Operator Splitting Methods for Convex Optimization

Analysis and Implementation

Goran Banjac

St Edmund Hall

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

March 2018

To my family.

Abstract

Convex optimization problems are a class of mathematical problems which arise in numerous applications. Although interior-point methods can in principle solve these problems efficiently, they may become intractable for solving large-scale problems or be unsuitable for real-time embedded applications.

Iterations of *operator splitting methods* are relatively simple and computationally inexpensive, which makes them suitable for these applications. However, some of their known limitations are slow asymptotic convergence, sensitivity to ill-conditioning, and inability to detect infeasible problems.

The aim of this thesis is to better understand operator splitting methods and to develop reliable software tools for convex optimization. The main analytical tool in our investigation of these methods is their characterization as the fixed-point iteration of a nonexpansive operator. The fixed-point theory of nonexpansive operators has been studied for several decades.

By exploiting the properties of such an operator, it is possible to show that the alternating direction method of multipliers (ADMM) can detect infeasible problems. Although ADMM iterates diverge when the problem at hand is unsolvable, the differences between subsequent iterates converge to a constant vector which is also a certificate of primal and/or dual infeasibility. Reliable termination criteria for detecting infeasibility are proposed based on this result.

Similar ideas are used to derive necessary and sufficient conditions for linear (geometric) convergence of an operator splitting method and a bound on the achievable convergence rate. The new bound turns out to be tight for the class of averaged operators.

Next, the OSQP solver is presented. OSQP is a novel general-purpose solver for quadratic programs (QPs) based on ADMM. The solver is very robust, is able to detect infeasible problems, and has been extensively tested on many problem instances from a wide variety of application areas.

Finally, operator splitting methods can also be effective in nonconvex optimization. The developed algorithm significantly outperforms a common approach based on convex relaxation of the original nonconvex problem.

Acknowledgements

I was lucky enough to pursue my doctorate at such a wonderful place as the University of Oxford. Yes, the weather in England could have been better in those 3 years, but the research environment could hardly be any better than the one in Oxford. Many exceptional people have accompanied me in this unique journey. Without their help, support and encouragement the completion of this thesis would not have been possible.

My deepest gratitude goes to Paul Goulart for giving me this opportunity and for being the best advisor *ever*. Paul helped me transform from a young graduate student to an experienced researcher. Having an advisor with such a vast scientific knowledge, a great intuition and clarity in explaining the ideas was amazing! He has also been the master in creating a friendly and informal atmosphere in the group. I thank him for giving me the freedom to pursue new research directions while keeping me in touch with the reality, and for his close guidance throughout my DPhil.

I am especially grateful to Stephen Boyd for hosting me in his group in Stanford. Working with him was a real pleasure and a truly inspirational experience for me. Stephen's passion for teaching and enthusiasm for developing tools that make optimization accessible to everyone is unbelievable. His examples illustrating the vast difference between theoretical and practical performance of optimization methods were extremely helpful for developing the OSQP solver.

My flatmate, officemate and collegemate, Bartolomeo Stellato, deserves a special thanks. He has been my closest friend in Oxford, and I have certainly spent more time with him than with anyone else in the last 3 years. Bartolomeo's tech suggestions and LaTeX templates made my DPhil life easier, and his dedication to the OSQP solver made it the best QP solver in the world. His insistence on simplicity (also with coffee and pizza!) and clarity, as well as his extraordinary presentation skills will always be an inspiration to me. Grazie mille!

Thanks to the EU Marie Curie project TEMPO for the generous financial support. This fellowship programme allowed me to conduct and present my research around the globe. I thank to Moritz Diehl, Tor Arne Johansen, Eric Kerrigan and other professors in the TEMPO family for organizing all the workshops and invited sessions, and to all TEMPO fellows for making these events unforgettable.

I owe my deepest thanks to Kostas Margellos not only for the nice collaboration we had, but also for bringing a new vigor in the group. His friendly and helpful

advices have undoubtedly influenced my career decisions. I would like to thank to Moritz Schulze Darup for making my time in Oxford so fun, and for never giving up teaching me some (very useful!) German expressions. Thanks to all the members of the Control Group for making it such a great place to work.

I am grateful to the many friends who have made my life happy and my stay in Oxford memorable. A special thanks go to Filip Pavetić for staying close in spite of living in distant countries.

I would probably not have ended up in Oxford without the help of my Master thesis advisor, Mato Baotić. I am indebted to him for being so positive and supportive during my DPhil application, and for triggering my interest in the fields of mathematical programming and optimization.

I would also like to express my gratitude and love to the better half of me, Maja. From the bottom of my heart I thank her for her constant love, patience and kindness in all these years.

Last but not least, I would like to thank to my parents Ranko and Gordana, my sister Jelena and my brother Milan for their unconditional love and support, and for always standing by me. I am very happy for having them in my life. Hvala vam na svemu!

Goran Banjac
Oxford, March 2018

Notation

Definitions and inequalities

$A \coloneqq B$	A is defined by B
$A =: B$	B is defined by A
$A \geq B$	element-wise inequality between A and B
$A \succeq B$	$A - B$ is positive semidefinite
$A \preceq B$	$A - B$ is negative semidefinite

Scalar sets

$\mathbb{N}$	the natural numbers $\{0, 1, 2, \ldots\}$
$\mathbb{R}$	the real numbers
$\mathbb{R}_+$ ($\mathbb{R}_{++}$)	the nonnegative (positive) real numbers
$\tilde{\mathbb{R}}$	the extended real numbers: $\tilde{\mathbb{R}} \coloneqq \mathbb{R} \cup \{-\infty, +\infty\}$

Vector and matrix sets

$\mathbb{R}^n$	the set of n-dimensional real vectors
$\mathbb{R}^{m \times n}$	the set of m-by-n real matrices
$\mathbb{S}^n$	the set of n-by-n real symmetric matrices
$\mathbb{S}^n_{++}$ ($\mathbb{S}^n_+$)	the set of n-by-n real symmetric positive (semi)definite matrices
$\mathcal{H}$	real Hilbert space

Sequences

$\{x^k\}_{k \in \mathbb{N}}$	sequence of vectors $\{x^0, x^1, x^2, \ldots\}$
δx^k	difference between subsequent elements of a sequence $\{x^k\}_{k \in \mathbb{N}}$: $\delta x^k \coloneqq x^k - x^{k-1}$

Norms

$\|\cdot\|$	vector norm
$\|x\|_1$	1-norm of a vector x: $\|x\|_1 \coloneqq \sum_{i=1}^{n} \|x_i\|$
$\|x\|_2$	2-norm of a vector x: $\|x\|_2 \coloneqq \sqrt{x^T x}$
$\|x\|_\infty$	∞-norm of a vector x: $\|x\|_\infty \coloneqq \max_i \|x_i\|$
$\|x\|_{[k]}$	largest–k norm, *i.e.* the sum of k elements of largest magnitude in x
$\|X\|_2$	spectral norm of a matrix X

Vectors and matrices

$\mathbf{1}$	vector of ones of appropriate dimension
$\langle x, y \rangle$	inner product of vectors x and y
(x, y)	vertical concatenation of vectors x and y
$x_+ \, (x_-)$	vector obtained by setting negative (positive) elements of x to zero
x_i	the i-th element of a vector x
x_{-i}	the remainder of a vector x when component x_i is removed
$\mathrm{card}(x)$	cardinality of a vector x
A^T	transpose of a matrix A
A_i	the i-th row of a matrix A
I	identity matrix in appropriate space
$\mathrm{vec}(A)$	vector composed by stacking columns of $A \in \mathbb{S}^n$
$\mathrm{mat}(x)$	inverse operator of vec
$\mathrm{diag}(x)$	operator mapping a vector x to a diagonal matrix

Set operations

$A^\perp$	orthogonal complement of a set A
$\overline{A}$	closure of a set A
$A \cup B$	union of sets A and B
$A \cap B$	intersection of sets A and B
$A + B$	Minkowski sum of sets A and B: $A + B \coloneqq \{a + b \,\|\, a \in A,\, b \in B\}$
$A \setminus B$	relative complement of sets A and B: $A \setminus B \coloneqq \{a \,\|\, a \in A \wedge a \notin B\}$

Convex sets and functions

$\operatorname{dist}_{\mathcal{C}}$	Euclidean distance to a set $\mathcal{C}$
$\operatorname{dist}(\mathcal{C}_1, \mathcal{C}_2)$	Euclidean distance between sets $\mathcal{C}_1$ and $\mathcal{C}_2$
$\mathcal{I}_{\mathcal{C}}$	indicator function of a set $\mathcal{C}$
$N_{\mathcal{C}}$	normal cone of a set $\mathcal{C}$
$S_{\mathcal{C}}$	support function of a set $\mathcal{C}$
$\mathcal{C}^{\infty}$	recession cone of a set $\mathcal{C}$
$\operatorname{aff}(\mathcal{C})$	affine hull of a set $\mathcal{C}$
$\operatorname{aff}_0(\mathcal{C})$	subspace obtainable by translation of $\operatorname{aff}(\mathcal{C})$
$\operatorname{relint} \mathcal{C}$	relative interior of a set $\mathcal{C}$
$\mathcal{K}^{\circ}$	polar of a cone $\mathcal{K}$
$\mathcal{K}_b$	translated cone: $\mathcal{K}_b \coloneqq \mathcal{K} + \{b\}$
$\mathcal{F}_P$	the set of all faces of a polyhedron P
$\operatorname{dom} f$	effective domain of a function f
$[a \leq f \leq b]$	subset of $\operatorname{dom} f$ whose function value is between a and b
$\operatorname{argmin} f$	the set of minimizers of a function f
∇f	gradient of a function f
$\nabla_i f(x)$	the i-th element of $\nabla f(x)$
∂f	subdifferential of a function f
$f'(\cdot, d)$	directional derivative of a function f in the direction d

Operators

Id	identity operator
$\Pi_{\mathcal{C}}$	projection onto a set $\mathcal{C}$
prox_f	proximal operator of a function f
$\operatorname{Fix} T$	fixed-point set of an operator T
$\operatorname{ran}(T)$	range of an operator T
$\min(x, y)$	element-wise minimum of x and y
$\max(x, y)$	element-wise maximum of x and y
sat	saturation operator
sth	soft thresholding operator

Acronyms

ADMM	Alternating Direction Method of Multipliers
APM	Alternating Projection Method
BCGD	Block Coordinate Gradient Descent
BCM	Block Coordinate Minimization
DRS	Douglas-Rachford Splitting
FNE	Firmly NonExpansive
HSDE	Homogeneous Self-Dual Embedding
LP	Linear Program
MHE	Moving Horizon Estimation
MIQP	Mixed-Integer Quadratic Program
MPC	Model Predictive Control
PGM	Proximal Gradient Method
PRS	Peaceman-Rachford Splitting
QNE	Quasi-NonExpansive
QP	Quadratic Program
SDP	SemiDefinite Program
SOCP	Second-Order Cone Program
SQNE	Strongly Quasi-NonExpansive
SQP	Sequential Quadratic Programming

Contents

1 Introduction

Contents

This dissertation is concerned with solving structured convex optimization problems. In particular, we consider methods for solving composite minimization problems of the form

$$\text{minimize} \quad f(x) + g(x), \tag{1.1}$$

where f and g are convex functions.

The above problem structure generalizes many problems of practical interest. Examples include regularized optimization when one of the functions represents some regularization term, constrained optimization when constraints are embedded in one of the functions through the indicator function of some set $\mathcal{C}$, *i.e.*

$$\mathcal{I}_{\mathcal{C}}(x) \coloneqq \begin{cases} 0 & x \in \mathcal{C} \\ +\infty & \text{otherwise,} \end{cases}$$

or feasibility problems when both functions are the indicator functions of some sets.

In applications such as image processing, statistics, and machine learning, we seek solutions of optimization problems whose dimensions can be very large. For such problems, classical optimization algorithms such as interior-point methods can fail to provide a solution. This has triggered interest to revisit the family of first-order optimization methods for solving problem (1.1), commonly known as *decomposition schemes* or *operator splitting methods*.

1.1 Background

Since interior-point methods were introduced, they have been used predominantly for solving constrained convex optimization problems because of their excellent theoretical and practical performance. Interior-point methods model the problem constraints as parametrized penalty functions, also referred to as *barrier functions*. At each iteration an unconstrained smooth optimization problem is solved with a Newton-type method, and then the barrier function parameters are updated. These methods are the methods of choice for solving small to medium size problems, and are currently the default algorithms for most off-the-shelf software packages for convex optimization [100, 126, 166].

In the last 15 years or so first-order methods have gained increasing attention in a wide range of applications for several reasons. First, in large-scale optimization interior-point methods may become intractable because of their relatively large per-iteration computational cost. On the other hand, first-order methods scale much better with the problem dimensions, can exploit sparsity in the problem data efficiently and are often easily parallelizable which is a favorable property in decentralized optimization [45]. Second, requirements on the solution accuracy are often moderate because of noise in the data and arbitrariness of the objective. This argument supports the use of first-order methods which are known to return solution of a medium accuracy at a reasonable effort [147]. Finally, iterations of first-order methods are usually very simple and computationally cheap, which makes them suitable for embedded applications where data are processed in real time on embedded systems with very limited computational and memory resources [157]. Moreover, these methods have also been used as heuristics in nonconvex optimization [24, 67, 114].

First-order methods use only first-order information of functions involved in the optimization problem, *i.e.* their gradients, or more generally, the subgradients. Gradient methods are well-known tools for smooth optimization based on the concept of iterative descent of the objective function. On the other hand, proximal methods are designed for solving nonsmooth and constrained optimization problems. The base operation of these methods is evaluating the *proximal operator* of a function, defined as

$$\operatorname{prox}_f(x) \coloneqq \operatorname*{argmin}_y \{f(y) + \tfrac{1}{2}\|y - x\|^2\},$$

which requires solution of an ancillary optimization problem. However, these subproblems often admit closed-form solutions [52], which makes proximal methods effective in practical applications. Note that, since the proximal operator of a function can be interpreted as the resolvent of the function's subdifferential operator [139], proximal methods belong to the class of first-order methods.

1.2 Operator splitting methods

Operator splitting methods are a class of first-order methods for solving problem (1.1) in a way that the functions f and g are tackled separately, usually through their gradients or proximal operators. This approach is particularly effective when at least one of the two functions is nonsmooth, or when the proximal operators of the functions f and g are easier to evaluate than the proximal operator of their sum. These methods encompass techniques such as the proximal gradient method (PGM), Peaceman-Rachford splitting (PRS), Douglas-Rachford splitting (DRS), and the alternating direction method of multipliers (ADMM).

A known limitation of these methods is that they can sometimes converge very slowly, and in particular converge sublinearly in many problems of practical interest. Many of the results on *linear convergence* of these methods require restrictive assumptions such as strong convexity and Lipschitz smoothness of one of the functions involved [92, 93]. Real-world problems rarely have such structure, so the aforementioned results cannot ensure linear convergence of splitting methods for solving such problems. Although linear convergence for some of these problems is still observed in practice [15, 38, 133], theoretical explanation for such behavior is still missing.

Moreover, the number of iterations required to provide a solution is highly dependent on the problem data and on the user's choice of the algorithm's parameters. Despite some recent theoretical results [93], it remains unclear how to select these parameters to improve the algorithm's convergence rate.

Another drawback of splitting methods is their inability to detect infeasible problems, which is very important in any embedded application or in mixed-integer optimization when branch-and-bound techniques are used [127]. It is well-known that for infeasible problems some of the iterates of DRS and ADMM diverge [80]. However, aside from this result, the asymptotic behavior for infeasible problems has been studied only in some very special cases, such as [18, 142].

We will often find it convenient to represent iterations of an optimization algorithm as the *fixed-point iteration* of some operator T, *i.e.*

$$x^{k+1} \leftarrow Tx^k, \tag{1.2}$$

where the sequence $\{x^k\}_{k\in\mathbb{N}}$ is equivalent to the iterates generated by the algorithm. Operators arising from the iteration of a particular splitting method often enjoy some useful properties such as nonexpansiveness [17, 154].

1.3 Outline

Operator splitting methods for solving problem (1.1) can be seen as methods for finding a zero of the sum of subdifferential operators of the functions involved, which are monotone for any convex functions [80, 150]. This characterization is important since there exist tight connections among the areas of convex analysis, monotone operator theory, and the theory of nonexpansive operators [17], which allows us to use results from operator theory to analyze behavior of splitting methods. From this central idea, a wealth of results on the asymptotic behavior of these methods can be derived.

Here we outline the contributions of each of the remaining chapters. If not otherwise specified, I am the main author of the contributions presented.

Background and infeasibility detection

Chapter 2: Convex analysis and operator theory play a central role throughout the dissertation. This chapter brings together some fundamental definitions and results related to convex sets and functions, convex optimization, and nonexpansive operators that are crucial for the development of later chapters.

Chapter 3: In this chapter we consider ADMM, a powerful operator splitting technique for solving structured optimization problems. For convex optimization problems, it is well-known that the iterates generated by ADMM converge to a solution provided that it exists, and diverge otherwise. Nevertheless, we show that the ADMM iterates yield conclusive information regarding problem infeasibility for a wide range of convex optimization problems. In particular, we show that in the limit the ADMM iterates either satisfy a set of first-order optimality conditions or produce a certificate of either primal or dual infeasibility. Based on these results, we propose reliable termination criteria for detecting primal and dual infeasibility. This chapter is based on

- G. Banjac, P. Goulart, B. Stellato, and S. Boyd. "Infeasibility detection in the alternating direction method of multipliers for convex optimization". In: *optimization-online.org* (2017).

Convergence rate analysis

Chapter 4: In this chapter we establish necessary and sufficient conditions for global linear convergence in methods represented as the fixed-point iteration of a quasi-nonexpansive (QNE) operator. We also provide a tight bound on the achievable convergence rate. Most existing results establishing global linear convergence in such methods require restrictive assumptions such as contractiveness of the fixed-point operator. However, there are several examples in the literature

showing that the linear convergence is possible even when such property does not hold. We provide a unifying framework for establishing global linear convergence based on linear regularity of the fixed-point operator, and show that some existing results are special cases of our approach. Moreover, we propose a novel linearly convergent splitting method for linear programming. This chapter is based on

- G. Banjac and P. Goulart. "Global linear convergence in operator splitting methods". In: *IEEE Conference on Decision and Control (CDC)*. 2016, pp. 233–238.
- G. Banjac and P. Goulart. "Tight global linear convergence rate bounds for operator splitting methods". In: *IEEE Transactions on Automatic Control (To appear)* (2018).

Chapter 5: In this chapter we consider a regularized version of the Jacobi algorithm, a decentralized method for convex optimization with an objective function consisting of the sum of a differentiable function and a block-separable function. Under certain regularity assumptions on the objective function, this algorithm has been shown to satisfy the so-called sufficient decrease condition, and consequently to converge in objective function value. In this chapter we revisit the convergence analysis of the regularized Jacobi algorithm and show that it also converges in iterates under very mild conditions on the objective function. Moreover, we establish conditions under which the algorithm achieves a linear convergence rate. This chapter is based on

- G. Banjac, K. Margellos, and P. Goulart. "On the convergence of a regularized Jacobi algorithm for convex optimization". In: *IEEE Transactions on Automatic Control* 63.4 (2018), pp. 1113–1119.

Implementation and software

Chapter 6: In this chapter we present a general-purpose solver for quadratic programs (QPs) based on ADMM, employing a novel operator splitting technique that requires the solution of a quasi-definite linear system with the same coefficient matrix in each iteration. The resulting algorithm is very robust, placing no requirements on the problem data such as strong convexity of the objective function or linear independence of the constraint functions. It is division-free once an initial matrix factorization is carried out, making it suitable for real-time applications in embedded systems. The method also supports factorization caching and warm starting, making it particularly efficient when solving parametrized problems arising in finance, control, and machine learning. Our open-source C implementation OSQP has been extensively tested on many problem instances from a wide variety of application areas. This chapter is based on

- G. Banjac, B. Stellato, N. Moehle, P. Goulart, A. Bemporad, and S. Boyd. "Embedded code generation using the OSQP solver". In: *IEEE Conference on Decision and Control (CDC)*. 2017, pp. 1906–1911.

- B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. "OSQP: an operator splitting solver for quadratic programs". In: *arXiv:1711.08013* (2017).

This work was co-authored with Bartolomeo Stellato. We worked together on the formulation and algorithm, as well as on the numerical implementation. Afterwards, I focused on the embedded code generation. Bartolomeo Stellato focused more on the interfaces and the extensive numerical testing.

Nonconvex optimization and conclusion

Chapter 7: In this chapter we consider the problem of minimizing a convex differentiable function subject to sparsity constraints. Such constraints are nonconvex and the resulting optimization problem is known to be hard to solve. We propose a novel generalization of this problem and demonstrate that it is equivalent to the original sparsity-constrained problem if a certain weighting term is sufficiently large. We use PGM to solve our generalized problem, and show that under certain regularity assumptions on the objective function the algorithm converges to a stationary point. Numerical results show that our algorithm outperforms other algorithms proposed in the literature. This chapter is based on

- G. Banjac and P. Goulart. "A novel approach for solving convex problems with cardinality constraints". In: *IFAC World Congress*. 2017, pp. 13182–13187.

Chapter 8: This chapter summarizes the main contributions of the dissertation and suggests some directions for future research.

2
Background

Contents

In this chapter we introduce some definitions and results from the areas of convex analysis, convex optimization, and operator theory which will be used in subsequent chapters. All the results presented here are standard, and can be found *e.g.* in [17, 46, 151, 152].

2.1 Convex sets

Definition 2.1 (Convex set)**.** *A set $\mathcal{C} \subseteq \mathbb{R}^n$ is convex if, for every pair of points $x \in \mathcal{C}$ and $y \in \mathcal{C}$, it includes the line segment joining them,* i.e.

$$(1-\tau)x + \tau y \in \mathcal{C}, \text{ for all } \tau \in (0,1).$$

Example 2.2 (Box)**.** *Given some $l \in \tilde{\mathbb{R}}^n$ and $u \in \tilde{\mathbb{R}}^n$ such that $l \leq u$, we define a* box *as the set of points that are lower-bounded element-wise by l and upper-bounded element-wise by u,* i.e.

$$[l, u] \coloneqq \{x \in \mathbb{R}^n \,|\, l \leq x \leq u\}.$$

Example 2.3 (Polyhedron)**.** *A set $\mathcal{C}$ is a (convex) polyhedron if it can be defined by a set of affine inequalities*

$$\mathcal{C} \coloneqq \{x \in \mathbb{R}^n \,|\, Ax \leq b\}$$

for some matrix $A \in \mathbb{R}^{m\times n}$ and vector $b \in \mathbb{R}^m$.

