 1967.

[55] Nicholas Maratos. *Exact Penalty Function Algorithms for Finite-Dimensional and Control Optimization Problems.* PhD thesis, Department of Computing and Control, Imperial College, University of London, 1978.

[56] Jorge J. Moré and Danny C. Sorensen. Newton's method. In Gene H. Golub, editor, *Studies in Mathematics, Volume 24. MAA Studies in Numerical Analysis*, pages 29–82. Math. Assoc. America, Washington, DC, 1984.

[57] Jorge Nocedal and Stephen J. Wright. *Numerical Optimization.* Springer-Verlag, New York, 1999.

[58] James M. Ortega and Werner C. Rheinboldt. *Iterative solution of nonlinear equations in several variables.* Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2000. Reprint of the 1970 original.

[59] Panos M. Pardalos and Georg Schnitger. Checking local optimality in constrained quadratic programming is NP-hard. *Oper. Res. Lett.*, 7(1):33–35, 1988.

[60] Panos M. Pardalos and Stephen A. Vavasis. Quadratic programming with one negative eigenvalue is NP-hard. *J. Global Optim.*, 1(1):15–22, 1991.

[61] Michael J. D. Powell. A method for nonlinear constraints in minimization problems. In Roger Fletcher, editor, *Optimization*, pages 283–298, London and New York, 1969. Academic Press.

[62] Daniel P. Robinson. *Primal-Dual Methods for Nonlinear Optimization*. PhD thesis, Department of Mathematics, University of California San Diego, La Jolla, CA, 2007.

[63] Stephen M. Robinson. A quadratically-convergent algorithm for general nonlinear programming problems. *Math. Program.*, 3:145–156, 1972.

[64] Stephen M. Robinson. Perturbed Kuhn-Tucker points and rates of convergence for a class of nonlinear programming algorithms. *Math. Program.*, 7:1–16, 1974.

[65] Olaf Schenk and Klaus Gärtner. Solving unsymmetric sparse systems of linear equations with PARDISO. In *Computational Science—ICCS 2002, Part II (Amsterdam)*, volume 2330 of *Lecture Notes in Comput. Sci.*, pages 355–363. Springer, Berlin, 2002.

[66] Robert B. Schnabel and Elizabeth Eskow. A new modified Cholesky factorization. *SIAM J. Sci. and Statist. Comput.*, 11:1136–1158, 1990.

[67] Gerard Van der Hoek. Asymptotic properties of reduction methods applying linearly equality constrained reduced problems. *Math. Program.*, 16:162–189, 1982.

[68] Andreas Wächter, Lorenz T. Biegler, Yi-Dong Lang, and Arvind Raghunathan. IPOPT: An interior point algorithm for large-scale nonlinear optimization. https://projects.coin-or.org/Ipopt, 2002.

[69] Robert B. Wilson. *A Simplicial Method for Convex Programming*. PhD thesis, Harvard University, 1963.

[70] Stephen J. Wright. Superlinear convergence of a stabilized SQP method to a degenerate solution. *Comput. Optim. Appl.*, 11(3):253–275, 1998.

[71] Stephen J. Wright. Modifying SQP for degenerate problems. *SIAM J. Optim.*, 13(2):470–497, 2002.

[72] Stephen J. Wright. An algorithm for degenerate nonlinear programming with rapid local convergence. *SIAM J. Optim.*, 15(3):673–696, 2005.