Both the box and the polyhedron are convex sets. Observe that a box is a special case of a polyhedron. A *face* of a polyhedron $\mathcal{C}$ is defined as a nonempty minimizer of a linear function over $\mathcal{C}$.

Proposition 2.4 (Set intersections [151, Thm. 2.1 & p.174])**.**

(i) The intersection of an arbitrary collection of convex sets is convex.

(ii) The intersection of a finite collection of (convex) polyhedra is a polyhedron.

Separation is an important principle in convex analysis. A hyperplane is said to *separate* two sets if one set is included in one of the corresponding closed half-spaces, and the other set is included in the other. The separation is said to be *strong* if the distance between the hyperplane and at least one of the sets is nonzero.

Theorem 2.5 (Separating hyperplane [151, Thm. 11.1])**.** *Let $\mathcal{C}_1 \subseteq \mathbb{R}^n$ and $\mathcal{C}_2 \subseteq \mathbb{R}^n$ be nonempty convex sets. There exists a hyperplane separating $\mathcal{C}_1$ and $\mathcal{C}_2$ strongly if and only if there exist some vector $a \in \mathbb{R}^n$ and scalar $b \in \mathbb{R}$ such that*

$$\inf_{x\in\mathcal{C}_1} \langle a, x\rangle \geq b > \sup_{x\in\mathcal{C}_2} \langle a, x\rangle .$$

The hyperplane $\{x \in \mathbb{R}^n \mid \langle a, x\rangle = b\}$ is called a separating hyperplane*.*

If two sets can be separated strongly, then the distance between them is nonzero, *i.e.* the sets are disjoint.

2.1.1 Convex cones

Definition 2.6 (Convex cone)**.** *A set $\mathcal{K} \subseteq \mathbb{R}^n$ is a convex cone if it is convex and closed under nonnegative scaling,* i.e. *for every point $x \in \mathcal{K}$ and scalar $\tau \geq 0$, $\tau x \in \mathcal{K}$.*

Example 2.7 (Nonnegative orthant)**.** *The set of points in $\mathbb{R}^n$ with nonnegative elements*

$$\mathbb{R}^n_+ \coloneqq \{x \in \mathbb{R}^n \,|\, x \geq 0\}$$

is called the nonnegative orthant*.*

Example 2.8 (Semidefinite cone)**.** *The set of real symmetric positive semidefinite matrices $\mathbb{S}^n_+$ is called the* semidefinite cone*.*

Both the nonnegative orthant and the semidefinite cone are convex cones.

Definition 2.9 (Polar cone)**.** *Given a convex cone $\mathcal{K} \subseteq \mathbb{R}^n$, we define its* polar cone *by*

$$\mathcal{K}^\circ \coloneqq \{y \in \mathbb{R}^n \mid \sup\nolimits_{x\in\mathcal{K}} \langle x, y\rangle \leq 0\}.$$

The dual cone of $\mathcal{K}$ is defined as $\mathcal{K}^* := -\mathcal{K}^\circ$. Note that both the nonnegative orthant and the semidefinite cone are *self-dual*, meaning that they are their own duals.

Definition 2.10 (Normal cone)**.** *Given a convex set $\mathcal{C} \subseteq \mathbb{R}^n$, we define its* normal cone *at a point $x \in \mathcal{C}$ as*

$$N_{\mathcal{C}}(x) := \{y \in \mathbb{R}^n \mid \sup_{x' \in \mathcal{C}} \langle x' - x, y \rangle \leq 0\}.$$

Every element of the normal cone of $\mathcal{C}$ at its boundary point x defines the outward normal of a supporting hyperplane of $\mathcal{C}$ at x. The normal cone at a point which is in the interior of $\mathcal{C}$ is $\{0\}$.

Definition 2.11 (Recession cone)**.** *Given a convex set $\mathcal{C} \subseteq \mathbb{R}^n$, its* recession cone *is defined as*

$$\mathcal{C}^\infty := \{y \in \mathbb{R}^n \mid x + \tau y \in \mathcal{C},\, x \in \mathcal{C},\, \tau \geq 0\},$$

i.e. *$\mathcal{C}$ includes all the half-lines in the direction of $y \in \mathcal{C}^\infty$ which start at points in $\mathcal{C}$.*

Note that the recession cone of a bounded set is $\{0\}$.

2.2 Convex functions

Definition 2.12 (Convex function)**.** *A function $f : \mathbb{R}^n \mapsto \tilde{\mathbb{R}}$ is convex if, for every pair of points $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$,*

$$f((1-\tau)x + \tau y) \leq (1-\tau)f(x) + \tau f(y), \text{ for all } \tau \in (0,1).$$

A function $f : \mathbb{R}^n \mapsto \tilde{\mathbb{R}}$ is called *concave* if $-f$ is convex.

Note that the function f in Definition 2.12 assigns a value on the extended real line $\tilde{\mathbb{R}}$. The *effective domain* of f is defined as

$$\operatorname{dom} f := \{x \in \mathbb{R}^n \mid f(x) < +\infty\}.$$

A function f is *proper* if it does not take the value $-\infty$ and its effective domain is nonempty.

The *epigraph* of f is defined as

$$\operatorname{epi} f := \{(x,t) \in \mathbb{R}^n \times \mathbb{R} \mid x \in \operatorname{dom} f,\, f(x) \leq t\}.$$

Note that a function is convex if and only if its epigraph is a convex set, and that the epigraph of a proper, convex, and lower semicontinuous function is nonempty, closed, and convex.

2.2.1 Indicator and support functions

We introduce here two convex functions associated to a convex set $\mathcal{C}$ that will be of particular interest in this dissertation.

Definition 2.13 (Indicator function)**.** *Given a set $\mathcal{C} \subseteq \mathbb{R}^n$, we define its* indicator function *as*

$$\mathcal{I}_{\mathcal{C}}(x) \coloneqq \begin{cases} 0 & x \in \mathcal{C} \\ +\infty & \text{otherwise.} \end{cases}$$

The indicator function of a nonempty, closed, and convex set is proper, convex, and lower semicontinuous.

Definition 2.14 (Support function)**.** *Given a convex set $\mathcal{C} \subseteq \mathbb{R}^n$, we define its* support function *as*

$$S_{\mathcal{C}}(x) \coloneqq \sup_{y \in \mathcal{C}} \langle x, y \rangle .$$

Note that the effective domain of $S_{\mathcal{C}}$ is $(\mathcal{C}^{\infty})^{\circ}$ [151, p.112 & Cor. 14.2.1].

2.2.2 Strong convexity and Lipschitz smoothness

Any differentiable function f is convex if and only if

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle \tag{2.1}$$

holds for all $x \in \operatorname{dom} f$ and $y \in \mathbb{R}^n$. The inequality (2.1) means that a convex function is lower-bounded by a (possibly infinite) collection of affine functions. We can define a more restrictive class of convex functions by imposing a minimum curvature of the function.

Definition 2.15 (Strong convexity)**.** *A differentiable function $f : \mathbb{R}^n \mapsto \tilde{\mathbb{R}}$ is* σ–strongly convex *with $\sigma > 0$ if, for every pair of points $x \in \operatorname{dom} f$ and $y \in \mathbb{R}^n$,*

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{\sigma}{2} \|y - x\|^2.$$

A strongly convex function has curvature at least σ. Functions with an upper bound on their curvature are called *Lipschitz smooth*.

Definition 2.16 (Lipschitz smoothness)**.** *A convex function $f : \mathbb{R}^n \mapsto \mathbb{R}$ is* L–Lipschitz smooth *with $L \geq 0$ if it is differentiable and, for every pair of points $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$,*

$$f(y) \leq f(x) + \langle \nabla f(x), y - x \rangle + \frac{L}{2} \|y - x\|^2.$$

Lipschitz smoothness of a (not necessarily convex) function can alternatively be defined as that

$$\|\nabla f(y) - \nabla f(x)\| \leq L \|y - x\|$$

holds for every pair of points x and y, which means that ∇f is L–Lipschitz continuous.

2.3 Convex optimization

A *convex optimization problem* is a minimization problem of the form

$$\begin{aligned} &\text{minimize} && f_0(x) \\ &\text{subject to} && f_i(x) \leq 0, \quad i = 1, \ldots, m \\ &&& g_i(x) = 0, \quad i = 1, \ldots, p, \end{aligned} \tag{2.2}$$

where each of the functions $f_i : \mathbb{R}^n \mapsto \tilde{\mathbb{R}}$ is convex, and each of the functions $g_i : \mathbb{R}^n \mapsto \tilde{\mathbb{R}}$ is affine. The function f_0 is referred to as the *cost* or *objective function*, while the remaining functions f_i and g_i are referred to as the *constraint functions*. The set of points satisfying all the problem constraints is referred to as the *feasible set*. If the feasible set is empty, then we say that the problem is *infeasible*.

2.3.1 Dual problem

We define the *Lagrangian* associated with problem (2.2) as

$$L(x, \lambda, \nu) \coloneqq f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i g_i(x), \tag{2.3}$$

where λ_i and ν_i are the *Lagrange multipliers* associated with the inequality constraint $f_i(x) \leq 0$ and the equality constraint $g_i(x) = 0$, respectively.

We define the *dual function* as the minimum value of the Lagrangian over x, *i.e.*

$$g(\lambda, \nu) \coloneqq \inf_x L(x, \lambda, \nu). \tag{2.4}$$

The dual function is always concave, and its value for any $\lambda \geq 0$ and ν is a lower bound on the optimal value of problem (2.2). We can thus look for the best lower bound that can be obtained from the dual function. This leads to the following optimization problem:

$$\begin{aligned} &\text{maximize} && g(\lambda, \nu) \\ &\text{subject to} && \lambda \geq 0, \end{aligned} \tag{2.5}$$

which we refer to as the *dual problem* associated with problem (2.2).

Note that the dual problem can be cast as a convex optimization problem since maximizing a concave function can be recast as minimizing a convex function.

2.3.2 Characterization of a minimizer

If we denote the feasible set in problem (2.2) by $\mathcal{C}$, then the problem can be reformulated as

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & x \in \mathcal{C}. \end{array} \tag{2.6}$$

Theorem 2.17 (Minimizer of constrained problem [152, Thm. 6.12])**.** *Let $f_0 : \mathbb{R}^n \mapsto \tilde{\mathbb{R}}$ be a convex differentiable function, and $\mathcal{C} \subseteq \mathbb{R}^n$ a nonempty convex set. A necessary and sufficient condition for $x \in \mathcal{C}$ to be a minimizer of problem* (2.6) *is that*

$$\inf_{x' \in \mathcal{C}} \langle \nabla f_0(x), x' - x \rangle \geq 0.$$

Note that by representing the constraint set $\mathcal{C}$ via its indicator function, we can reformulate problem (2.6) as the unconstrained minimization of a function that is still convex, but not differentiable. We next show how to characterize a minimizer of a function via its subdifferential.

Theorem 2.18 (Fermat's rule [17, Thm. 16.2])**.** *Let $h : \mathbb{R}^n \mapsto \tilde{\mathbb{R}}$ be a proper and convex function. Then $x \in \operatorname{dom} h$ is a minimizer of h if and only if $0 \in \partial h(x)$.*

Definition 2.19 (Stationary point [163, §3])**.** *Let $h : \mathbb{R}^n \mapsto \tilde{\mathbb{R}}$ be a proper function. Then $x \in \operatorname{dom} h$ is called a stationary point of h if*

$$h'(x, d) \geq 0, \text{ for all } d \in \mathbb{R}^n.$$

Note that a stationary point of a convex function is also its minimizer [17, Prop. 17.3]. If h is differentiable, then $h'(x, d) = \langle \nabla h(x), d \rangle$.

2.3.3 Linear and quadratic programming

A *quadratic program (QP)* is a convex optimization problem of the form

$$\begin{array}{ll} \text{minimize} & \frac{1}{2} x^T P x + q^T x \\ \text{subject to} & Ax \leq b \\ & Cx = d, \end{array} \tag{2.7}$$

where $P \in \mathbb{S}^n_+$, $A \in \mathbb{R}^{m \times n}$, and $C \in \mathbb{R}^{p \times n}$. In a QP we minimize a convex quadratic function over a polyhedron. If $P = 0$, then the objective function in (2.7) is linear, and the problem is referred to as a *linear program (LP)*.

2.3.4 Semidefinite programming

A *semidefinite program (SDP)* is a problem of the form

$$\begin{aligned} &\text{minimize} && c^T x \\ &\text{subject to} && F_0 + \sum_{i=1}^{n} F_i x_i \succeq 0, \end{aligned} \tag{2.8}$$

where $F_i \in \mathbb{S}^m$ and $x \coloneqq (x_1, \ldots, x_n)$. The constraint set in (2.8) is convex since it is the intersection of an affine set and a semidefinite cone, both of which are convex. Therefore, an SDP is a convex optimization problem.

2.4 Nonexpansive operators

Operator theory plays an important role in this dissertation since it allows us to analyze behavior of operator splitting methods through the fixed-point iteration of some operator T. We denote the identity operator in $\mathbb{R}^n$ by $\mathrm{Id} : \mathbb{R}^n \mapsto \mathbb{R}^n$.

Definition 2.20 (Fixed-point set)**.** *Given an operator $T : \mathbb{R}^n \mapsto \mathbb{R}^n$, we define its* fixed-point set *as*

$$\operatorname{Fix} T \coloneqq \{x \in \mathbb{R}^n \mid Tx = x\}.$$

Many optimization problems can be reformulated as problems of finding a fixed-point of some operator. Therefore, we are interested in analyzing properties of such an operator that ensures convergence of its fixed-point iteration. Let $\mathcal{D} \subseteq \mathbb{R}^n$ be a nonempty subset of $\mathbb{R}^n$.

Definition 2.21 (Nonexpansive operator)**.** *An operator $T : \mathcal{D} \mapsto \mathbb{R}^n$ is nonexpansive if, for every pair of points $x \in \mathcal{D}$ and $y \in \mathcal{D}$,*

$$\|Tx - Ty\| \leq \|x - y\|.$$

The fixed-point iteration of a nonexpansive operator does not necessarily converge. An example is the operator $-\mathrm{Id}$ whose fixed-point set is $\{0\}$. If the initial iterate of the fixed-point iteration is $x^0 \neq 0$, then the sequence $\{x^k\}_{k\in\mathbb{N}}$ does not converge.

However, it is possible to modify a nonexpansive operator so that the fixed-point iteration of the modified operator *does* converge.

Definition 2.22 (Averaged operator)**.** *An operator $T : \mathcal{D} \mapsto \mathbb{R}^n$ is* α–averaged *with $\alpha \in (0, 1)$ if there exists a nonexpansive operator $R : \mathcal{D} \mapsto \mathbb{R}^n$ such that*

$$T = (1 - \alpha)\,\mathrm{Id} + \alpha R.$$

Iteration of an α–averaged operator converges to its fixed-point, provided that it exists. An alternative definition of an α–averaged operator is that

$$\|Tx - Ty\|^2 + \alpha^{-1}(1 - \alpha)\,\|(\mathrm{Id} - T)x - (\mathrm{Id} - T)y\|^2 \leq \|x - y\|^2 \tag{2.9}$$

holds for all $x \in \mathcal{D}$ and $y \in \mathcal{D}$. An operator satisfying (2.9) for $\alpha = 1/2$ is called *firmly nonexpansive (FNE)*.

2.4.1 Proximal and projection operators

Definition 2.23 (Proximal operator)**.** *Given a proper, convex, and lower semi-continuous function $f : \mathbb{R}^n \mapsto \tilde{\mathbb{R}}$, we define its* proximal operator *as*

$$\operatorname{prox}_f(x) \coloneqq \operatorname*{argmin}_y \{f(y) + \tfrac{1}{2}\|y - x\|^2\}.$$

We treat prox_f as a single valued operator $\operatorname{prox}_f : \mathbb{R}^n \mapsto \mathbb{R}^n$ since the quadratic term in the preceding definition ensures that the set of minimizers is always single-valued.

Proposition 2.24 (See *e.g.* [18, Prop. 12.27])**.** *Let $f : \mathbb{R}^n \mapsto \tilde{\mathbb{R}}$ be a proper, convex, and lower semicontinuous function. Then* prox_f *is FNE.*

Note from Definition 2.23 that the proximal operator of f can be characterized via its subdifferential operator, *i.e.*

$$p = \operatorname{prox}_f(x) \quad \Longleftrightarrow \quad 0 \in \partial f(p) + p - x \quad \Longleftrightarrow \quad p = (\partial f + \mathrm{Id})^{-1} x.$$

Example 2.25 (Soft thresholding)**.** *The proximal operator of the function $\gamma\,|x| : \mathbb{R} \mapsto \mathbb{R}_+$, where $\gamma > 0$, is*

$$\operatorname{prox}_{\gamma\,|\cdot|}(x) = \operatorname{sth}_\gamma(x),$$

where the soft thresholding operator $\operatorname{sth}_\gamma$ *is defined as*

$$\operatorname{sth}_\gamma(x) \coloneqq \begin{cases} x + \gamma & x < -\gamma \\ 0 & |x| \le \gamma \\ x - \gamma & x > \gamma. \end{cases}$$

Definition 2.26 (Projection operator)**.** *Given a nonempty, closed, and convex set $\mathcal{C} \subseteq \mathbb{R}^n$, we define its* projection operator *as*

$$\Pi_{\mathcal{C}}(x) \coloneqq \operatorname*{argmin}_{y \in \mathcal{C}} \|y - x\|.$$

Note that the projection on a set can be seen as the proximal operator of its indicator function. Therefore, the projection operator of a nonempty, closed, and convex set is also FNE.

Example 2.27. *The projection of $x \in \mathbb{R}^n$ onto the positive orthant is given by $x_+ \coloneqq \max(x, 0)$.*

Example 2.28 (Saturation)**.** *The projection of $x \in \mathbb{R}$ onto the box $[l, u] \subseteq \mathbb{R}$ is*

$$\Pi_{[l,u]}(x) = \operatorname{sat}_{[l,u]}(x),$$

where $\operatorname{sat}_{[l,u]}(x) \coloneqq \max(\min(x, u), l)$ is called the saturation operator*.*

Theorem 2.29 (Moreau decomposition [17, Thm 6.29])**.** *Let $\mathcal{K} \subseteq \mathbb{R}^n$ be a nonempty, closed, and convex cone. Then we can decompose any point $x \in \mathbb{R}^n$ as*

$$x = \Pi_{\mathcal{K}}(x) + \Pi_{\mathcal{K}^\circ}(x),$$

where $\langle \Pi_{\mathcal{K}}(x), \Pi_{\mathcal{K}^\circ}(x) \rangle = 0$.

It is easy to show from Theorem 2.29 that the projection onto a cone is a positively homogeneous operator, *i.e.* $\Pi_{\mathcal{K}}(\tau x) = \tau \Pi_{\mathcal{K}}(x)$ for every $\tau \ge 0$.

3

Infeasibility Detection in ADMM

Contents

3.1 Introduction

Operator splitting methods can be used to solve composite minimization problems of the form

$$\text{minimize} \quad f(x) + g(x), \tag{3.1}$$

where f and g are proper, convex, and lower semicontinuous functions. These methods encompass algorithms such as the proximal gradient method (PGM), Douglas-Rachford splitting (DRS) and the alternating direction method of multipliers (ADMM) [139], and have been applied to problems ranging from feasibility and best approximation problems [16, 18] to quadratic and conic programs [38, 136, 176]. Due to their relatively low per-iteration computational cost and ability

to exploit sparsity in the problem data [176], splitting methods are suitable for embedded [109, 137] and large-scale optimization [26], and have increasingly been applied for solving problems arising in signal processing [52, 53], machine learning [45] and optimal control [157].

In order to solve problem (3.1), PGM requires differentiability of one of the two functions. If a fixed step-size is used in the algorithm, then one also requires that the function is Lipschitz smooth [26]. On the other hand, ADMM and DRS, which turn out to be equivalent to each other, do not require any additional assumptions on the problem beyond convexity, making them more robust to the problem data.

The growing popularity of ADMM has triggered a strong interest in understanding its theoretical properties. Provided that problem (3.1) is solvable and satisfies certain constraint qualification (see [17, Cor. 26.3] for more details), both ADMM and DRS are known to converge to an optimal solution [17, 45]. The use of ADMM for solving convex quadratic programs (QPs) was analyzed in [38] and was shown to admit an asymptotic linear convergence rate. The authors in [90] analyze global linear convergence of ADMM for solving convex QPs, and the authors in [93] extended these results to a wider class of optimization problems. A particularly convenient framework for analyzing asymptotic behavior of such method is by representing it as a fixed-point iteration of an averaged operator [9, 17, 80, 93].

It is well-known that for infeasible convex optimization problems some of the iterates of ADMM and DRS diverge [80]. The ability to detect infeasible problems is very important in many applications, *e.g.* in any embedded application or in mixed-integer optimization when branch-and-bound techniques are used [127]. However, terminating the algorithm when the iterates become large is unreliable in practice for several reasons. First, an upper bound on the norm of iterates should be big enough in order to reduce the number of false detections of infeasibility. Second, divergence of the algorithm's iterates is observed to be very slow in practice. Finally, such termination criterion is just an indication that a problem might be infeasible, and not a certificate of infeasibility.

Aside from [80], the asymptotic behavior of ADMM and DRS for infeasible problems has been studied only in some special cases. DRS for solving feasibility problems involving two convex sets that do not necessarily intersect was studied in [18, 19, 21, 22]. The authors in [142] study the asymptotic behavior of ADMM for solving convex QPs in the case when the problem is infeasible, but impose some strong assumptions on the problem data such as full rank and positive definiteness of certain problem matrices. The authors in [136] apply ADMM to the homogeneous self-dual embedding (HSDE) of a convex conic program, thereby producing a larger problem which is always feasible and whose solutions can be used either to produce a primal-dual solution or a certificate of infeasibility for the original problem. A disadvantage of this approach in application to QPs is that the problem needs to be transformed into an equivalent conic program which is harder to solve than the original QP in general.

In this chapter we consider a very general class of convex optimization problems that includes linear programs (LPs), QPs, second-order cone programs (SOCPs) and semidefinite programs (SDPs) as special cases. We use a particular version of ADMM that imposes no conditions on the problem data such as strong convexity of the objective function or full rank of the constraint matrix. We show that the method either generates iterates for which the violation of the optimality conditions goes to zero, or produces a certificate of primal or dual infeasibility. These results are directly applicable to infeasibility detection in ADMM for a wide range of problems.

3.2 Problem description

Consider the following convex optimization problem:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^T P x + q^T x \\ \text{subject to} & Ax \in \mathcal{C}, \end{array} \tag{3.2}$$

with $P \in \mathbb{S}^n_+$, $q \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$, and $\mathcal{C} \subseteq \mathbb{R}^m$ a nonempty, closed, and convex set. This problem formulation is suitable for representing and analyzing both quadratic (when $\mathcal{C} = [l, u]$) and conic programs (when $\mathcal{C} = \mathcal{K}_b$). We are interested in finding either an optimal solution to problem (3.2) or a certificate of either primal or dual infeasibility.

3.2.1 Optimality conditions

We will find it convenient to rewrite problem (3.2) in an equivalent form by introducing a variable $z \in \mathbb{R}^m$ to obtain

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^T P x + q^T x \\ \text{subject to} & Ax = z, \quad z \in \mathcal{C}. \end{array} \tag{3.3}$$

We can now derive the optimality conditions for this problem.

Lemma 3.1. *The first-order optimality conditions for problem* (3.3) *are*

$$Ax = z, \tag{3.4a}$$
$$Px + q + A^T y = 0, \tag{3.4b}$$
$$z \in \mathcal{C}, \quad y \in N_{\mathcal{C}}(z), \tag{3.4c}$$

where $y \in \mathbb{R}^m$ is a Lagrange multiplier associated with the constraint $Ax = z$.

Proof. The linear constraint in problem (3.3) can be relaxed by using the following Lagrangian subproblem:

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}x^T Px + q^T x + y^T(Ax - z) \\ \text{subject to} \quad & z \in \mathcal{C}. \end{aligned}$$

If we denote the objective function in the above problem by $F(x, y, z)$, then according to Theorem 2.17, the optimality conditions can be written as

$$\begin{aligned} & z \in \mathcal{C}, \\ & 0 = \nabla_x F(x, y, z) = Px + q + A^T y, \\ & 0 = \nabla_y F(x, y, z) = Ax - z, \\ & 0 \geq \sup_{z' \in \mathcal{C}} \langle -\nabla_z F(x, y, z), z' - z \rangle = \sup_{z' \in \mathcal{C}} \langle y, z' - z \rangle , \end{aligned}$$

where the last condition is equivalent to $y \in N_{\mathcal{C}}(z)$. □

If there exist $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$ and $y \in \mathbb{R}^m$ that satisfy conditions (3.4), then we say that (x, z) is a *primal* and y is a *dual* solution of problem (3.3).

3.2.2 Infeasibility certificate

In this section we derive conditions for primal and dual infeasibility. We first derive the dual problem associated with problem (3.2).

Lemma 3.2. *The dual of problem* (3.2) *is*

$$\begin{aligned} \text{maximize} \quad & -\tfrac{1}{2}x^T Px - S_{\mathcal{C}}(y) \\ \text{subject to} \quad & Px + A^T y = -q, \quad y \in (\mathcal{C}^\infty)^\circ. \end{aligned} \tag{3.5}$$

Proof. We first rewrite problem (3.2) in the form

$$\begin{aligned} \text{minimize} \quad & \tfrac{1}{2}x^T Px + q^T x + \mathcal{I}_{\mathcal{C}}(z) \\ \text{subject to} \quad & Ax = z, \end{aligned}$$

then form its Lagrangian,

$$L(x, z, y) \coloneqq \tfrac{1}{2}x^T Px + q^T x + \mathcal{I}_{\mathcal{C}}(z) + y^T(Ax - z),$$

and finally derive the dual function as follows:

$$\begin{aligned} g(y) &\coloneqq \inf_{x,z} L(x, z, y) \\ &= \inf_x \{\tfrac{1}{2}x^T Px + (A^T y + q)^T x\} + \inf_{z \in \mathcal{C}} \{-y^T z\} \\ &= \inf_x \{\tfrac{1}{2}x^T Px + (A^T y + q)^T x\} - \sup_{z \in \mathcal{C}} \{y^T z\}. \end{aligned}$$

Note that the minimum of the Lagrangian over x is obtained when $Px+A^Ty+q = 0$, and that the second term in the last line is the support function of $\mathcal{C}$. The dual problem can then be written in the form (3.5), where the conic constraint on y is just the restriction of y to the domain of $S_{\mathcal{C}}$. □

We will use the following pair of results to certify infeasibility of (3.2) in cases where it primal and/or dual *strongly infeasible*:

Proposition 3.3.

(i) If there exists some $\bar{y} \in \mathbb{R}^m$ such that

$$A^T\bar{y} = 0 \quad \textit{and} \quad S_{\mathcal{C}}(\bar{y}) < 0, \tag{3.6}$$

then the primal problem (3.2) *is infeasible.*

(ii) If there exists some $\bar{x} \in \mathbb{R}^n$ such that

$$P\bar{x} = 0, \quad A\bar{x} \in \mathcal{C}^\infty \quad \textit{and} \quad \langle q, \bar{x}\rangle < 0, \tag{3.7}$$

then the dual problem (3.5) *is infeasible.*

Proof. (i): The first condition in (3.6) implies

$$\inf_x \langle \bar{y}, Ax\rangle = \inf_x \left\langle A^T\bar{y}, x\right\rangle = 0,$$

and the second condition is equivalent to

$$\sup_{z\in\mathcal{C}} \langle \bar{y}, z\rangle < 0.$$

Therefore, $\{z \in \mathbb{R}^m \mid \langle \bar{y}, z\rangle = 0\}$ is a hyperplane that separates the sets $\{Ax \,|\, x \in \mathbb{R}^n\}$ and $\mathcal{C}$ strongly, meaning that problem (3.2) is infeasible.

(ii): Define the set $\mathcal{Q} := \{Px+A^Ty \,|\, (x, y) \in \mathbb{R}^n \times (\mathcal{C}^\infty)^\circ\}$. The first two conditions in (3.7) imply

$$\begin{aligned}\sup_{s\in\mathcal{Q}} \langle \bar{x}, s\rangle &= \sup\left\{\left\langle \bar{x}, Px + A^Ty\right\rangle \,\middle|\, x \in \mathbb{R}^n,\, y \in (\mathcal{C}^\infty)^\circ\right\} \\ &= \sup_x \langle P\bar{x}, x\rangle + \sup\{\langle A\bar{x}, y\rangle \mid y \in (\mathcal{C}^\infty)^\circ\} \\ &\leq 0,\end{aligned}$$

where we used the fact that the inner product between vectors in a cone and its polar is nonpositive. Since the third condition in (3.7) can be written as $\langle \bar{x}, -q\rangle > 0$, this means that $\{x \in \mathbb{R}^n \mid \langle \bar{x}, x\rangle = 0\}$ is a hyperplane that separates the sets $\mathcal{Q}$ and $\{-q\}$ strongly, and thus the dual problem (3.5) is infeasible. □

Note that if the condition (3.6) holds, then $\bar{y}$ also represents an unbounded direction in the dual problem assuming it is feasible. Likewise, $\bar{x}$ in the condition (3.7) represents an unbounded direction for the primal problem if it is feasible. However, since we cannot exclude the possibility of simultaneous primal and dual infeasibility, we will refer to the condition (3.6) as *primal infeasibility* rather than *dual unboundedness*, and *vice versa* for (3.7).

In some cases, *e.g.* when $\mathcal{C}$ is compact or polyhedral, conditions (3.6) and (3.7) in Proposition 3.3 are also necessary for infeasibility, and we say that (3.6) and (3.7) are *strong alternatives* for primal and dual feasibility, respectively. When $\mathcal{C}$ is a convex cone, additional assumptions are required for having strong alternatives; see *e.g.* [46, §5.9.4].

3.3 Alternating direction method of multipliers

ADMM is an operator splitting method that can be used for solving structured optimization problems [45]. The iterates of ADMM in application to problem (3.1) can be written as

$$\tilde{x}^{k+1} \leftarrow \operatorname{prox}_f(x^k - u^k) \tag{3.8a}$$
$$x^{k+1} \leftarrow \operatorname{prox}_g(\tilde{x}^{k+1} + u^k) \tag{3.8b}$$
$$u^{k+1} \leftarrow u^k + \tilde{x}^{k+1} - x^{k+1}. \tag{3.8c}$$

If $\tilde{x}^{k+1}$ in (3.8b) and (3.8c) is replaced by $\alpha\tilde{x}^{k+1} + (1-\alpha)x^k$ where $\alpha \in (0,2)$ is a *relaxation parameter*, then the resulting algorithm is called the *relaxed* ADMM.

We can write problem (3.3) in the general form (3.1) by setting

$$f(x,z) = \tfrac{1}{2}x^T P x + q^T x + \mathcal{I}_{Ax=z}(x,z),$$
$$g(x,z) = \mathcal{I}_{\mathcal{C}}(z).$$

If we use the norm $\|(x,z)\| = \sqrt{\sigma\|x\|_2^2 + \rho\|z\|_2^2}$ with $(\sigma,\rho) > 0$ in the proximal operators of functions f and g, then an iteration of the relaxed ADMM consists of the following steps:

$$(\tilde{x}^{k+1}, \tilde{z}^{k+1}) \leftarrow \underset{(\tilde{x},\tilde{z}):A\tilde{x}=\tilde{z}}{\operatorname{argmin}} \tfrac{1}{2}\tilde{x}^T P\tilde{x} + q^T\tilde{x} + \tfrac{\sigma}{2}\|\tilde{x} - x^k + \sigma^{-1}w^k\|_2^2 + \tfrac{\rho}{2}\|\tilde{z} - z^k + \rho^{-1}y^k\|_2^2 \tag{3.9a}$$
$$x^{k+1} \leftarrow \alpha\tilde{x}^{k+1} + (1-\alpha)x^k + \sigma^{-1}w^k \tag{3.9b}$$
$$z^{k+1} \leftarrow \Pi_{\mathcal{C}}\left(\alpha\tilde{z}^{k+1} + (1-\alpha)z^k + \rho^{-1}y^k\right) \tag{3.9c}$$
$$w^{k+1} \leftarrow w^k + \sigma\left(\alpha\tilde{x}^{k+1} + (1-\alpha)x^k - x^{k+1}\right) \tag{3.9d}$$
$$y^{k+1} \leftarrow y^k + \rho\left(\alpha\tilde{z}^{k+1} + (1-\alpha)z^k - z^{k+1}\right) \tag{3.9e}$$

Algorithm 3.1 Relaxed ADMM for problem (3.2).

1: **given** initial values x^0, z^0, y^0 and parameters $\rho > 0$, $\sigma > 0$, $\alpha \in (0, 2)$
2: **repeat**
3: $\quad (\tilde{x}^{k+1}, \tilde{z}^{k+1}) \leftarrow \underset{(\tilde{x},\tilde{z}):A\tilde{x}=\tilde{z}}{\operatorname{argmin}} \frac{1}{2}\tilde{x}^T P\tilde{x} + q^T\tilde{x} + \frac{\sigma}{2}\|\tilde{x} - x^k\|_2^2 + \frac{\rho}{2}\|\tilde{z} - z^k + \rho^{-1}y^k\|_2^2$
4: $\quad x^{k+1} \leftarrow \alpha\tilde{x}^{k+1} + (1-\alpha)x^k$
5: $\quad z^{k+1} \leftarrow \Pi_{\mathcal{C}}\left(\alpha\tilde{z}^{k+1} + (1-\alpha)z^k + \rho^{-1}y^k\right)$
6: $\quad y^{k+1} \leftarrow y^k + \rho\left(\alpha\tilde{z}^{k+1} + (1-\alpha)z^k - z^{k+1}\right)$
7: $\quad k \leftarrow k + 1$
8: **until** termination criterion is satisfied

The scalars ρ and σ are called the *penalty parameters* of the algorithm. Strict positivity of both ρ and σ ensure that the equality constrained QP in (3.9a) has a unique solution for any $P \in \mathbb{S}_+^n$ and $A \in \mathbb{R}^{m\times n}$. Observe from steps (3.9b) and (3.9d) that $w^{k+1} = 0$ for all k, and consequently the w-iterate and the step (3.9d) can be disregarded. Finally, the ADMM iterations reduce to Algorithm 3.1.

It is well-known that ADMM and DRS are equivalent methods [87]. The authors in [95] show that ADMM can be described alternatively in terms of the fixed-point iteration of an averaged operator. In particular, an iteration of Algorithm 3.1 is equivalent to

$$(\tilde{x}^k, \tilde{z}^k) \leftarrow \underset{(\tilde{x},\tilde{z}):A\tilde{x}=\tilde{z}}{\operatorname{argmin}} \frac{1}{2}\tilde{x}^T P\tilde{x} + q^T\tilde{x} + \frac{\sigma}{2}\|\tilde{x} - x^k\|_2^2 + \frac{\rho}{2}\|\tilde{z} - (2\Pi_{\mathcal{C}} - \mathrm{Id})(v^k)\|_2^2 \quad (3.10a)$$

$$x^{k+1} \leftarrow x^k + \alpha\left(\tilde{x}^k - x^k\right) \quad (3.10b)$$

$$v^{k+1} \leftarrow v^k + \alpha\left(\tilde{z}^k - \Pi_{\mathcal{C}}(v^k)\right) \quad (3.10c)$$

where

$$z^k = \Pi_{\mathcal{C}}(v^k) \quad \text{and} \quad y^k = \rho(\mathrm{Id} - \Pi_{\mathcal{C}})(v^k). \quad (3.11)$$

We will exploit the following result in the next section to analyze asymptotic behavior of the algorithm:

Fact 3.4. *The iteration described in* (3.10) *amounts to*

$$(x^{k+1}, v^{k+1}) \leftarrow T(x^k, v^k),$$

where T is an $(\alpha/2)$–averaged operator.

Proof. Follows from [95, §IV-C]. □

Due to [17, Prop. 6.46], the identities in (3.11) imply that at every iteration the pair (z^k, y^k) satisfies the optimality condition (3.4c) by construction. The

solution of the equality constrained QP in (3.10a) satisfies the following pair of optimality conditions:

$$0 = A\tilde{x}^k - \tilde{z}^k \tag{3.12a}$$

$$0 = (P + \sigma I)\tilde{x}^k + q - \sigma x^k + \rho A^T \left(\tilde{z}^k - (2\Pi_{\mathcal{C}} - \mathrm{Id})(v^k)\right). \tag{3.12b}$$

If we rearrange (3.10b) and (3.10c) to isolate $\tilde{x}^k$ and $\tilde{z}^k$, *i.e.* write

$$\tilde{x}^k = x^k + \alpha^{-1}\delta x^{k+1} \tag{3.13a}$$

$$\tilde{z}^k = z^k + \alpha^{-1}\delta v^{k+1}, \tag{3.13b}$$

where $\delta x^{k+1} \coloneqq x^{k+1} - x^k$, and substitute them into (3.12), then we obtain the following relations between the iterates:

$$Ax^k - \Pi_{\mathcal{C}}(v^k) = -\alpha^{-1}\left(A\delta x^{k+1} - \delta v^{k+1}\right) \tag{3.14a}$$

$$Px^k + q + \rho A^T(\mathrm{Id} - \Pi_{\mathcal{C}})(v^k) = -\alpha^{-1}\left((P + \sigma I)\delta x^{k+1} + \rho A^T \delta v^{k+1}\right). \tag{3.14b}$$

Observe that the right-hand terms of (3.14) are a direct measure of how far the iterates (x^k, z^k, y^k) are from satisfying the optimality conditions (3.4a) and (3.4b). In the next section, we will show that the successive differences $(\delta x^k, \delta v^k)$ appearing in the right hand side of (3.14) converge and can be used to test for primal and dual infeasibility.

3.4 Asymptotic behavior of ADMM

In order to analyze the asymptotic behavior of the iteration (3.10), which is equivalent to Algorithm 3.1, we will rely heavily on the following results:

Lemma 3.5. *Let $\mathcal{D}$ be a nonempty, closed and convex subset of $\mathbb{R}^n$ and suppose that $T : \mathcal{D} \mapsto \mathcal{D}$ is an averaged operator. Let $x^0 \in \mathcal{D}$, $x^k = T^k x^0$ and δx be the projection of the zero vector onto $\overline{\mathrm{ran}(T - \mathrm{Id})}$. Then*

(i) $\frac{1}{k}x^k \to \delta x$.

(ii) $\delta x^k \to \delta x$.

(iii) If $\mathrm{Fix}\, T \neq \emptyset$, *then x^k converges to a point in* $\mathrm{Fix}\, T$.

Proof. The first result is [140, Cor. 3], the second is [6, Cor. 2.3] and the third is [17, Thm. 5.14]. □

Note that, since $\mathrm{ran}(T - \mathrm{Id})$ is not necessarily closed, the projection onto this set may not exist, but the projection onto its closure always exists. Moreover, since $\overline{\mathrm{ran}(T - \mathrm{Id})}$ is convex [140, Lem. 4], the projection is unique. Due to Fact 3.4, Lemma 3.5 ensures that $(\frac{1}{k}x^k, \frac{1}{k}v^k) \to (\delta x, \delta v)$ and $(\delta x^k, \delta v^k) \to (\delta x, \delta v)$.

We make the following assumption on the constraint set in problem (3.2):

Assumption 3.6. *The set $\mathcal{C}$ is the Cartesian product of a convex compact set $\mathcal{B} \subseteq \mathbb{R}^{m_1}$ and a translated closed convex cone $\mathcal{K}_b \subseteq \mathbb{R}^{m_2}$, where $m_1 + m_2 = m$,* i.e. *$\mathcal{C} = \mathcal{B} \times \mathcal{K}_b$.*

Many convex problems of practical interest, including LPs, QPs, SOCPs and SDPs, can be written in the form of problem (3.2) with $\mathcal{C}$ satisfying the conditions of Assumption 3.6.

Core results of this chapter are contained within the following two propositions, which establish various relationships between the limits δx and δv. Given these two results, it will then be straightforward to extract certificates of optimality or infeasibility in Section 3.4.1. For both of these central results, and in the remainder of the chapter, we define

$$\delta z \coloneqq \Pi_{\mathcal{C}^\infty}(\delta v) \quad \text{and} \quad \delta y \coloneqq \rho \Pi_{(\mathcal{C}^\infty)^\circ}(\delta v).$$

Proposition 3.7. *Suppose that Assumption 3.6 holds. Then the following relations hold between the limits δx, δz and δy:*

(i) $A\delta x = \delta z$.

(ii) $P\delta x = 0$.

(iii) $A^T \delta y = 0$.

(iv) $\frac{1}{k} z^k \to \delta z$ *and* $\delta z^k \to \delta z$.

(v) $\frac{1}{k} y^k \to \delta y$ *and* $\delta y^k \to \delta y$.

Proof. See Appendix 3.B.1. □

Proposition 3.7 shows that the limits δy and δx will always satisfy the subspace and conic constraints in the primal and dual infeasibility tests (3.6) and (3.7), respectively. We next consider the terms appearing in the inequalities in (3.6) and (3.7).

Proposition 3.8. *Suppose that Assumption 3.6 holds. Then the following identities hold for the limits δx and δy:*

(i) $\langle q, \delta x \rangle = -\sigma\alpha^{-1}\|\delta x\|^2 - \rho\alpha^{-1}\|A\delta x\|^2$.

(ii) $S_{\mathcal{C}}(\delta y) = -\rho^{-1}\alpha^{-1}\|\delta y\|^2$.

Proof. See Appendix 3.B.2. □

3.4.1 Optimality and infeasibility certificates

We are now in a position to prove that, in the limit, the iterates of Algorithm 3.1 either satisfy the optimality conditions (3.4) or produce a certificate of infeasibility. Recall that Fact 3.4, Lemma 3.5(ii) and Proposition 3.7(iv)–(v) ensure convergence of the sequence $\{\delta x^k, \delta z^k, \delta y^k\}_{k\in\mathbb{N}}$.

Proposition 3.9 (Optimality)**.** *If $(\delta x^k, \delta z^k, \delta y^k) \to 0$, then the optimality conditions* (3.4) *are satisfied in the limit,* i.e.

$$\|Ax^k - z^k\| \to 0 \quad \textit{and} \quad \|Px^k + q + A^T y^k\| \to 0. \tag{3.15}$$

Proof. Follows from (3.11) and (3.14). □

Lemma 3.5(iii) is sufficient to prove that if problem (3.2) is solvable then the sequence of iterates $\{x^k, z^k, y^k\}_{k\in\mathbb{N}}$ converges to its primal-dual solution. However, convergence of $\{\delta x^k, \delta z^k, \delta y^k\}_{k\in\mathbb{N}}$ to zero is not itself sufficient to prove convergence of $\{x^k, z^k, y^k\}_{k\in\mathbb{N}}$; we provide a numerical example in Section 3.5.3 to show that this scenario can occur. According to Proposition 3.9, in this case the violation of optimality conditions still goes to zero in the limit. A meaningful criterion for detecting optimality is that the norms in (3.15) are small.

We next show that if $\{\delta x^k, \delta z^k, \delta y^k\}_{k\in\mathbb{N}}$ converges to a nonzero value, then we can construct a certificate of primal and/or dual infeasibility. Note that due to Proposition 3.7(i), δz can be nonzero only when δx is nonzero.

Theorem 3.10. *Suppose that Assumption 3.6 holds.*

(i) If $\delta y \neq 0$, then the problem (3.2) *is infeasible and δy satisfies the primal infeasibility conditions* (3.6)*.*

(ii) If $\delta x \neq 0$, then the problem (3.5) *is infeasible and δx satisfies the dual infeasibility conditions* (3.7)*.*

(iii) If $\delta x \neq 0$ and $\delta y \neq 0$, then problems (3.2) *and* (3.5) *are simultaneously infeasible.*

Proof. (i): Follows from Proposition 3.7(iii) and Proposition 3.8(ii).

(ii): Follows from Proposition 3.7(i)–(ii) and Proposition 3.8(i).

(iii): Follows from (i) and (ii). □

Since $(\delta x^k, \delta y^k) \to (\delta x, \delta y)$, a meaningful criterion for detecting primal and dual infeasibility would be to use δy^k and δx^k to check the conditions (3.6) and (3.7), respectively.

Remark 3.11. *It is easy to show that δy and δx would still provide certificates of primal and dual infeasibility if we instead used the norm $\|(x, z)\| = \sqrt{x^T S x + z^T R z}$ in the proximal operators in* (3.8)*, with R and S being diagonal positive definite matrices.*

3.5 Numerical examples

In this section we demonstrate via several numerical examples the different asymptotic behaviors of iterates generated by Algorithm 3.1 for solving optimization problems of the form (3.2).

3.5.1 Parametric QP

Consider the following QP:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x_1^2 + x_1 - x_2 \\ \text{subject to} & 0 \leq x_1 + ax_2 \leq u_1 \\ & 1 \leq x_1 \leq 3 \\ & 1 \leq x_2 \leq u_3, \end{array} \tag{3.16}$$

where $a \in \mathbb{R}$, $u_1 \geq 0$ and $u_3 \geq 1$ are parameters. Note that the problem above is an instance of problem (3.2) with

$$P = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}, \quad q = \begin{bmatrix} 1 \\ -1 \end{bmatrix}, \quad A = \begin{bmatrix} 1 & a \\ 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathcal{C} = [l, u], \quad l = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}, \quad u = \begin{bmatrix} u_1 \\ 3 \\ u_3 \end{bmatrix}.$$

Depending on the values of parameters u_1 and u_3, the constraint set in (3.16) can be either bounded or unbounded. The projection onto the set $[l, u]$ can be evaluated as

$$\Pi_{[l,u]}(x) = \max\left(\min(x, u), l\right),$$

and its support function as

$$S_{[l,u]}(y) = \langle l, \min(y, 0)\rangle + \langle u, \max(y, 0)\rangle,$$

where min and max functions should be taken element-wise. The support function of the translated cone $\mathcal{K}_b$ is

$$S_{\mathcal{K}_b}(y) = \begin{cases} \langle b, y\rangle & y \in \mathcal{K}^\circ \\ +\infty & \text{otherwise.} \end{cases}$$

In the sequel we will discuss four scenarios that can occur depending on the values of the parameters: (i) optimality, (ii) primal infeasibility, (iii) dual infeasibility, (iv) simultaneous primal and dual infeasibility, and will show that Algorithm 3.1 correctly produces certificates for all four scenarios. In all cases we set the parameters $\alpha = \rho = \sigma = 1$ and set the initial iterate $(x^0, z^0, y^0) = (0, 0, 0)$.

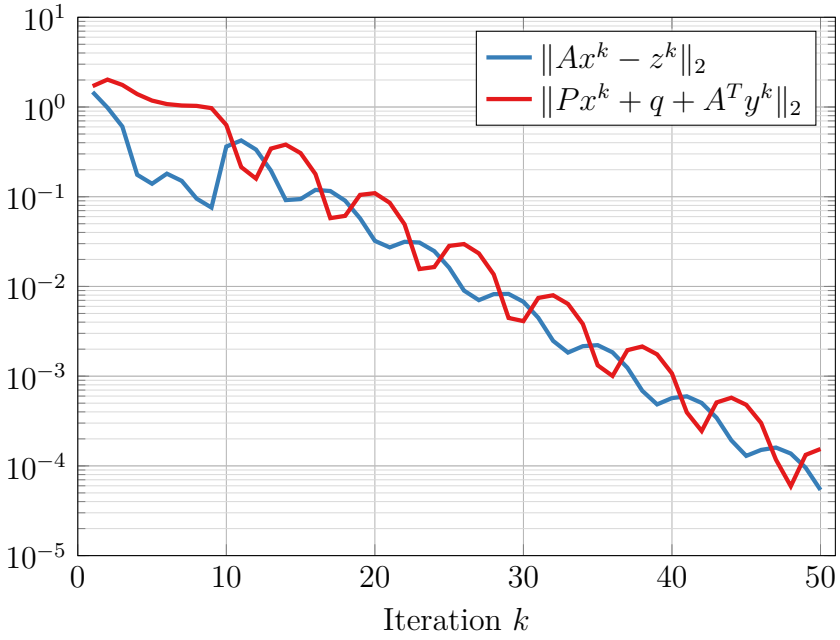


Figure 3.1: Convergence of $\{x^k, z^k, y^k\}_{k\in\mathbb{N}}$ to a certificate of optimality for problem (3.16) with $a = 1$, $u_1 = 5$ and $u_3 = 3$.

Optimality. Consider the problem (3.16) with parameters

$$a = 1, \quad u_1 = 5, \quad u_3 = 3.$$

Algorithm 3.1 converges to $x^\star = (1, 3)$, $z^\star = (4, 1, 3)$, $y^\star = (0, -2, 1)$, and we have

$$Ax^\star = z^\star \quad \text{and} \quad Px^\star + q + A^T y^\star = 0,$$

i.e. the pair $(x^\star, y^\star)$ is a primal-dual solution of problem (3.16). Figure 3.1 shows convergence of $\{x^k, z^k, y^k\}_{k\in\mathbb{N}}$ to a certificate of optimality. Recall that iterates of the algorithm always satisfy the optimality conditions (3.4c).

Primal infeasibility. We next set the parameters of problem (3.16) to

$$a = 1, \quad u_1 = 0, \quad u_3 = 3.$$

Note that in this case the constraint set is $\mathcal{C} = \mathcal{B} = \{0\} \times [1, 3] \times [1, 3]$. The sequence $\{\delta y^k\}_{k\in\mathbb{N}}$ generated by Algorithm 3.1 converges to $\delta y = (2/3, -2/3, -2/3)$, and we have

$$A^T \delta y = 0 \quad \text{and} \quad S_{\mathcal{C}}(\delta y) = -4/3 < 0.$$

According to Proposition 3.3(i), δy is a certificate of primal infeasibility for the problem. Figure 3.2 shows convergence of $\{\delta y^k\}_{k\in\mathbb{N}}$ to a certificate of primal infeasibility.

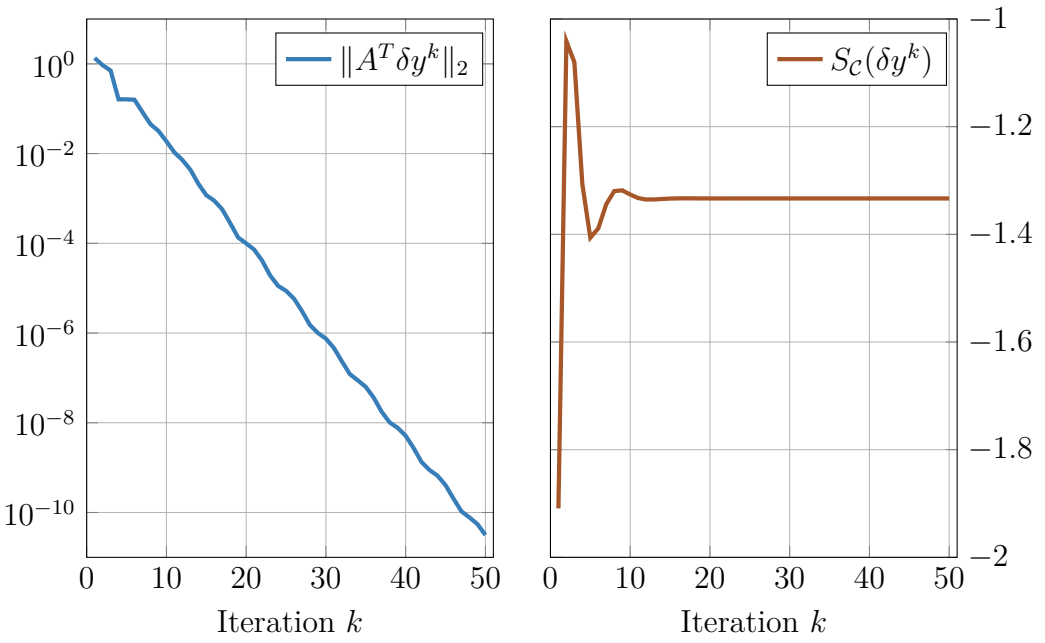


Figure 3.2: Convergence of $\{\delta y^k\}_{k\in\mathbb{N}}$ to a certificate of primal infeasibility for problem (3.16) with $a = 1$, $u_1 = 0$ and $u_3 = 3$.

Dual infeasibility. We set the parameters to

$$a = 0, \quad u_1 = 2, \quad u_3 = +\infty.$$

The constraint set has the form $\mathcal{C} = \mathcal{B} \times \mathcal{K}_b$ with

$$\mathcal{B} = [0, 2] \times [1, 3], \quad \mathcal{K} = \mathbb{R}_+, \quad b = 1,$$

and the constraint matrix A can be written as

$$A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix} \quad \text{with} \quad A_1 = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \quad \text{and} \quad A_2 = \begin{bmatrix} 0 & 1 \end{bmatrix}. \tag{3.17}$$

The sequence $\{\delta x^k\}_{k\in\mathbb{N}}$ generated by Algorithm 3.1 converges to $\delta x = (0, 1/2)$, and we have

$$P\delta x = 0, \quad A_1\delta x = 0, \quad A_2\delta x = 1/2 \in \mathcal{K}, \quad \langle q, \delta x\rangle = -1/2 < 0.$$

According to Proposition 3.3(ii), δx is a certificate of dual infeasibility of the problem. Figure 3.3 shows convergence of $\{\delta x^k\}_{k\in\mathbb{N}}$ to a certificate of dual infeasibility, where $\text{dist}_{\mathcal{C}^\infty}$ denotes the Euclidean distance to the set $\mathcal{C}^\infty = \{0\} \times \{0\} \times \mathbb{R}_+$.

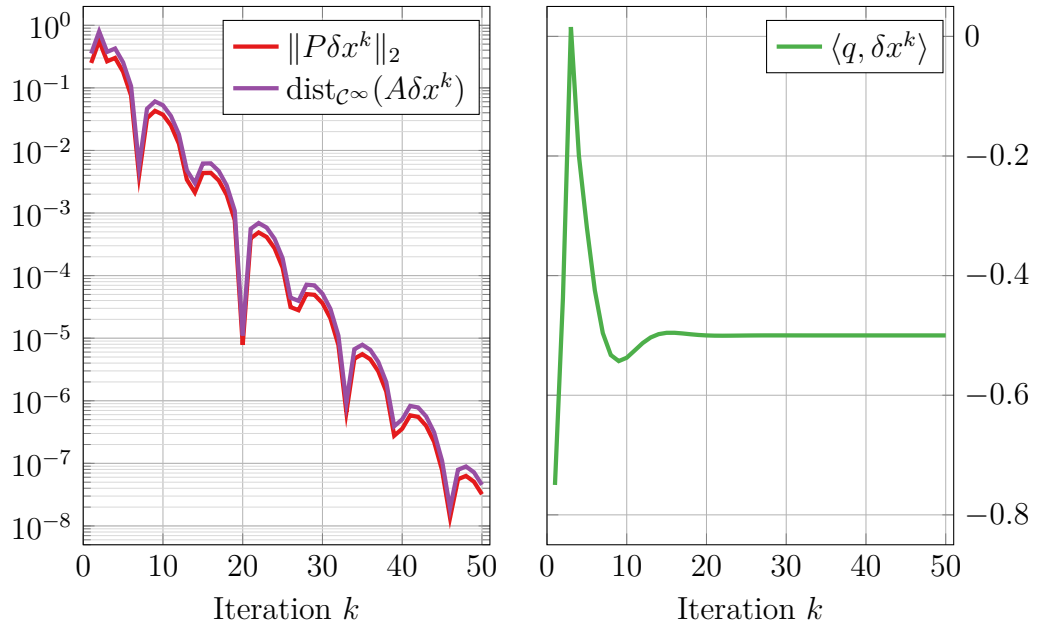


Figure 3.3: Convergence of $\{\delta x^k\}_{k\in\mathbb{N}}$ to a certificate of dual infeasibility for problem (3.16) with $a = 0$, $u_1 = 2$ and $u_3 = +\infty$.

Simultaneous primal and dual infeasibility. We set

$$a = 0, \quad u_1 = 0, \quad u_3 = +\infty.$$

The constraint set has the form $\mathcal{C} = \mathcal{B} \times \mathcal{K}_b$ with

$$\mathcal{B} = \{0\} \times [1, 3], \quad \mathcal{K} = \mathbb{R}_+, \quad b = 1,$$

and the constraint matrix A can be written as in (3.17). The sequences $\{\delta x^k\}_{k\in\mathbb{N}}$ and $\{\delta y^k\}_{k\in\mathbb{N}}$ generated by Algorithm 3.1 converge to $\delta x = (0, 1/2)$ and $\delta y = (1/2, -1/2, 0)$, respectively. If we partition δy as $\delta y = (\delta y_1, \delta y_2)$ with $\delta y_1 = (1/2, -1/2)$ and $\delta y_2 = 0$, then we have

$$A^T \delta y = 0, \quad S_{\mathcal{C}}(\delta y) = S_{\mathcal{B}}(\delta y_1) + S_{\mathcal{K}_b}(\delta y_2) = -1/2 < 0,$$

and

$$P\delta x = 0, \quad A_1 \delta x = 0, \quad A_2 \delta x = 1/2 \in \mathcal{K}, \quad \langle q, \delta x \rangle = -1/2 < 0.$$

Therefore, δx and δy are certificates that the problem is simultaneously primal and dual infeasible. Figure 3.4 shows convergence of $\{\delta y^k\}_{k\in\mathbb{N}}$ and $\{\delta x^k\}_{k\in\mathbb{N}}$ to certificates of primal and dual infeasibility, respectively.

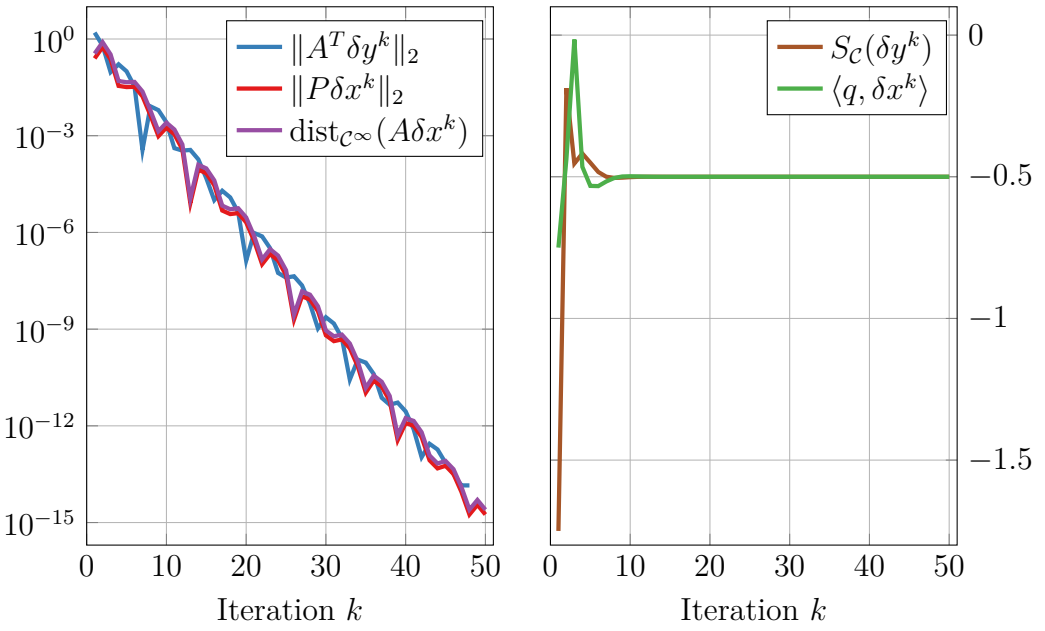


Figure 3.4: Convergence of $\{\delta y^k\}_{k\in\mathbb{N}}$ and $\{\delta x^k\}_{k\in\mathbb{N}}$ to certificates of primal and dual infeasibility, respectively, for problem (3.16) with $a = 0$, $u_1 = 0$ and $u_3 = +\infty$.

3.5.2 Infeasible SDPs from SDPLIB

We next demonstrate the asymptotic behavior of Algorithm 3.1 on two infeasible SDPs from the benchmark library SDPLIB [40]. The problems are given in the following form:

$$\begin{aligned} &\text{minimize} && q^T x \\ &\text{subject to} && Ax = z, \quad z \in \mathcal{S}_b^m, \end{aligned}$$

where $\mathcal{S}^m$ denotes the vectorized form of $\mathbb{S}_+^m$, *i.e.* $z \in \mathcal{S}^m$ is equivalent to $\mathrm{mat}(z) \in \mathbb{S}_+^m$, and $\mathcal{S}_b^m := \mathcal{S}^m + \{b\}$.

Let $X \in \mathbb{S}^m$ have the following eigenvalue decomposition:

$$X = U\,\mathrm{diag}(\lambda_1, \ldots, \lambda_m)U^T.$$

Then the projection of X onto $\mathbb{S}_+^m$ is

$$\Pi_{\mathbb{S}_+^m}(X) = U\,\mathrm{diag}\left(\max(\lambda_1, 0), \ldots, \max(\lambda_m, 0)\right)U^T.$$

Primal infeasible SDP. The primal infeasible problem `infp1` from SDPLIB has decision variables $x \in \mathbb{R}^{10}$ and $z \in \mathcal{S}^{30}$. We run Algorithm 3.1 with parameters $\alpha = 1$ and $\rho = \sigma = 0.1$ from the initial iterate $(x^0, z^0, y^0) = (0, 0, 0)$. Figure 3.5 shows convergence of $\{\delta y^k\}_{k\in\mathbb{N}}$ to a certificate of primal infeasibility, where $\mathrm{dist}_{\mathbb{S}_+^m}$ denotes the spectral norm distance to the positive semidefinite cone $\mathbb{S}_+^m$.

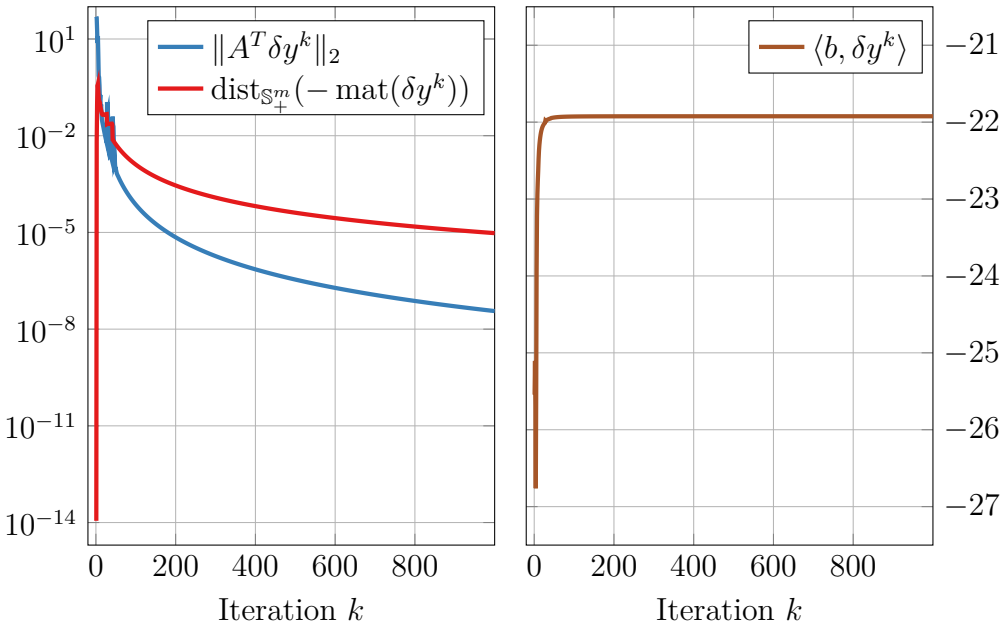


Figure 3.5: Convergence of $\{\delta y^k\}_{k\in\mathbb{N}}$ to a certificate of primal infeasibility for problem `infp1` from SDPLIB.

Dual infeasible SDP. Dual infeasible problem `infd1` from SDPLIB has decision variables $x \in \mathbb{R}^{10}$ and $z \in \mathcal{S}^{30}$. We run Algorithm 3.1 with parameters $\alpha = 1$ and $\rho = \sigma = 0.001$ from the initial iterate $(x^0, z^0, y^0) = (0, 0, 0)$. Figure 3.6 shows convergence of $\{\delta x^k\}_{k\in\mathbb{N}}$ to a certificate of dual infeasibility.

3.5.3 Infeasible SDP with no certificate

Consider the following feasibility problem [143, Ex. 5]:

$$\begin{array}{ll} \text{minimize} & 0 \\ \text{subject to} & \begin{bmatrix} x_1 & 1 & 0 \\ 1 & x_2 & 0 \\ 0 & 0 & -x_1 \end{bmatrix} \succeq 0, \end{array} \tag{3.18}$$

noting that it is primal infeasible by inspection. If we write the constraint set in (3.18) as

$$\underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -1 \end{bmatrix}}_{A_1} x_1 + \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{A_2} x_2 + \underbrace{\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{A_0} \succeq 0$$

and denote by

$$A = [\mathrm{vec}(A_1)\ \mathrm{vec}(A_2)] \quad \text{and} \quad b = -\mathrm{vec}(A_0),$$

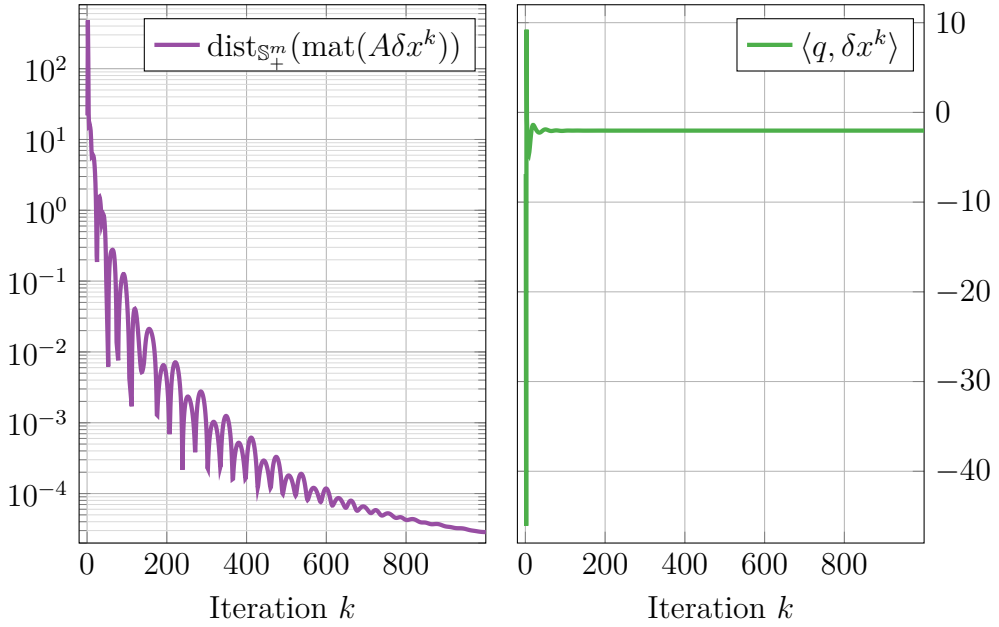


Figure 3.6: Convergence of $\{\delta x^k\}_{k\in\mathbb{N}}$ to a certificate of dual infeasibility for problem `infd1` from SDPLIB.

then the constraint can be written as $Ax \in \mathcal{S}^3_b$, where $\mathcal{S}^3$ denotes the vectorized form of $\mathbb{S}^3_+$. If we define $Y := \text{mat}(y)$, then the primal infeasibility conditions (3.6) for the above problem amount to

$$Y_{11} - Y_{33} = 0, \quad Y_{22} = 0, \quad Y_{12} < 0, \quad Y \preceq 0,$$

where Y_{ij} denotes the element of $Y \in \mathbb{S}^3$ in the i-th row and j-th column. Given that $Y \preceq 0$ and $Y_{22} = 0$ imply $Y_{12} = 0$, the above system is infeasible as well. Note that $Y = 0$ is a feasible point for the dual of problem (3.18), and problem (3.18) is thus not dual infeasible.

We next show that $(\delta x^k, \delta Z^k, \delta Y^k) \to 0$, where $\delta Z^k := \text{mat}(\delta z^k)$ and $\delta Y^k := \text{mat}(\delta y^k)$. Set $x^k = ((1+\rho\sigma^{-1})\varepsilon, \varepsilon^{-1})$ and $V^k := \text{mat}(v^k) = \text{diag}(\varepsilon, \varepsilon^{-1}, 0)$ where $\varepsilon > 0$. The iteration (3.10) then produces the following iterates:

$$Z^k = V^k, \quad \tilde{x}^k = (\varepsilon, \varepsilon^{-1}), \quad \tilde{Z}^k = \text{diag}(\varepsilon, \varepsilon^{-1}, -\varepsilon),$$

and therefore we have

$$\begin{aligned}\delta x^{k+1} &= \alpha\,(\tilde{x}^k - x^k) = \alpha\,(-\rho\sigma^{-1}\varepsilon, 0),\\ \delta V^{k+1} &= \alpha\,(\tilde{Z}^k - Z^k) = \alpha\ \text{diag}(0, 0, -\varepsilon).\end{aligned}$$

By taking ε arbitrarily small, we can make $(\delta x^{k+1}, \delta V^{k+1})$ arbitrarily close to zero, which according to Lemma 3.5(ii) means that $(\delta x^k, \delta V^k) \to (\delta x, \delta V) = 0$, and according to Proposition 3.9 the optimality conditions (3.4) are satisfied in

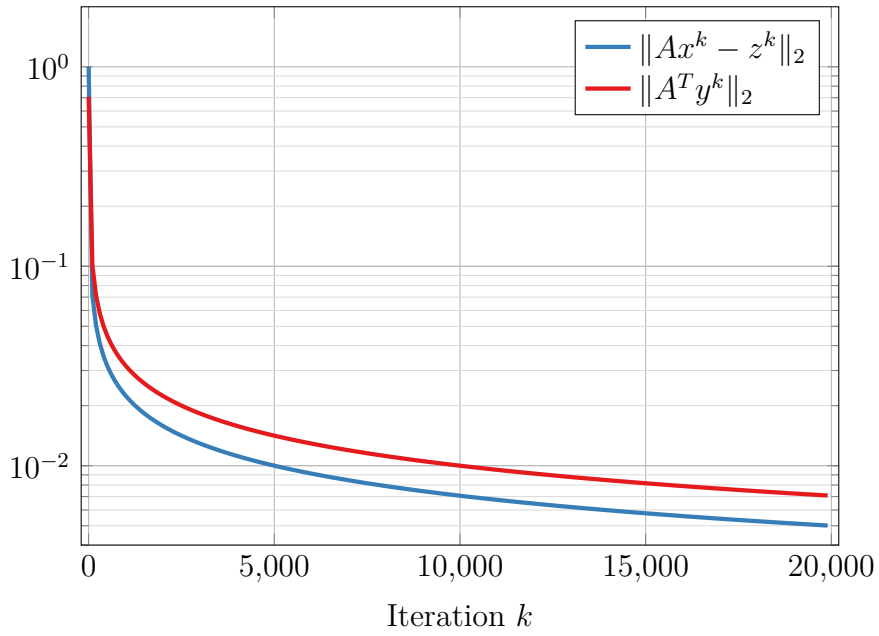


Figure 3.7: Convergence of residuals $\|Ax^k - z^k\|_2$ and $\|A^T y^k\|_2$ for problem (3.18).

the limit. However, the sequence $\{x^k, Z^k, Y^k\}_{k\in\mathbb{N}}$ does not have a limit point; otherwise, such a point would be a certificate for optimality of the problem. Let T denote the fixed-point operator mapping (x^k, V^k) to (x^{k+1}, V^{k+1}). Since $(\delta x, \delta V) \in \overline{\operatorname{ran}(T - \mathrm{Id})}$ by definition, and $(\delta x, \delta V) \notin \operatorname{ran}(T - \mathrm{Id})$, this means that the set $\operatorname{ran}(T - \mathrm{Id})$ is not closed, and the distance from $(\delta x, \delta V)$ to $\operatorname{ran}(T - \mathrm{Id})$ is zero. In other words, the set of matrices in (3.18) and the semidefinite cone $\mathbb{S}^3_+$ do not intersect, but are not strongly separable.

We run Algorithm 3.1 with parameters $\alpha = \rho = \sigma = 1$ from the initial iterate $(x^0, z^0, y^0) = (0, 0, 0)$. Figure 3.7 shows convergence of residuals $\|Ax^k - z^k\|_2$ and $\|A^T y^k\|_2$ to zero.

Remark 3.12. *Let $\varepsilon > 0$. Consider the following perturbation of problem* (3.18)*:*

$$\begin{array}{ll} \text{minimize} & 0 \\ \text{subject to} & \begin{bmatrix} x_1 & 1 & 0 \\ 1 & x_2 & 0 \\ 0 & 0 & -x_1 \end{bmatrix} \succeq -\varepsilon I. \end{array}$$

This problem is feasible since the constraint above is satisfied for $x_1 = 0$ and $x_2 = 1/\varepsilon - \varepsilon$.

Consider now the following problem:

$$\begin{array}{ll} \text{minimize} & 0 \\ \text{subject to} & \begin{bmatrix} x_1 & 1 & 0 \\ 1 & x_2 & 0 \\ 0 & 0 & -x_1 \end{bmatrix} \succeq \varepsilon I. \end{array}$$

This problem is strongly infeasible since the vector $\bar{y} = \text{vec}\,(\text{diag}(-1, 0, -1))$ *satisfies primal infeasibility conditions* (3.6).

These two examples show that an infinitesimally small perturbation of problem (3.18) *can make the problem feasible or strongly infeasible.*

3.6 Conclusions

We have analyzed the asymptotic behavior of ADMM for solving a wide class of convex optimization problems, and have shown that if the sequence of successive differences of the algorithm's iterates does not converge to zero then the problem is primal and/or dual infeasible. Based on these results, we have proposed termination criteria for detecting primal and dual infeasibility, providing for the first time a set of reliable and generic stopping criteria applicable to infeasible convex problems for ADMM. We have also provided numerical examples to demonstrate different asymptotic behaviors of the algorithm's iterates.

3.A Auxiliary results

Lemma 3.13. *For any vectors $v \in \mathbb{R}^n$, $b \in \mathbb{R}^n$ and a nonempty, closed, and convex cone $\mathcal{K} \subseteq \mathbb{R}^n$,*

(i) $\Pi_{\mathcal{K}_b}(v) = b + \Pi_{\mathcal{K}}(v - b)$.

(ii) $(\mathrm{Id} - \Pi_{\mathcal{K}_b})(v) = \Pi_{\mathcal{K}^\circ}(v - b)$.

(iii) $\langle \Pi_{\mathcal{K}_b}(v), (\mathrm{Id} - \Pi_{\mathcal{K}_b})(v) \rangle = \langle b, \Pi_{\mathcal{K}^\circ}(v - b) \rangle$.

(iv) $\langle \Pi_{\mathcal{K}}(v), v \rangle = \|\Pi_{\mathcal{K}}(v)\|^2$.

Proof. Part (i) is from [17, Prop. 28.1(i)].

(ii): From part (i) we have

$$(\mathrm{Id} - \Pi_{\mathcal{K}_b})(v) = v - b - \Pi_{\mathcal{K}}(v - b) = \Pi_{\mathcal{K}^\circ}(v - b),$$

where the second equality follows from the Moreau decomposition.

(iii): Follows directly from parts (i) and (ii), and the Moreau decomposition.

(iv): From the Moreau decomposition, we have

$$\langle \Pi_{\mathcal{K}}(v), v \rangle = \langle \Pi_{\mathcal{K}}(v), \Pi_{\mathcal{K}}(v) + \Pi_{\mathcal{K}^\circ}(v) \rangle = \|\Pi_{\mathcal{K}}(v)\|^2. \qquad \square$$

Lemma 3.14. *Suppose that $\mathcal{K} \subseteq \mathbb{R}^n$ is a nonempty, closed, and convex cone and for some sequence $\{v^k\}_{k \in \mathbb{N}}$, where $v^k \in \mathbb{R}^n$, we denote by $\delta v \coloneqq \lim_{k \to \infty} \frac{1}{k} v^k$, assuming that the limit exists. Then for any $b \in \mathbb{R}^n$,*

$$\lim_{k \to \infty} \tfrac{1}{k} \Pi_{\mathcal{K}_b}(v^k) = \Pi_{\mathcal{K}}(\delta v).$$

Proof. Write the limit as

$$\begin{aligned} \lim_{k \to \infty} \tfrac{1}{k} \Pi_{\mathcal{K}_b}(v^k) &= \lim_{k \to \infty} \tfrac{1}{k} \left(b + \Pi_{\mathcal{K}}(v^k - b) \right) \\ &= \lim_{k \to \infty} \Pi_{\mathcal{K}} \left(\tfrac{1}{k} (v^k - b) \right) \\ &= \Pi_{\mathcal{K}} \left(\lim_{k \to \infty} \tfrac{1}{k} v^k \right), \end{aligned}$$

where the first equality uses Lemma 3.13(i), and the second and third follow from the positive homogeneity [17, Prop. 28.22] and continuity [17, Prop. 4.8] of $\Pi_{\mathcal{K}}$, respectively. $\square$

Lemma 3.15. *Suppose that $\mathcal{B} \subseteq \mathbb{R}^n$ is a nonempty, convex and compact set and for some sequence $\{v^k\}_{k \in \mathbb{N}}$, where $v^k \in \mathbb{R}^n$, we denote by $\delta v \coloneqq \lim_{k \to \infty} \frac{1}{k} v^k$, assuming that the limit exists. Then*

$$\lim_{k \to \infty} \tfrac{1}{k} \left\langle v^k, \Pi_{\mathcal{B}}(v^k) \right\rangle = \lim_{k \to \infty} \left\langle \delta v, \Pi_{\mathcal{B}}(v^k) \right\rangle = S_{\mathcal{B}}(\delta v).$$

Proof. Let $z^k \coloneqq \Pi_{\mathcal{B}}(v^k)$. We have the following inclusion [17, Prop. 6.46]

$$v^k - z^k \in N_{\mathcal{B}}(z^k),$$

which is equivalent to [17, Thm. 16.23]

$$\left\langle \tfrac{1}{k}(v^k - z^k), z^k \right\rangle = S_{\mathcal{B}}\left(\tfrac{1}{k}(v^k - z^k)\right).$$

Taking the limit of the above identity, we obtain

$$\begin{aligned} \lim_{k\to\infty} \left\langle \tfrac{1}{k}(v^k - z^k), z^k \right\rangle &= \lim_{k\to\infty} S_{\mathcal{B}}\left(\tfrac{1}{k}(v^k - z^k)\right) \\ &= S_{\mathcal{B}}\Big(\lim_{k\to\infty} \tfrac{1}{k}(v^k - z^k)\Big) \\ &= S_{\mathcal{B}}(\delta v), \end{aligned} \tag{3.19}$$

where the second equality follows from the continuity of $S_{\mathcal{B}}$ [17, Ex. 11.2], and the third from the compactness of $\mathcal{B}$. Since $\{z^k\}_{k\in\mathbb{N}}$ remains in the compact set $\mathcal{B}$, we can derive the following relation from (3.19):

$$\begin{aligned} \left| S_{\mathcal{B}}(\delta v) - \lim_{k\to\infty} \left\langle \delta v, z^k \right\rangle \right| &= \left| \lim_{k\to\infty} \left\langle \tfrac{1}{k}(v^k - z^k), z^k \right\rangle - \left\langle \delta v, z^k \right\rangle \right| \\ &= \left| \lim_{k\to\infty} \left\langle \tfrac{1}{k} v^k - \delta v, z^k \right\rangle - \tfrac{1}{k} \left\langle z^k, z^k \right\rangle \right| \\ &\leq \lim_{k\to\infty} \underbrace{\left\| \tfrac{1}{k} v^k - \delta v \right\|}_{\to 0} \|z^k\| + \tfrac{1}{k}\|z^k\|^2 \\ &= 0, \end{aligned}$$

where the third row follows from the triangle and Cauchy-Schwarz inequalities, and the fourth from the compactness of $\mathcal{B}$. Finally, we can derive the following identity from (3.19):

$$S_{\mathcal{B}}(\delta v) = \lim_{k\to\infty} \left\langle \tfrac{1}{k}(v^k - z^k), z^k \right\rangle = \lim_{k\to\infty} \left\langle \tfrac{1}{k} v^k, z^k \right\rangle - \underbrace{\tfrac{1}{k}\|z^k\|^2}_{\to 0}.$$

This concludes the proof. □

3.B Proofs

3.B.1 Proof of Proposition 3.7

Commensurate with our partitioning of the constraint set as $\mathcal{C} = \mathcal{B} \times \mathcal{K}_b$, we partition the matrix A and the iterates (v^k, z^k, y^k) into components of appropriate dimensions. We use subscript 1 for those components associated with the set $\mathcal{B}$ and subscript 2 for those associated with the set $\mathcal{K}_b$, *e.g.* $z^k = (z_1^k, z_2^k)$ where $z_1 \in \mathcal{B}$ and $z_2 \in \mathcal{K}_b$ and the matrix $A = \left[\begin{smallmatrix} A_1 \\ A_2 \end{smallmatrix}\right]$. Note throughout that $\mathcal{C}^\infty = \{0\} \times \mathcal{K}$ and $(\mathcal{C}^\infty)^\circ = \mathbb{R}^{m_1} \times \mathcal{K}^\circ$.

Proof of (i)

Divide (3.14a) by k, take the limit and apply Lemma 3.5 to get

$$A\delta x = \lim_{k\to\infty} \tfrac{1}{k}\Pi_{\mathcal{C}}(v^k).$$

Due to Lemma 3.14 and the compactness of $\mathcal{B}$, we then obtain

$$\begin{aligned} A_1\delta x &= \lim_{k\to\infty} \tfrac{1}{k}\Pi_{\mathcal{B}}(v_1^k) = 0, \\ A_2\delta x &= \lim_{k\to\infty} \tfrac{1}{k}\Pi_{\mathcal{K}_b}(v_2^k) = \Pi_{\mathcal{K}}(\delta v_2). \end{aligned}$$

Proof of (ii)

Divide (3.14b) by ρk, take the inner product of both sides with δx and take the limit to obtain

$$\begin{aligned} -\rho^{-1}\langle P\delta x, \delta x\rangle &= \lim_{k\to\infty} \left\langle A\delta x, \tfrac{1}{k}v_k - \tfrac{1}{k}\Pi_{\mathcal{C}}(v^k)\right\rangle \\ &= \left\langle A_1\delta x, \delta v_1 - \lim_{k\to\infty}\tfrac{1}{k}\Pi_{\mathcal{B}}(v_1^k)\right\rangle + \left\langle A_2\delta x, \delta v_2 - \lim_{k\to\infty}\tfrac{1}{k}\Pi_{\mathcal{K}_b}(v_2^k)\right\rangle \\ &= \langle \Pi_{\mathcal{K}}(\delta v_2), \delta v_2 - \Pi_{\mathcal{K}}(\delta v_2)\rangle \\ &= \langle \Pi_{\mathcal{K}}(\delta v_2), \Pi_{\mathcal{K}^\circ}(\delta v_2)\rangle \\ &= 0, \end{aligned}$$

where we used Lemma 3.5 in the second equality, $A_1\delta x = 0$, $A_2\delta x = \Pi_{\mathcal{K}}(\delta v_2)$, the compactness of $\mathcal{B}$ and Lemma 3.14 in the third, and the Moreau decomposition in the fourth and fifth. Then $P\delta x = 0$ since $P \in \mathbb{S}^n_+$.

Proof of (iii)

Divide (3.14b) by k, take the limit and use $P\delta x = 0$ to obtain

$$\begin{aligned} 0 &= \lim_{k\to\infty} \tfrac{1}{k}\rho A^T(\mathrm{Id} - \Pi_{\mathcal{C}})(v^k) \\ &= \lim_{k\to\infty} \tfrac{1}{k}\rho\left(A_1^T(\mathrm{Id} - \Pi_{\mathcal{B}})(v_1^k) + A_2^T(\mathrm{Id} - \Pi_{\mathcal{K}_b})(v_2^k)\right) \\ &= \rho A_1^T\left(\delta v_1 - \lim_{k\to\infty}\tfrac{1}{k}\Pi_{\mathcal{B}}(v_1^k)\right) + \rho A_2^T\left(\delta v_2 - \lim_{k\to\infty}\tfrac{1}{k}\Pi_{\mathcal{K}_b}(v_2^k)\right) \\ &= \rho A_1^T\delta v_1 + \rho A_2^T\left(\delta v_2 - \Pi_{\mathcal{K}}(\delta v_2)\right) \\ &= \rho A_1^T\delta v_1 + \rho A_2^T\Pi_{\mathcal{K}^\circ}(\delta v_2) \\ &= \rho\begin{bmatrix} A_1 \\ A_2\end{bmatrix}^T\begin{bmatrix} \delta v_1 \\ \Pi_{\mathcal{K}^\circ}(\delta v_2)\end{bmatrix} \\ &= A^T\rho\Pi_{(\mathcal{C}^\infty)^\circ}(\delta v) \\ &= A^T\delta y, \end{aligned}$$

where we used Lemma 3.5 in the third equality, Lemma 3.14 and the compactness of $\mathcal{B}$ in the fourth, the Moreau decomposition in the fifth, and $(\mathcal{C}^\infty)^\circ = \mathbb{R}^{m_1} \times \mathcal{K}^\circ$ in the seventh.

Proof of (iv)

We first show that the sequence $\{\delta z^k\}_{k\in\mathbb{N}}$ converges to δz. From (3.13) we have

$$-\alpha^{-1}\left(\delta x^{k+1} - \delta x^k\right) = \delta x^k - \delta \tilde{x}^k, \tag{3.20a}$$

$$-\alpha^{-1}\left(\delta v^{k+1} - \delta v^k\right) = \delta z^k - \delta \tilde{z}^k. \tag{3.20b}$$

Take the limit of (3.20a) to obtain

$$\lim_{k\to\infty} \delta \tilde{x}^k = \lim_{k\to\infty} \delta x^k = \delta x.$$

From (3.12) we now have $\delta \tilde{z}^k = A\delta \tilde{x}^k \to A\delta x$. Take the limit of (3.20b) and use the result from (i) to obtain

$$\lim_{k\to\infty} \delta z^k = \lim_{k\to\infty} \delta \tilde{z}^k = A\delta x = \Pi_{\mathcal{C}^\infty}(\delta v).$$

We now show that the sequence $\{\frac{1}{k} z^k\}_{k\in\mathbb{N}}$ also converges to δz. Dividing the expression for z^k in (3.11) by k and taking the limit, we obtain

$$\lim_{k\to\infty} \tfrac{1}{k} z^k = \lim_{k\to\infty} \tfrac{1}{k} \begin{bmatrix} \Pi_{\mathcal{B}}(v_1^k) \\ \Pi_{\mathcal{K}_b}(v_2^k) \end{bmatrix} = \begin{bmatrix} 0 \\ \Pi_{\mathcal{K}}(\delta v_2) \end{bmatrix} = \Pi_{\mathcal{C}^\infty}(\delta v).$$

Proof of (v)

We first show that the sequence $\{\delta y^k\}_{k\in\mathbb{N}}$ converges to δy. Since $\rho^{-1} y^k = v^k - z^k$, we have

$$\begin{aligned} \lim_{k\to\infty} \rho^{-1}\delta y^k &= \lim_{k\to\infty} \delta v^k - \lim_{k\to\infty} \delta z^k \\ &= \delta v - \Pi_{\mathcal{C}^\infty}(\delta v) \\ &= \Pi_{(\mathcal{C}^\infty)^\circ}(\delta v), \end{aligned}$$

where we used the Moreau decomposition in the last equality.

We now show that the sequence $\{\frac{1}{k} y^k\}_{k\in\mathbb{N}}$ also converges to δy. Dividing the expression for y^k in (3.11) by k and taking the limit, we obtain

$$\lim_{k\to\infty} \tfrac{1}{k} y^k = \rho \lim_{k\to\infty} \tfrac{1}{k}(v^k - z^k) = \rho(\delta v - \delta z) = \rho(\delta v - \Pi_{\mathcal{C}^\infty}(\delta v)) = \rho \Pi_{(\mathcal{C}^\infty)^\circ}(\delta v).$$

3.B.2 Proof of Proposition 3.8

Take the inner product of both sides of (3.14b) with δx and use Proposition 3.7(ii) to obtain

$$\langle \delta x, q\rangle + \rho\left\langle A\delta x, (\mathrm{Id} - \Pi_{\mathcal{C}})(v^k)\right\rangle = -\sigma\alpha^{-1}\left\langle \delta x, \delta x^{k+1}\right\rangle - \rho\alpha^{-1}\left\langle A\delta x, \delta v^{k+1}\right\rangle.$$

Using $A_1\delta x = 0$ from Proposition 3.7(i) and then taking the limit gives

$$\begin{aligned}
\langle q, \delta x\rangle &= -\sigma\alpha^{-1}\|\delta x\|^2 - \rho\alpha^{-1}\langle A\delta x, \delta v\rangle - \rho\lim_{k\to\infty}\left\langle A_2\delta x, \Pi_{\mathcal{K}^\circ}(v_2^k - b)\right\rangle \\
&= -\sigma\alpha^{-1}\|\delta x\|^2 - \rho\alpha^{-1}\langle \Pi_{\mathcal{C}^\infty}(\delta v), \delta v_2\rangle \\
&\quad -\rho\lim_{k\to\infty}\left\langle \Pi_{\mathcal{K}}(\delta v_2), \Pi_{\mathcal{K}^\circ}(v_2^k - b)\right\rangle \qquad (3.21)\\
&= -\sigma\alpha^{-1}\|\delta x\|^2 - \rho\alpha^{-1}\|\Pi_{\mathcal{C}^\infty}(\delta v)\|^2 \\
&\quad -\rho\lim_{k\to\infty}\left\langle \Pi_{\mathcal{K}}(\delta v_2), \Pi_{\mathcal{K}^\circ}(v_2^k - b)\right\rangle,
\end{aligned}$$

where we used Lemma 3.13(ii) in the first equality, Proposition 3.7(i) in the second, and Lemma 3.13(iv) in the third.

Now take the inner product of both sides of (3.14a) with $\Pi_{(\mathcal{C}^\infty)^\circ}(\delta v)$ to obtain

$$\begin{aligned}
\alpha^{-1}\left\langle \Pi_{(\mathcal{C}^\infty)^\circ}(\delta v), \delta v^{k+1}\right\rangle &= \left\langle A^T\Pi_{(\mathcal{C}^\infty)^\circ}(\delta v), x^k + \alpha^{-1}\delta x^{k+1}\right\rangle \\
&\quad - \left\langle \Pi_{(\mathcal{C}^\infty)^\circ}(\delta v), \Pi_{\mathcal{C}}(v^k)\right\rangle.
\end{aligned}$$

According to Proposition 3.7(iii) the first inner product on the right-hand side is zero, and taking the limit we obtain

$$\begin{aligned}
\lim_{k\to\infty}\left\langle \Pi_{(\mathcal{C}^\infty)^\circ}(\delta v), \Pi_{\mathcal{C}}(v^k)\right\rangle &= -\alpha^{-1}\left\langle \Pi_{(\mathcal{C}^\infty)^\circ}(\delta v), \delta v\right\rangle \\
&= -\alpha^{-1}\|\Pi_{(\mathcal{C}^\infty)^\circ}(\delta v)\|^2,
\end{aligned}$$

where the second equality follows from Lemma 3.13(iv). The limit in the identity above can be expressed as

$$\begin{aligned}
\lim_{k\to\infty}\left\langle \Pi_{(\mathcal{C}^\infty)^\circ}(\delta v), \Pi_{\mathcal{C}}(v^k)\right\rangle &= \lim_{k\to\infty}\left\langle \delta v_1, \Pi_{\mathcal{B}}(v_1^k)\right\rangle + \lim_{k\to\infty}\left\langle \Pi_{\mathcal{K}^\circ}(\delta v_2), \Pi_{\mathcal{K}_b}(v_2^k)\right\rangle \\
&= S_{\mathcal{B}}(\delta v_1) + \lim_{k\to\infty}\left\langle \Pi_{\mathcal{K}^\circ}(\delta v_2), b + \Pi_{\mathcal{K}}(v_2^k - b)\right\rangle \\
&= S_{\mathcal{B}}(\delta v_1) + \langle \Pi_{\mathcal{K}^\circ}(\delta v_2), b\rangle \\
&\quad + \lim_{k\to\infty}\left\langle \Pi_{\mathcal{K}^\circ}(\delta v_2), \Pi_{\mathcal{K}}(v_2^k - b)\right\rangle,
\end{aligned}$$

where the second equality follows from Lemma 3.13(i) and Lemma 3.15. Since the support function of $\mathcal{K}$ evaluated at any point in $\mathcal{K}^\circ$ is zero, we can write

$$\langle b, \Pi_{\mathcal{K}^\circ}(\delta v_2)\rangle = \langle b, \Pi_{\mathcal{K}^\circ}(\delta v_2)\rangle + \sup_{z\in\mathcal{K}}\langle z, \Pi_{\mathcal{K}^\circ}(\delta v_2)\rangle = S_{\mathcal{K}_b}(\Pi_{\mathcal{K}^\circ}(\delta v_2)).$$

Now we have

$$\begin{aligned} S_{\mathcal{C}}(\Pi_{(\mathcal{C}^\infty)^\circ}(\delta v)) &= S_{\mathcal{B}}(\delta v_1) + S_{\mathcal{K}_b}(\Pi_{\mathcal{K}^\circ}(\delta v_2)) \\ &= -\alpha^{-1}\|\Pi_{(\mathcal{C}^\infty)^\circ}(\delta v)\|^2 - \lim_{k\to\infty} \left\langle \Pi_{\mathcal{K}^\circ}(\delta v_2), \Pi_{\mathcal{K}}(v_2^k - b) \right\rangle, \end{aligned}$$

and due to the positive homogeneity of the support function,

$$S_{\mathcal{C}}(\delta y) = -\rho\alpha^{-1}\|\Pi_{(\mathcal{C}^\infty)^\circ}(\delta v)\|^2 - \rho \lim_{k\to\infty} \left\langle \Pi_{\mathcal{K}^\circ}(\delta v_2), \Pi_{\mathcal{K}}(v_2^k - b) \right\rangle. \tag{3.22}$$

We will next show that the limits in (3.21) and (3.22) are equal to zero. Summing the two equalities, we obtain

$$\begin{aligned} \langle q, \delta x\rangle + S_{\mathcal{C}}(\delta y) + \sigma\alpha^{-1}\|\delta x\|^2 + \rho\alpha^{-1}\|\delta v\|^2 & \\ = -\rho \lim_{k\to\infty} \left\langle \Pi_{\mathcal{K}}(\delta v_2), \Pi_{\mathcal{K}^\circ}(v_2^k - b) \right\rangle & \\ -\rho \lim_{k\to\infty} \left\langle \Pi_{\mathcal{K}^\circ}(\delta v_2), \Pi_{\mathcal{K}}(v_2^k - b) \right\rangle, & \end{aligned} \tag{3.23}$$

where we used $\|\delta v\|^2 = \|\Pi_{\mathcal{C}^\infty}(\delta v)\|^2 + \|\Pi_{(\mathcal{C}^\infty)^\circ}(\delta v)\|^2$ [17, Thm. 6.29].

Take the inner product of both sides of (3.14b) with x^k to obtain

$$\begin{aligned} \left\langle Px^k, x^k \right\rangle + \left\langle q, x^k \right\rangle + \rho \left\langle Ax^k, (\mathrm{Id} - \Pi_{\mathcal{C}})(v^k) \right\rangle = -\alpha^{-1} \left\langle P\delta x^{k+1}, x^k \right\rangle & \\ -\sigma\alpha^{-1} \left\langle \delta x^{k+1}, x^k \right\rangle & \\ -\rho\alpha^{-1} \left\langle Ax^k, \delta v^{k+1} \right\rangle. & \end{aligned} \tag{3.24}$$

We can rewrite the third inner product on the left-hand side of the equality above as

$$\begin{aligned} \left\langle Ax^k, (\mathrm{Id} - \Pi_{\mathcal{C}})(v^k) \right\rangle &= \left\langle \Pi_{\mathcal{C}}(v^k) + \alpha^{-1}\left(\delta v^{k+1} - A\delta x^{k+1}\right), (\mathrm{Id} - \Pi_{\mathcal{C}})(v^k) \right\rangle \\ &= \left\langle \Pi_{\mathcal{B}}(v_1^k), v_1^k \right\rangle - \|\Pi_{\mathcal{B}}(v_1^k)\|^2 + \left\langle \Pi_{\mathcal{K}_b}(v_2^k), (\mathrm{Id} - \Pi_{\mathcal{K}_b})(v_2^k) \right\rangle \\ &\quad + \alpha^{-1} \left\langle \delta v^{k+1} - A\delta x^{k+1}, \rho^{-1} y^k \right\rangle \\ &= \left\langle \Pi_{\mathcal{B}}(v_1^k), v_1^k \right\rangle - \|\Pi_{\mathcal{B}}(v_1^k)\|^2 + \left\langle b, \Pi_{\mathcal{K}^\circ}(v_2^k - b) \right\rangle \\ &\quad + \alpha^{-1} \left\langle \delta v^{k+1} - A\delta x^{k+1}, \rho^{-1} y^k \right\rangle, \end{aligned}$$

where we used (3.14a) in the first equality, (3.11) in the second, and Lemma 3.13(iii) in the third. Substituting this expression into (3.24), dividing by k and taking the limit, we then obtain

$$\begin{aligned} \lim_{k\to\infty} \tfrac{1}{k} \left\langle Px^k, x^k \right\rangle + \langle q, \delta x\rangle + \rho \lim_{k\to\infty} \tfrac{1}{k} \left\langle \Pi_{\mathcal{B}}(v_1^k), v_1^k \right\rangle - \rho \lim_{k\to\infty} \tfrac{1}{k} \|\Pi_{\mathcal{B}}(v_1^k)\|^2 & \\ + \rho \langle b, \Pi_{\mathcal{K}^\circ}(\delta v_2)\rangle + \rho\alpha^{-1} \left\langle \delta v - A\delta x, \rho^{-1}\delta y \right\rangle = -\alpha^{-1} \langle P\delta x, \delta x\rangle & \\ -\sigma\alpha^{-1}\|\delta x\|^2 & \\ -\rho\alpha^{-1} \langle A\delta x, \delta v\rangle. & \end{aligned}$$

Due to Lemma 3.15, Lemma 3.7(ii) and the compactness of $\mathcal{B}$, the equality above simplifies to

$$\begin{aligned}\lim_{k\to\infty} \tfrac{1}{k}\left\langle Px^k, x^k\right\rangle + \langle q, \delta x\rangle + S_{\mathcal{C}}(\delta y) + \sigma\alpha^{-1}\|\delta x\|^2 \\ = -\rho\alpha^{-1}\left\langle \delta v - A\delta x, \rho^{-1}\delta y\right\rangle \\ - \rho\alpha^{-1}\left\langle A\delta x, \delta v\right\rangle.\end{aligned} \tag{3.25}$$

The sum of inner products appearing on the right-hand side of the equality above can be written as

$$\begin{aligned}\left\langle \delta v - A\delta x, \rho^{-1}\delta y\right\rangle + \langle A\delta x, \delta v\rangle &= \left\langle \delta v - \Pi_{\mathcal{C}^\infty}(\delta v), \Pi_{(\mathcal{C}^\infty)^\circ}(\delta v)\right\rangle + \langle \Pi_{\mathcal{C}^\infty}(\delta v), \delta v\rangle \\ &= \|\Pi_{(\mathcal{C}^\infty)^\circ}(\delta v)\|^2 + \|\Pi_{\mathcal{C}^\infty}(\delta v)\|^2 \\ &= \|\delta v\|^2,\end{aligned}$$

where we used Proposition 3.7(i) in the first equality, and Lemma 3.13(iv) and the Moreau decomposition in the second. Substituting this expression into (3.25), we obtain

$$\langle q, \delta x\rangle + S_{\mathcal{C}}(\delta y) + \sigma\alpha^{-1}\|\delta x\|^2 + \rho\alpha^{-1}\|\delta v\|^2 = -\lim_{k\to\infty} \tfrac{1}{k}\left\langle Px^k, x^k\right\rangle. \tag{3.26}$$

Comparing identities in (3.23) and (3.26), we get the following relation:

$$\begin{aligned}\lim_{k\to\infty} \tfrac{1}{k}\left\langle Px^k, x^k\right\rangle = \rho \lim_{k\to\infty} \left\langle \Pi_{\mathcal{K}}(\delta v_2), \Pi_{\mathcal{K}^\circ}(v_2^k - b)\right\rangle \\ + \rho \lim_{k\to\infty} \left\langle \Pi_{\mathcal{K}^\circ}(\delta v_2), \Pi_{\mathcal{K}}(v_2^k - b)\right\rangle.\end{aligned}$$

Positive semidefiniteness of P implies that the sequence on the left-hand side is term-wise nonnegative. Since the two sequences on the right-hand side involve inner products of elements in $\mathcal{K}$ and its polar, each sequence is term-wise nonpositive. Consequently, each of these limits must be zero. The claims of the proposition then follow directly from (3.21) and (3.22).

4

Global Linear Convergence in Operator Splitting Methods

Contents

In this chapter we analyze convergence rate of operator splitting methods via associated fixed-point operators. In particular, we investigate properties of such operators that ensure linear convergence of their fixed-point iterations.

4.1 Introduction

Over the last decade convergence rate analysis of operator splitting methods has attracted significant interest in the control community. This is mainly motivated by the need to provide a computational complexity certificate for optimal control algorithms [148]. The authors in [149] establish linear convergence of the accelerated proximal gradient method (PGM) for solving a restricted class of input-constrained model predictive control (MPC) problems provided that the constraint set is simple enough. Linear convergence of the same algorithm with

an adaptive restart technique is shown in [135, 159]. The alternating direction method of multipliers (ADMM) for solving convex quadratic programs (QPs) was shown to converge linearly provided that the cost matrix is positive definite and the constraint matrix has full row-rank [90]. The authors in [93] extend this result to a more general class of convex problems, while the tightness of the obtained convergence rate is shown in [92]. The authors in [108] establish linear convergence of a distributed version of ADMM under similar assumptions.

All of the aforementioned results on linear convergence require strong convexity and Lipschitz smoothness of a function involved in the optimization problem [130], *e.g.* that the problem can be written in the form

$$\text{minimize} \quad f(x) + g(x), \tag{4.1}$$

where f is strongly convex and Lipschitz smooth, and g is a function whose proximal operator is easy to evaluate. Real-world problems rarely have the aforementioned structure, so current analysis techniques cannot ensure linear convergence of operator splitting methods. This situation has motivated many researchers to look for alternative algorithms and analytical methods that could ensure linear convergence.

The authors in [79] proposed a particular linearly convergent proximal method for linear programming. Linear convergence of some projection methods, which are a special class of operator splitting methods for solving feasibility and best-approximation problems, has been established in certain special cases. In the case of two affine subspaces, the linear convergence rate of the alternating projection method (APM) [64], generalized APM [82], the Douglas-Rachford splitting (DRS) [15], and the generalized DRS [14] is characterized in terms of the Friedrichs angle between the subspaces. The authors in [133] show that the linear convergence rate of APM applied to two convex polyhedra is characterized by the smallest nonzero Friedrichs angle between faces of the polyhedra. The authors in [23] have identified linear regularity as a sufficient condition for global linear convergence of the fixed-point iteration of an averaged operator and provided a linear convergence rate.

In this chapter we establish necessary and sufficient conditions for global linear convergence of the fixed-point iteration of a strongly quasi-nonexpansive (SQNE) operator in Hilbert spaces and provide a tight bound on the convergence rate. This class of operators includes averaged operators as a special case. We demonstrate that some existing results establishing linear convergence in projection methods are special cases of our analysis. We also propose a linearly convergent method for linear programs (LPs) that does not assume feasibility of the problem or boundedness of the objective value.

4.2 Linear convergence via linear regularity

Let $\mathcal{D}$ be a nonempty subset of a real Hilbert space $\mathcal{H}$. We define an operator $T : \mathcal{D} \mapsto \mathcal{D}$ such that the iterates $\{x^k\}_{k\in\mathbb{N}}$ computed by an algorithm from some initial point $x^0 \in \mathcal{D}$ are equivalent to the iteration

$$x^{k+1} \leftarrow Tx^k. \tag{4.2}$$

We refer to (4.2) as a *Picard* or *fixed-point iteration* of the operator T. We make the following assumption throughout the chapter:

Assumption 4.1. *The fixed-point set of T is nonempty.*

The solution set $\mathcal{X}^\star$ of an optimization problem is usually closely related to the fixed-point set of the related operator T. It is often the case that either $\mathcal{X}^\star = \operatorname{Fix} T$ (*e.g.* when T represents PGM [154]), or it is easy to reconstruct $x^\star \in \mathcal{X}^\star$ from some $\overline{x} \in \operatorname{Fix} T$ (*e.g.* when T represents DRS [154]). If this is the case, then solving an optimization problem can be reformulated as a problem of finding a fixed-point of the operator T.

We are particularly interested in showing that the operator T satisfies the condition

$$\operatorname{dist}_{\operatorname{Fix} T}(Tx) \leq \beta \operatorname{dist}_{\operatorname{Fix} T}(x) \tag{4.3}$$

for all $x \in \mathcal{D}$, and for some $\beta \in [0, 1)$ *independent of the choice of* x. The constant β is called the *convergence factor* and determines the linear convergence rate.

In the sequel we will introduce some definitions and results from operator theory that we require in order to analyze the convergence of a fixed-point iteration; we refer the reader to [17] for a comprehensive review.

Definition 4.2 ([17, 49])**.** *Let $\mathcal{D}$ be a nonempty subset of $\mathcal{H}$ and let $T : \mathcal{D} \mapsto \mathcal{H}$. Then T is*

(i) quasi-nonexpansive (QNE) *if* $(\forall x \in \mathcal{D})(\forall y \in \operatorname{Fix} T)$

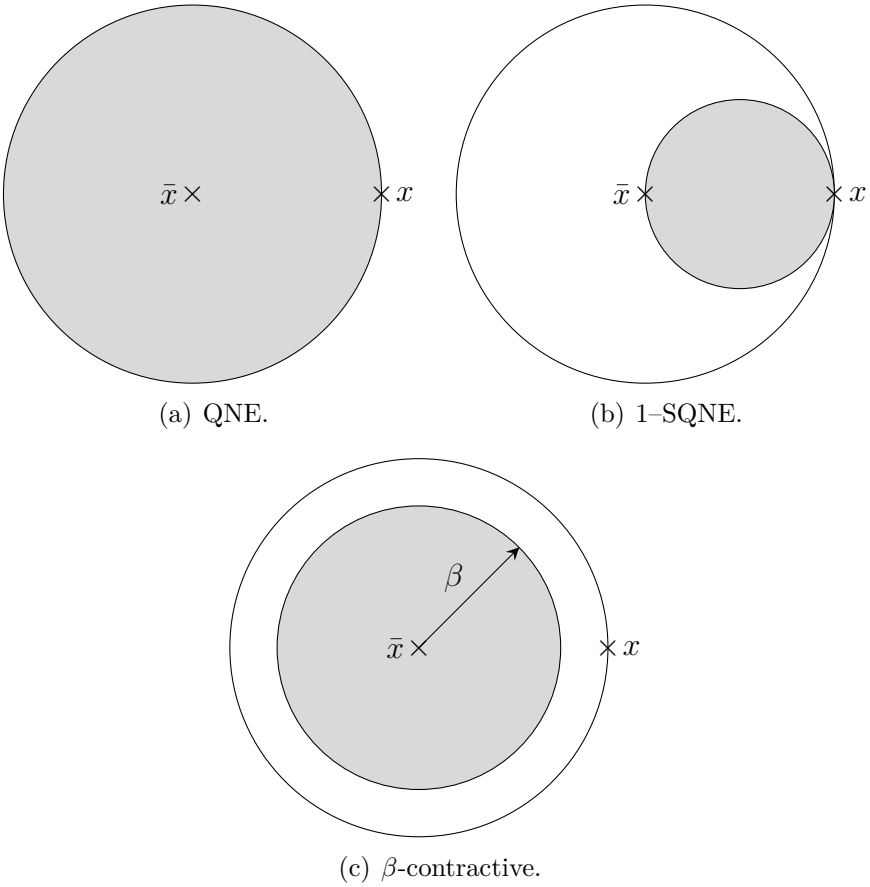
$$\|Tx - y\| \leq \|x - y\|, \tag{4.4a}$$

(ii) ρ-strongly quasi-nonexpansive (ρ–SQNE) *with* $\rho > 0$ *if* $(\forall x \in \mathcal{D})(\forall y \in \operatorname{Fix} T)$

$$\|Tx - y\|^2 \leq \|x - y\|^2 - \rho \|x - Tx\|^2, \tag{4.4b}$$

(iii) β-contractive *with* $\beta \in [0, 1)$ *if* $(\forall x \in \mathcal{D})(\forall y \in \mathcal{D})$

$$\|Tx - Ty\| \leq \beta \|x - y\|. \tag{4.4c}$$



(a) QNE. (b) 1–SQNE.

(c) β-contractive.

Figure 4.1: Graphical representation of some operator properties. Tx lies somewhere inside the gray-shaded area. We assume that x is on the unit circle with the center at $\bar{x} \in \operatorname{Fix} T$.

It can be shown that T is ρ–SQNE if and only if there exists a QNE operator R such that

$$T = \left(1 - (1+\rho)^{-1}\right) \operatorname{Id} + (1+\rho)^{-1} R. \tag{4.5}$$

Note that every nonexpansive operator is also QNE, and every α–averaged operator is $\frac{1-\alpha}{\alpha}$–SQNE. In particular, all the operators in Definition 4.2 are QNE for which the following fact applies:

Fact 4.3 ([49])**.** *If $T : \mathcal{H} \mapsto \mathcal{H}$ is QNE, then* $\operatorname{Fix} T$ *is closed and convex.*

Operators arising from the iteration of a particular optimization method often enjoy at least some subset of the properties described in Definition 4.2. Figure 4.1 illustrates these properties and highlights that the distance of the iterates of a fixed-point iteration to any $\bar{x} \in \operatorname{Fix} T$ is nonincreasing if any of the properties above holds. In the case of a contractive operator its fixed-point iteration converges

and it satisfies inequality (4.3). Without additional assumptions the same is not true for a general SQNE operator.

In the sequel we will identify the essential additional property of an SQNE operator to ensure that a sequence generated by the related fixed-point iteration satisfies (4.3).

Remark 4.4. *If* (4.3) *holds and* $\operatorname{Fix} T$ *is a singleton, then* $\{x^k\}_{k\in\mathbb{N}}$ *converges linearly with convergence factor* β. *If* $\operatorname{Fix} T$ *is not a singleton, condition* (4.3) *itself does not imply that the sequence is linearly convergent. However, if* T *is QNE then the concept of Fejér monotonicity can be used to show that* (4.3) *implies linear convergence of* $\{x^k\}_{k\in\mathbb{N}}$ *[17, Thm. 5.12].*

4.2.1 Linear regularity

The linear convergence results we present will largely exploit the concept of *linear regularity* of an operator [23].

Definition 4.5 (Linear regularity)**.** *Let* $T : \mathcal{D} \mapsto \mathcal{H}$ *and suppose that Assumption 4.1 holds. We say that* T *is* linearly regular *with constant* $\kappa > 0$ *if, for all* $x \in \mathcal{D}$,

$$\operatorname{dist}_{\operatorname{Fix} T}(x) \leq \kappa \|x - Tx\|. \tag{4.6}$$

Linear regularity of an operator means that the distance between successive iterates of its fixed-point iteration, x^k and x^{k+1}, is lower-bounded by $\kappa^{-1} \operatorname{dist}_{\operatorname{Fix} T}(x^k)$. If the operator is in addition ρ–SQNE, then we can derive a lower bound on its linear regularity constant κ.

Proposition 4.6. *Let* $T : \mathcal{D} \to \mathcal{H}$ *be a* ρ*–SQNE operator. Then its linear regularity constant satisfies* $\kappa \geq (1+\rho)/2$.

Proof. Since T is ρ–SQNE, then according to (4.5),

$$R = (1+\rho)T - \rho \operatorname{Id}$$

is QNE, which implies that for every $x \in \mathcal{D} \setminus \operatorname{Fix} T$ and $y = \Pi_{\operatorname{Fix} T}(x)$

$$\begin{aligned} \|x - y\| \geq \|Rx - y\| &= \|(1+\rho)Tx - \rho x - y\| \\ &= \|(1+\rho)(Tx - x) + (x - y)\| \\ &\geq (1+\rho)\|Tx - x\| - \|x - y\|, \end{aligned}$$

where the last line follows from the triangle inequality. Rearranging the terms gives

$$2\|x - y\| \geq (1+\rho)\|Tx - x\| \geq (1+\rho)\kappa^{-1}\|x - y\|,$$

which implies that $\kappa \geq (1+\rho)/2$. □

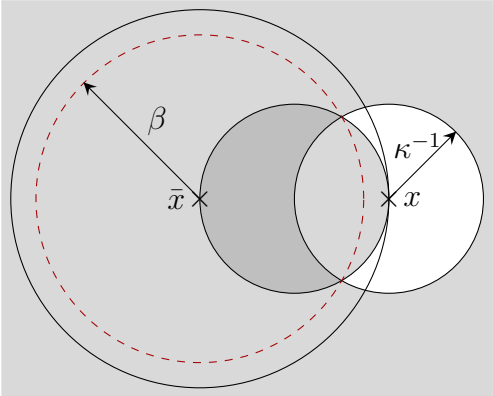


Figure 4.2: Linear convergence of an SQNE and linearly regular operator holds if the linear regularity constant κ is bounded from above. In that case Tx lies inside the circle with a radius $\beta < 1$. We assume that x is on the unit circle with the center at $\bar{x} \in \operatorname{Fix} T$.

Figure 4.2 illustrates that if the linear regularity constant of an SQNE operator T is bounded from above, *i.e.* $\kappa < \infty$, then Tx will lie inside the circle with a radius $\beta < 1$. This observation leads to our main result:

Theorem 4.7. *Let $T : \mathcal{D} \mapsto \mathcal{H}$ be a ρ–SQNE operator and suppose that Assumption 4.1 holds. Then the inequality*

$$\operatorname{dist}_{\operatorname{Fix} T}(Tx) \leq \beta \operatorname{dist}_{\operatorname{Fix} T}(x)$$

holds for some constant $\beta \in [0, 1)$ and for all $x \in \mathcal{D}$ if and only if T is linearly regular. If the linear regularity constant of T is κ, then

$$\beta = \sqrt{1 - \frac{\rho}{\kappa^2}} \in [0, 1). \tag{4.7}$$

Proof. See Appendix 4.A.1. □

Theorem 4.7 states that linear regularity, or a lack thereof, is the essential determinant of global linear convergence (in the sense of inequality (4.3)) for a sequence generated by the fixed-point iteration of an SQNE operator. We provide an example in Section 4.3 of an operator for which our bound in (4.7) is tight.

The bound in (4.7) was first derived in [9] for the class of α–averaged operators with $\rho = (1 - \alpha)/\alpha$. Recently, the same bound was reported independently in [117]. Previous results in [23] have established linear regularity as a sufficient condition for (4.3), albeit with the weaker convergence factor

$$\beta = \sqrt{\frac{\rho^{-1}\kappa^2}{1 + \rho^{-1}\kappa^2}}. \tag{4.8}$$

It is easy to show that our bound in Theorem 4.7 is strictly better than (4.8). A related result appears in [79], which establishes linear convergence of a particular proximal method for linear programming based on the weaker condition

$$\operatorname{dist}_{\operatorname{Fix} T}(Tx) \leq \tilde{\kappa}\|x - Tx\|, \tag{4.9}$$

for some ρ–SQNE operator T and some $\tilde{\kappa} > 0$, *i.e.* by lower-bounding $\|x - Tx\|$ via $\operatorname{dist}_{\operatorname{Fix} T}(Tx)$ rather than $\operatorname{dist}_{\operatorname{Fix} T}(x)$. In this case one can show that condition (4.3) still holds with the rate β as in (4.8), but with κ replaced by $\tilde{\kappa}$. By virtue of Theorem 4.7 any such operator must be linearly regular, so that (4.9) is also both necessary and sufficient for linear convergence of the fixed-point iteration of an SQNE operator.

4.2.2 Improving the convergence factor

It follows from (4.5) that it is possible to obtain an SQNE operator from QNE via a suitable transformation. It turns out that if T_1 is ρ_1–SQNE, we can construct a ρ_2–SQNE operator T_2 via an additional similar transformation, assuming that $\rho_1 > 0$ and $\rho_2 > 0$. We next show how to select ρ_2 in order to obtain the smallest convergence factor of the resulting fixed-point iteration:

Proposition 4.8. *Let $\rho_1 > 0$ and $\rho_2 > 0$. Suppose $T_1 : \mathcal{D} \mapsto \mathcal{H}$ is ρ_1–SQNE, with a linear regularity constant $\kappa_1 > 0$, and let*

$$T_2 \coloneqq \left(1 - \frac{1+\rho_1}{1+\rho_2}\right)\operatorname{Id} + \frac{1+\rho_1}{1+\rho_2} T_1. \tag{4.10}$$

Then $\operatorname{Fix} T_2 = \operatorname{Fix} T_1$ and the following hold:

(i) T_2 is ρ_2–SQNE with linear regularity constant $\kappa_2 = \kappa_1(1+\rho_2)/(1+\rho_1)$, and this estimate is tight.

(ii) The smallest convergence factor is attained for $\rho_2 = 1$.

Proof. It is easy to show that T_1 and T_2 have the same fixed-points. Indeed, if $T_1 x = x$ then according to (4.10) we have $T_2 x = x$, and *vice versa*.

(i): Since T_1 is ρ_1–SQNE, then according to (4.5),

$$R \coloneqq (1+\rho_1)T_1 - \rho_1 \operatorname{Id}$$

is QNE, which itself implies that

$$\begin{aligned} T_2 &\coloneqq (1 - \frac{1+\rho_1}{1+\rho_2})\operatorname{Id} + \frac{1+\rho_1}{1+\rho_2} T_1 \\ &= \left(1 - (1+\rho_2)^{-1}\right)\operatorname{Id} + (1+\rho_2)^{-1} R \end{aligned} \tag{4.11}$$

is ρ_2–SQNE. Since T_1 is κ_1-linearly regular, *i.e.*

$$\operatorname{dist}_{\operatorname{Fix} T_1}(x) \leq \kappa_1 \|x - T_1 x\|,$$

we have

$$\begin{aligned} \|x - T_2 x\| &= \frac{1+\rho_1}{1+\rho_2}\|x - T_1 x\| \\ &\geq \frac{1+\rho_1}{1+\rho_2}\kappa_1^{-1} \operatorname{dist}_{\operatorname{Fix} T_2}(x), \end{aligned}$$

which proves that T_2 is linearly regular with constant $\kappa_2 := \kappa_1(1+\rho_2)/(1+\rho_1)$.

Suppose now that κ_1 is a tight estimate of the linear regularity constant, *i.e.* there exists some $\bar{x}$ such that

$$\operatorname{dist}_{\operatorname{Fix} T_1}(\bar{x}) = \kappa_1 \|\bar{x} - T_1 \bar{x}\|.$$

Then we have

$$\|\bar{x} - T_2 \bar{x}\| = \kappa_2^{-1} \operatorname{dist}_{\operatorname{Fix} T_2}(\bar{x}),$$

which means that κ_2 is a tight estimate of the linear regularity constant of T_2.

(ii): The convergence factor of T_2 as a function of ρ_2 is

$$\beta_2(\rho_2) = \sqrt{1 - \frac{\rho_2}{\kappa_2^2}} = \sqrt{1 - \frac{\rho_2}{(1+\rho_2)^2}\left(\frac{1+\rho_1}{\kappa_1}\right)^2}.$$

This function is minimized when $\rho_2/(1+\rho_2)^2$ is maximized, which implies that the smallest convergence factor is attained for $\rho_2 = 1$. □

In other words, the worst-case convergence factor of an SQNE operator satisfying inequality (4.3) can be minimized by a simple transformation to make it 1–SQNE.

We will next show that some existing results establishing linear convergence for projection methods can be seen as special cases of Theorem 4.7.

4.3 Projection methods

Let A and B be two nonempty, closed, and convex subsets of $\mathcal{H}$ such that $A \cap B \neq \emptyset$. The *convex feasibility problem* is to find a point in their intersection, *i.e.*

$$\begin{aligned} &\text{find} && x \\ &\text{subject to} && x \in A \cap B. \end{aligned}$$

Note that this problem is a special case of problem (4.1) since it can be reformulated as

$$\text{minimize} \quad \mathcal{I}_A(x) + \mathcal{I}_B(x), \tag{4.12}$$

where $\mathcal{I}_A$ and $\mathcal{I}_B$ are the indicator functions of sets A and B, respectively.

Projection methods are a special class of operator splitting methods for solving feasibility problems and date to von Neumann's APM [132], given by the following iteration:

$$a^{k+1} \leftarrow \Pi_A(b^k) \tag{4.13a}$$
$$b^{k+1} \leftarrow \Pi_B(a^{k+1}). \tag{4.13b}$$

It is generally assumed that projection operators for both sets, Π_A and Π_B, can be evaluated efficiently. DRS is a related operator splitting method that has received increasing attention due to its generally observed good practical performance. In the case of the feasibility problem (4.12), DRS is defined as a fixed-point iteration of the Douglas-Rachford operator

$$T_{DR} \coloneqq \Pi_B(2\Pi_A - \mathrm{Id}) + \mathrm{Id} - \Pi_A, \tag{4.14}$$

or more explicitly

$$x^{k+1} \leftarrow \Pi_A(z^k) \tag{4.15a}$$
$$y^{k+1} \leftarrow \Pi_B(2x^{k+1} - z^k) \tag{4.15b}$$
$$z^{k+1} \leftarrow z^k + y^{k+1} - x^{k+1}. \tag{4.15c}$$

In the case when A and B are closed subspaces and $A + B$ is closed, both methods converge linearly and the convergence rate is characterized in terms of the Friedrichs angle between the subspaces [15, 64]. The Friedrichs angle is a generalization of the angle between subspaces in higher dimensions [15]:

Definition 4.9 (Friedrichs angle)**.** *Suppose that U and V are closed subspaces of $\mathcal{H}$. Cosine of the* Friedrichs angle *between U and V is*

$$\begin{aligned} c_F(U,V) \coloneqq \sup\Big\{ \langle u, v\rangle \mid u \in U \cap (U \cap V)^\perp,\ \|u\| \le 1, \\ v \in V \cap (U \cap V)^\perp,\ \|v\| \le 1 \Big\}. \end{aligned} \tag{4.16}$$

If $U + V$ is closed, which is true for any finite dimensional space $\mathcal{H}$, then $c_F(U,V) < 1$ [15, Fact 2.3(i)].

We will next show that the linear convergence in such cases can be recovered as a special case of Theorem 4.7 by establishing that the underlying fixed-point operators are both linearly regular and SQNE.

4.3.1 Alternating projection method

The authors in [133] show that APM for two convex polyhedra in $\mathbb{R}^n$, A and B, converges linearly and that the convergence rate is characterized via the Friedrichs angle between the faces of the two polyhedra. In the sequel we will briefly present this result in the case when $A \cap B \neq \emptyset$. Note that the original result is established without this assumption.

Theorem 4.10 ([133, Thm. 2 & Cor. 5])**.** *Let $A \subseteq \mathbb{R}^n$ and $B \subseteq \mathbb{R}^n$ be closed convex polyhedra such that $A \cap B \neq \emptyset$, and $b^0 \in B$. Then the sequences $\{a^k\}_{k\in\mathbb{N}}$ and $\{b^k\}_{k\in\mathbb{N}}$ generated by APM converge linearly towards some point in $A \cap B$, so that*

$$\operatorname{dist}_{A\cap B}(a^{k+1}) \leq \beta \operatorname{dist}_{A\cap B}(a^k)$$
$$\operatorname{dist}_{A\cap B}(b^{k+1}) \leq \beta \operatorname{dist}_{A\cap B}(b^k),$$

where the convergence factor is given by

$$\beta = \max_{\substack{A_x\in\mathcal{F}_A \\ B_y\in\mathcal{F}_B}} c_F^2(\operatorname{aff}_0(A_x), \operatorname{aff}_0(B_y)) < 1. \tag{4.17}$$

We will show that this result can be derived as a special case of Theorem 4.7. We can represent iteration (4.13) as the fixed-point iteration of the following operator defined on $\mathcal{D} = A \cup B$:

$$T : x \mapsto \begin{cases} \Pi_A(x) & x \in B \\ \Pi_B(x) & x \in A \end{cases} \tag{4.18}$$

performed twice. Since Π_A and Π_B are firmly nonexpansive (FNE) operators [17, Prop. 4.8], for every $x \in \mathcal{D}$ and $y \in A \cap B$ we have

$$\|Tx - y\|^2 \leq \|x - y\|^2 - \|x - Tx\|^2,$$

which means that T is 1–SQNE with $\operatorname{Fix} T = A \cap B$. Also, it can be shown that this operator is linearly regular with constant

$$\kappa = \left(1 - \max_{\substack{A_x\in\mathcal{F}_A \\ B_y\in\mathcal{F}_B}} c_F^2(\operatorname{aff}_0(A_x), \operatorname{aff}_0(B_y))\right)^{-1/2},$$

which holds for all $x \in \mathcal{D}$ [133, Prop. 4 & Cor. 5]. According to Theorem 4.7 these two properties ensure global linear convergence with convergence factor $\sqrt{\beta}$, where β is given in (4.17). Double iteration of the fixed-point operator given in (4.18), which is equivalent to the iteration (4.13), consequently has convergence factor β. Therefore, convergence rate given in Theorem 4.10 can be seen as a special case of Theorem 4.7. Observe that, although linear regularity does not hold for the whole space $\mathbb{R}^n$, by restricting $b^0 \in B$, linear convergence of the generated sequence is ensured since then the sequences $\{a^k\}_{k\in\mathbb{N}}$ and $\{b^k\}_{k\in\mathbb{N}}$ are in $\mathcal{D}$. Note that the linear convergence of APM for two subspaces is a special case of this result.

4.3.2 Douglas-Rachford splitting

The authors in [15] show that the linear convergence rate of DRS for two subspaces can also be characterized in terms of the Friedrichs angle.

Proposition 4.11 ([15, Cor. 4.5])**.** *Suppose that $A \subseteq \mathcal{H}$ and $B \subseteq \mathcal{H}$ are closed subspaces such that $A + B$ is closed, and let $z \in \mathcal{H}$. Then as $k \to \infty$,*

$$\begin{aligned} T_{DR}^k z &\to \Pi_{\operatorname{Fix} T_{DR}}(z) \\ \Pi_A(T_{DR}^k z) &\to \Pi_{A\cap B}(z) \\ \Pi_B(T_{DR}^k z) &\to \Pi_{A\cap B}(z). \end{aligned}$$

The convergence is linear with convergence factor $c_F(A, B) < 1$.

We will show that the convergence rate in Proposition 4.11 is again a special case of Theorem 4.7 by quantifying the linear regularity constant for the Douglas-Rachford operator in the case of a feasibility problem involving two subspaces.

Proposition 4.12. *Under the assumptions of Proposition 4.11 the linear regularity constant of $T_{DR} : \mathcal{H} \mapsto \mathcal{H}$ is*

$$\kappa_{DR} = \left(1 - c_F^2(A, B)\right)^{-1/2}.$$

Proof. See Appendix 4.A.2. □

Corollary 4.13. *Under the assumptions of Proposition 4.11, for all $z \in \mathcal{H}$*

$$\operatorname{dist}_{\operatorname{Fix} T_{DR}}(T_{DR} z) \leq c_F(A, B) \operatorname{dist}_{\operatorname{Fix} T_{DR}}(z).$$

Proof. The Douglas-Rachford operator in (4.14) is FNE [93], and thus 1–SQNE. The result follows directly from Theorem 4.7 by setting $\rho = 1$ and $\kappa = \kappa_{DR}$ from Proposition 4.12. □

We will now provide an example showing that the global linear convergence rate of DRS for two subspaces cannot be extended to the case of two convex polyhedra as is the case for APM.

Example 4.14. *Let $A = \varepsilon > 0$, $B = \mathbb{R}_+$, and $z^0 = t \gg \varepsilon$ the initial point of DRS. The first iteration of DRS is*

$$\begin{aligned} x^1 &\leftarrow \Pi_A(z^0) = \varepsilon \\ y^1 &\leftarrow \Pi_B(2x^1 - z^0) = 0 \\ z^1 &\leftarrow z^0 + y^1 - x^1 = t - \varepsilon. \end{aligned}$$

The sequence $\{z^k\}_{k\in\mathbb{N}}$ converges to $\varepsilon \in \operatorname{Fix} T_{DR}$. Note that the convergence factor in the first iteration is

$$\frac{\operatorname{dist}_{\operatorname{Fix} T_{DR}}(z^1)}{\operatorname{dist}_{\operatorname{Fix} T_{DR}}(z^0)} = \frac{|t - 2\varepsilon|}{|t - \varepsilon|}.$$

By taking t to be arbitrarily large, the convergence factor becomes arbitrarily close to 1 and, as long as the z-iterates are far enough from ε, the sequence converges with a constant step-size. Note however that the auxiliary sequence $\{x^k\}_{k\in\mathbb{N}}$ converges to the fixed-point in one iteration.

The absence of a global linear convergence rate arises from the absence of linear regularity of the Douglas-Rachford operator, which is, according to Theorem 4.7, essential for a global linear convergence rate. According to (4.6)*, the linear regularity constant is an upper bound to the following ratio:*

$$\kappa \geq \frac{\operatorname{dist}_{\operatorname{Fix} T_{DR}}(z^0)}{\|z^0 - z^1\|} = \frac{|t-\varepsilon|}{|\varepsilon|},$$

and by taking t be arbitrarily large with respect to ε, it is clear that such an upper bound is not finite.

4.3.3 Generalized DRS

Generalized DRS is a relaxed version of DRS, and is defined via the following fixed-point operator [14]:

$$T_{GDR} \coloneqq (1-2\alpha)\operatorname{Id} + 2\alpha\, T_{DR}, \tag{4.19}$$

where $\alpha \in (0,1)$. Since T_{DR} is $(1/2)$–averaged, it is easy to show that T_{GDR} is α–averaged and thus $\frac{1-\alpha}{\alpha}$–SQNE.

We next derive a linear convergence rate of the generalized DRS for two subspaces. We will use Theorem 4.7 to provide an upper bound on the convergence factor. We first derive a linear regularity constant of T_{GDR} in the following lemma:

Lemma 4.15. *Suppose that $A \subseteq \mathcal{H}$ and $B \subseteq \mathcal{H}$ are closed subspaces such that $A + B$ is closed. Then the linear regularity constant of $T_{GDR} : \mathcal{H} \mapsto \mathcal{H}$ is*

$$\kappa_{GDR} = (2\alpha)^{-1}\left(1 - c_F^2(A,B)\right)^{-1/2}.$$

Proof. The result follows directly from Proposition 4.8(i), Proposition 4.12, and the facts that T_{DR} is 1–SQNE and T_{GDR} is $\frac{1-\alpha}{\alpha}$–SQNE. □

Corollary 4.16. *Under the assumptions of Lemma 4.15,*

$$\sup_{z \notin \operatorname{Fix} T_{GDR}} \frac{\operatorname{dist}_{\operatorname{Fix} T_{GDR}}(T_{GDR} z)}{\operatorname{dist}_{\operatorname{Fix} T_{GDR}}(z)} \leq \beta_{GDR}, \tag{4.20}$$

where

$$\beta_{GDR} \coloneqq \sqrt{c_F^2(A,B) + (1-2\alpha)^2(1 - c_F^2(A,B))}. \tag{4.21}$$

Proof. The generalized Douglas-Rachford operator in (4.19) is $\frac{1-\alpha}{\alpha}$–SQNE. Choosing $\rho = \frac{1-\alpha}{\alpha}$ and $\kappa = \kappa_{GDR}$ from Lemma 4.15, Theorem 4.7 implies that, for all $z \in \mathcal{H}$,

$$\operatorname{dist}_{\operatorname{Fix} T_{DR}}(T_{DR} z) \leq \beta_{GDR} \operatorname{dist}_{\operatorname{Fix} T_{DR}}(z),$$

from which (4.20) follows directly. □

We now derive a lower bound on the linear convergence factor in the following lemma:

Lemma 4.17. *Under the assumptions of Lemma 4.15,*

$$\sup_{z \notin \operatorname{Fix} T_{GDR}} \frac{\operatorname{dist}_{\operatorname{Fix} T_{GDR}}(T_{GDR} z)}{\operatorname{dist}_{\operatorname{Fix} T_{GDR}}(z)} \geq \beta_{GDR},$$

where β_{GDR} is given in (4.21)*.*

Proof. See Appendix 4.A.3. □

We can now state the following theorem that substantially improves our result from Theorem 4.7. Note that an upper bound is said to be *tight* or *the least upper bound* if no smaller value is an upper bound.

Theorem 4.18. *The convergence rate bound in Theorem 4.7 for a ρ–SQNE and κ-linearly regular operator is tight for all admissible values of ρ and κ.*

Proof. Under the assumptions of Lemma 4.15, we have the following equality:

$$\sup_{z \notin \operatorname{Fix} T_{GDR}} \frac{\operatorname{dist}_{\operatorname{Fix} T_{GDR}}(T_{GDR} z)}{\operatorname{dist}_{\operatorname{Fix} T_{GDR}}(z)} = \beta_{GDR},$$

which follows directly from Corollary 4.16 and Lemma 4.17. Since the upper bound in (4.20) is obtained from (4.7), this means that the convergence rate bound in Theorem 4.7 is tight.

Since $A + B$ is closed, according to [15, Fact 2.3(i)], we have

$$0 \leq c_F(A, B) < 1,$$

and thus κ_{GDR} from Lemma 4.15 satisfies

$$\kappa_{GDR} \geq (2\alpha)^{-1} = (1 + \rho)/2,$$

where $\rho = (1 - \alpha)/\alpha$ is the constant of strong quasi-nonexpansiveness of operator T_{GDR}. Note that the range of values of κ_{GDR} covers all admissible values of linear regularity constant given in Proposition 4.6. By changing $\alpha \in (0, 1)$ and the angle between subspaces, we can produce ρ–SQNE and κ-linearly regular operator T_{GDR} with arbitrary $\rho > 0$ and $\kappa \geq (1 + \rho)/2$. □

Note that the results in Corollary 4.16 and Lemma 4.17 provide linear convergence rate of the generalized DRS for subspaces in a real Hilbert space, and thus generalize those in [15] where the linear convergence rate is established in a real Hilbert space, but for $\alpha = 1/2$ only, and those in [14] where the rate β_{GDR} is obtained for all $\alpha \in (0, 1)$, but in finite dimensional spaces only. Observe from (4.21) that the smallest value of β_{GDR} is obtained for $\alpha = 1/2$ which is consistent with the result in Proposition 4.8.

Remark 4.19. *Tightness of the bound in Theorem 4.7 is proved by providing an example with an averaged operator T_{GDR} for which an upper bound on the worst-case convergence factor coincides with a lower bound. This means that the bound on linear convergence rate for the more restrictive class of α–averaged operators given in [9, Thm. 1] is also tight.*

4.4 Linear programming

Most existing results on linear convergence for optimization problems arising in MPC assume a strongly convex quadratic objective function and linear system dynamics, resulting in a QP. However, there exist well-known applications of predictive control for which an LP arises, including problems based on ℓ_1–norm minimization [30, 144] and robust min–max predictive control [29, 112].

Linear convergence of a particular operator splitting method for linear programming was shown in [79], with an assumption that the problem is feasible with a bounded objective value. It is easy to show that this result is a special case of Theorem 4.7. The underlying fixed-point operator can be shown to be SQNE (see [79, Lem. 8]) and that inequality (4.9) holds (see [79, proof of Thm. 4]). As we note at the end of Section 4.2.1 this implies linear regularity of the operator. These two properties taken together imply linear convergence of a sequence generated by this method by virtue of Theorem 4.7.

In this section we propose a linearly convergent method for LPs that does not assume feasibility of the problem or boundedness of the objective value. We first introduce a reformulation of the original problem that is used in the proposed method.

4.4.1 Homogeneous self-dual embedding

Consider the following primal-dual pair of the convex conic optimization problem:

$$\begin{array}{lll} \text{minimize} & c^T x & \\ \text{subject to} & Ax + s = b & \\ & (x, s) \in \mathbb{R}^n \times \mathcal{K} & \end{array} \qquad \begin{array}{lll} \text{maximize} & -b^T y & \\ \text{subject to} & -A^T y + r = c & \\ & (r, y) \in \{0\}^n \times \mathcal{K}^*, & \end{array} \tag{4.22}$$

where $x \in \mathbb{R}^n$ and $s \in \mathbb{R}^m$ (with $n \leq m$) are the primal variables, $r \in \mathbb{R}^n$ and $y \in \mathbb{R}^m$ are the dual variables, $\mathcal{K} \subseteq \mathbb{R}^m$ is a nonempty, closed, and convex cone, and $\mathcal{K}^*$ is its dual cone. The problem data are $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, and $c \in \mathbb{R}^n$. In the case of linear programming $\mathcal{K} = \mathcal{K}^* = \mathbb{R}^m_+$.

The *homogeneous self-dual embedding (HSDE)* has been widely used with interior-point methods. The authors in [136] proposed solving such an embedding with an operator splitting method instead. The HSDE is a formulation that encodes the primal-dual pair of optimization problems into the convex feasibility problem

$$\begin{array}{ll} \text{find} & (u, v) \\ \text{subject to} & v = Qu \\ & (u, v) \in \mathcal{C} \times \mathcal{C}^*, \end{array} \tag{4.23}$$

where u, v, Q, $\mathcal{C}$ and $\mathcal{C}^*$ are defined as

$$u \coloneqq \begin{bmatrix} x \\ y \\ \tau \end{bmatrix}, \qquad v \coloneqq \begin{bmatrix} r \\ s \\ \kappa \end{bmatrix}, \qquad Q \coloneqq \begin{bmatrix} 0 & A^T & c \\ -A & 0 & b \\ -c^T & -b^T & 0 \end{bmatrix},$$
$$\mathcal{C} \coloneqq \mathbb{R}^n \times \mathcal{K}^* \times \mathbb{R}_+, \qquad \mathcal{C}^* \coloneqq \{0\}^n \times \mathcal{K} \times \mathbb{R}_+.$$

The objective of the above optimization problem is to find a point (u, v) that satisfies both the subspace and the conic constraint in (4.23). Any solution of the HSDE falls into one of the following three cases:

1. If $\tau > 0$ and $\kappa = 0$, then $(x/\tau, y/\tau, s/\tau)$ is a primal-dual solution of (4.22).
2. If $\tau = 0$ and $\kappa > 0$, then either primal or dual problem is infeasible. The case $b^T y < 0$ is a certificate for primal infeasibility, and the case $c^T x < 0$ is a certificate for dual infeasibility.
3. If $\tau = \kappa = 0$, then nothing can be concluded about the solution of (4.22).

The problem (4.23) is referred to as *homogeneous* because the feasible set is a convex cone, hence any nonnegative scaling of a solution is also in the solution set. The authors in [136] showed that by an appropriate selection of the initial point (u^0, v^0) any convergent method whose associated fixed-point operator is nonexpansive will not converge to zero if a nonzero solution $(u^\star, v^\star)$ exists. The appropriate initial point is any pair (u^0, v^0) satisfying $\langle (u^0, v^0), (u^\star, v^\star) \rangle > 0$. Since $(u^\star, v^\star)$ lies on the cone $\mathcal{C} \times \mathcal{C}^*$, it is sufficient for (u^0, v^0) to be contained in the (relative) interior of $\mathcal{C}^* \times \mathcal{C}$. In the case of an LP we have

$$(u^0, v^0) \in \operatorname{relint}(\mathcal{C}^* \times \mathcal{C}) = \{0\}^n \times \mathbb{R}^m_{++} \times \mathbb{R}_{++} \times \mathbb{R}^n \times \mathbb{R}^m_{++} \times \mathbb{R}_{++}.$$

4.4.2 APM for solving LPs

The authors in [136] proposed solving conic optimization problems in HSDE form using ADMM. In this section we propose solving an LP in the same form using APM. Since in the case of linear programming the cone $\mathcal{C} \times \mathcal{C}^*$ is polyhedral, we can apply Theorem 4.10 to show that the sequence of iterates generated by the method is linearly convergent.

Corollary 4.20. *The sequence generated by APM for solving an LP in the form* (4.23) *converges linearly.*

The proposed method is as follows:

$$(u_A^{k+1}, v_A^{k+1}) \leftarrow \Pi_{Qu=v}(u_B^k, v_B^k) \tag{4.24a}$$
$$(u_B^{k+1}, v_B^{k+1}) \leftarrow \Pi_{\mathcal{C}\times\mathcal{C}^*}(u_A^{k+1}, v_A^{k+1}). \tag{4.24b}$$

If the initial point (u_B^0, v_B^0) is selected as described in the previous subsection, then the sequence generated by the method will converge to a nonzero fixed-point. It should be noted that the projection (4.24a) requires solving a linear system involving a matrix $(I + Q^TQ)$. Step (4.24a) of the method can be computed as

$$u_A^{k+1} = (I + Q^TQ)^{-1}\left(u_B^k - Qv_B^k\right)$$
$$v_A^{k+1} = Qu_A^{k+1}.$$

Since $(I+Q^TQ)$ does not change throughout the iterations, it can be factored once and the factors are then used in cheaper back-solve operations in the subsequent iterations [45]. Projection onto the cone in (4.24b) is trivial and separable component-wise. Therefore, all the operations except matrix factorization in the first iteration are basic arithmetic operations.

4.5 Conclusions

In this chapter we provide necessary and sufficient conditions for global linear convergence of the fixed-point iteration of an SQNE operator. We also provide the tight bound on the achievable convergence rate. The proposed framework is based on properties of a fixed-point operator. We show that some published results on linear convergence in projection methods can be viewed as special cases of the proposed framework. Also, we propose a novel linearly convergent method for linear programming which not only finds solutions of solvable problems, but also detects infeasible and unbounded problems.

4.A Proofs

4.A.1 Proof of Theorem 4.7

We first provide a lemma that will be useful for proving the theorem.

Lemma 4.21. *Let $\mathcal{X} \subseteq \mathcal{H}$ and $\mathcal{Y} \subseteq \mathcal{H}$ be two nonempty closed sets. Then for any $z \in \mathcal{H}$, the following* generalized triangle inequality *holds:*

$$\operatorname{dist}(\mathcal{X}, \mathcal{Y}) \leq \operatorname{dist}_{\mathcal{X}}(z) + \operatorname{dist}_{\mathcal{Y}}(z). \tag{4.25}$$

Proof. Suppose that there exists some $z_1 \in \mathcal{H}$ for which (4.25) does not hold. Then we have

$$\begin{aligned} \operatorname{dist}(\mathcal{X}, \mathcal{Y}) &> \operatorname{dist}_{\mathcal{X}}(z_1) + \operatorname{dist}_{\mathcal{Y}}(z_1) \\ &= \|x_1 - z_1\| + \|y_1 - z_1\| \\ &\geq \|x_1 - y_1\| \end{aligned}$$

where x_1 and y_1 are projections of z_1 onto $\mathcal{X}$ and $\mathcal{Y}$, respectively, and the last inequality follows from the triangle inequality. The above inequality means that the distance between $x_1 \in \mathcal{X}$ and $y_1 \in \mathcal{Y}$ is strictly smaller than $\operatorname{dist}(\mathcal{X}, \mathcal{Y})$, which is a contradiction. This concludes the proof. □

To show that linear regularity is a necessary property of T for (4.3) to hold, we combine the generalized triangle inequality (4.25) with $\mathcal{X} = \operatorname{Fix} T$, $\mathcal{Y} = \{x\}$ and $z = Tx$, *i.e.*

$$\operatorname{dist}_{\operatorname{Fix} T}(x) \leq \operatorname{dist}_{\operatorname{Fix} T}(Tx) + \|x - Tx\|$$

with (4.3) to produce

$$(1 - \beta) \operatorname{dist}_{\operatorname{Fix} T}(x) \leq \operatorname{dist}_{\operatorname{Fix} T}(x) - \operatorname{dist}_{\operatorname{Fix} T}(Tx) \leq \|x - Tx\|,$$

which implies

$$\operatorname{dist}_{\operatorname{Fix} T}(x) \leq (1 - \beta)^{-1} \|x - Tx\|.$$

This proves that T is linearly regular.

The authors in [23, Lem. 3.8] show that linear regularity is a sufficient condition for (4.3) when T is α–averaged. However, we derive this result with a better convergence factor, and thus we repeat some arguments from [23] for the sake of completeness.

Suppose that T is κ–linearly regular, that is

$$\kappa^{-2} \operatorname{dist}^2_{\operatorname{Fix} T}(x) \leq \|x - Tx\|^2. \tag{4.26}$$

Note that since T is SQNE, according to Fact 4.3, $\operatorname{Fix} T$ is closed and convex, and thus the projection $\Pi_{\operatorname{Fix} T}$ is well defined. Taking $y = \Pi_{\operatorname{Fix} T}(x)$, we have

$$\begin{aligned} \operatorname{dist}^2_{\operatorname{Fix} T}(Tx) + \rho\|x - Tx\|^2 &\leq \|Tx - \Pi_{\operatorname{Fix} T}(x)\|^2 + \rho\|x - Tx\|^2 \\ &\leq \|x - \Pi_{\operatorname{Fix} T}(x)\|^2 \\ &= \operatorname{dist}^2_{\operatorname{Fix} T}(x), \end{aligned} \tag{4.27}$$

where the first inequality exploits properties of the distance between a point and a set, *i.e.* $\operatorname{dist}_{\operatorname{Fix} T}(Tx) \leq \|Tx - \Pi_{\operatorname{Fix} T}(x)\|$, and the second follows from (4.4b). Combining (4.26) and (4.27) implies

$$\kappa^{-2} \operatorname{dist}^2_{\operatorname{Fix} T}(x) \leq \rho^{-1} \left(\operatorname{dist}^2_{\operatorname{Fix} T}(x) - \operatorname{dist}^2_{\operatorname{Fix} T}(Tx) \right), \tag{4.28}$$

hence

$$\operatorname{dist}_{\operatorname{Fix} T}(Tx) \leq \sqrt{1 - \frac{\rho}{\kappa^2}} \operatorname{dist}_{\operatorname{Fix} T}(x). \tag{4.29}$$

Observe that the authors in [23] consider the following inequality which follows directly from (4.28):

$$\kappa^{-2} \operatorname{dist}^2_{\operatorname{Fix} T}(Tx) \leq \rho^{-1} \left(\operatorname{dist}^2_{\operatorname{Fix} T}(x) - \operatorname{dist}^2_{\operatorname{Fix} T}(Tx) \right),$$

from which they obtain a weaker convergence factor given by (4.8).

Note that (4.28) implies

$$\kappa^{-2} \operatorname{dist}^2_{\operatorname{Fix} T}(x) \leq \rho^{-1} \operatorname{dist}^2_{\operatorname{Fix} T}(x),$$

which means that $\kappa^2 \geq \rho$. Since $\rho > 0$ and $\kappa > 0$, we have

$$0 \leq \sqrt{1 - \frac{\rho}{\kappa^2}} < 1,$$

so (4.29) satisfies condition (4.3).

4.A.2 Proof of Proposition 4.12

The Douglas-Rachford operator T_{DR} is denoted by T in the sequel. Note from [15, Prop. 3.6(i)] that $z^\star \coloneqq \Pi_{\operatorname{Fix} T}(z^k)$ is in general not equal to $x^\star \coloneqq \Pi_{A \cap B}(z^k)$, and they coincide only when $z^k \in A + B$. Since T is an averaged operator, the sequence $\{z^k\}_{k \in \mathbb{N}}$ is Fejér monotone with respect to $\operatorname{Fix} T$, and according to [15, Prop. 3.6(i)] $\operatorname{Fix} T$ is a closed subspace. This means that $\Pi_{\operatorname{Fix} T}(z^k) = \Pi_{\operatorname{Fix} T}(z^0)$ for all $k \in \mathbb{N}$ [17, Prop. 5.9], *i.e.* $z^\star$ does not depend on the iteration k. Also, from $\Pi_{A \cap B} = \Pi_{A \cap B} \Pi_{\operatorname{Fix} T}$ [15, Prop. 3.6(v)], it follows that $x^\star = \Pi_{A \cap B}(z^\star)$, meaning that $x^\star$ does not depend on the iteration k either.

Let $z_P^k \coloneqq \Pi_{A+B}(z^k)$. We first provide a lemma that will be helpful in proving the main result.

Lemma 4.22. *Let A and B be two closed subspaces of $\mathcal{H}$ such that $A + B$ is closed. Then the following hold for the iterates of DRS:*

(i) $z^k - z^\star = z_P^k - x^\star$.

(ii) $x^{k+1} = \Pi_A(z_P^k)$ *and* $y^{k+1} = \Pi_B(2x^{k+1} - z_P^k)$.

(iii) $\|x^{k+1} - \Pi_B(x^{k+1})\|^2 \geq (1 - c_F^2(A, B))\ \|x^{k+1} - x^\star\|^2$.

(iv) $\|y^{k+1} - \Pi_B(x^{k+1})\|^2 \geq (1 - c_F^2(A, B))\ \|x^{k+1} - z_P^k\|^2$.

Proof. (i): By [15, Prop. 3.6(ii)] we have

$$z^\star = x^\star + \Pi_{A^\perp \cap B^\perp}(z^k),$$

which combined with the identity

$$(A^\perp \cap B^\perp)^\perp = A + B, \tag{4.30}$$

implies

$$z^k - \Pi_{A+B}(z^k) = \Pi_{(A+B)^\perp}(z^k) = z^\star - x^\star.$$

(ii): From the linearity of the projection onto a subspace, it follows

$$\begin{aligned} x^{k+1} &= \Pi_A(\overbrace{z^k}^{\in (A+B)^\perp}) \\ &= \Pi_A(\overbrace{z^k - \Pi_{A+B}(z^k)}) + \Pi_A(\Pi_{A+B}(z^k)) \\ &= \Pi_A(\Pi_{A+B}(z^k)), \end{aligned}$$

and similarly

$$\begin{aligned} y^{k+1} &= \Pi_B(2x^{k+1} - z^k) \\ &= \Pi_B(2x^{k+1} - \Pi_{A+B}(z^k)) - \Pi_B(\underbrace{z^k - \Pi_{A+B}(z^k)}_{\in (A+B)^\perp}) \\ &= \Pi_B(2x^{k+1} - \Pi_{A+B}(z^k)). \end{aligned}$$

(iii): We first show that the following hold:

$$\begin{aligned} x^{k+1} - x^\star &\in A \cap (A \cap B)^\perp \\ \Pi_B(x^{k+1}) - x^\star &\in B \cap (A \cap B)^\perp. \end{aligned}$$

From (4.15a) and the definition of $x^\star$ it is clear that $x^{k+1} - x^\star \in A$. Since the projection onto $A \cap B$ is a linear operation, we have

$$\begin{aligned} \Pi_{A\cap B}(x^{k+1}) &= \Pi_{A\cap B}\left(\Pi_A(z^k)\right) \\ &= \Pi_{A\cap B}\underbrace{\left(\Pi_A(z^k) - z^k\right)}_{\in A^\perp} + \Pi_{A\cap B}(z^k) \\ &= x^\star, \end{aligned}$$

which proves that $x^{k+1} - x^\star = x^{k+1} - \Pi_{A\cap B}(x^{k+1}) \in (A \cap B)^\perp$. Similarly, from the definition of $x^\star$ it is clear that $\Pi_B(x^{k+1}) - x^\star \in B$. We also have

$$\begin{aligned}\Pi_{A\cap B}\left(\Pi_B(x^{k+1})\right) &= \Pi_{A\cap B}\underbrace{\left(\Pi_B(x^{k+1}) - x^{k+1}\right)}_{\in B^\perp} + \Pi_{A\cap B}(x^{k+1}) \\ &= x^\star,\end{aligned}$$

which proves that

$$\Pi_B(x^{k+1}) - x^\star = \Pi_B(x^{k+1}) - \Pi_{A\cap B}\left(\Pi_B(x^{k+1})\right) \in (A \cap B)^\perp.$$

From (4.16) it follows

$$\begin{aligned}c_F(A,B) &\geq \frac{\left\langle \overbrace{x^{k+1} - x^\star}^{\in A\cap(A\cap B)^\perp}, \overbrace{\Pi_B(x^{k+1}) - x^\star}^{\in B\cap(A\cap B)^\perp} \right\rangle}{\|x^{k+1} - x^\star\| \, \|\Pi_B(x^{k+1}) - x^\star\|} \\ &= \frac{\|\Pi_B(x^{k+1}) - x^\star\|^2}{\|x^{k+1} - x^\star\| \, \|\Pi_B(x^{k+1}) - x^\star\|} \\ &= \frac{\|\Pi_B(x^{k+1}) - x^\star\|}{\|x^{k+1} - x^\star\|},\end{aligned}$$

so that

$$\begin{aligned}1 - c_F^2(A,B) &\leq \frac{\|x^{k+1} - x^\star\|^2 - \|\overbrace{\Pi_B(x^{k+1}) - x^\star}^{\in B}\|^2}{\|x^{k+1} - x^\star\|^2} \\ &= \frac{\|\overbrace{x^{k+1} - \Pi_B(x^{k+1})}^{\in B^\perp}\|^2}{\|x^{k+1} - x^\star\|^2}.\end{aligned}$$

(iv): We first show that the following hold:

$$\begin{aligned}x^{k+1} - z_P^k &\in A^\perp \cap (A^\perp \cap B^\perp)^\perp \\ \Pi_{B^\perp}(x^{k+1} - z_P^k) &\in B^\perp \cap (A^\perp \cap B^\perp)^\perp.\end{aligned}$$

From part (ii) of the lemma it is clear that $x^{k+1} - z_P^k = \Pi_A(z_P^k) - z_P^k \in A^\perp$. From the definition of x^{k+1} and z_P^k we also have $x^{k+1} - z_P^k \in A + B$, which combined with (4.30) proves that $x^{k+1} - z_P^k \in (A^\perp \cap B^\perp)^\perp$. Similarly, it is clear that $\Pi_{B^\perp}(x^{k+1} - z_P^k) \in B^\perp$. Note that $\Pi_{B^\perp}(x^{k+1} - z_P^k)$ can be split as follows

$$\Pi_{B^\perp}(x^{k+1} - z_P^k) = (x^{k+1} - z_P^k) - \Pi_B(x^{k+1} - z_P^k).$$

Since the first summand of the right hand side is in $A+B$ and the second one is in B, their difference is in $A+B$, which proves that $\Pi_{B^\perp}(x^{k+1}-z_P^k) \in (A^\perp \cap B^\perp)^\perp$. From (4.16) it follows

$$\begin{aligned} c_F(A^\perp, B^\perp) &\geq \frac{\left\langle \overbrace{x^{k+1}-z_P^k}^{\in A^\perp \cap (A+B)}, \overbrace{\Pi_{B^\perp}(x^{k+1}-z_P^k)}^{\in B^\perp \cap (A+B)} \right\rangle}{\|x^{k+1}-z_P^k\| \, \|\Pi_{B^\perp}(x^{k+1}-z_P^k)\|} \\ &= \frac{\|\Pi_{B^\perp}(x^{k+1}-z_P^k)\|^2}{\|x^{k+1}-z_P^k\| \, \|\Pi_{B^\perp}(x^{k+1}-z_P^k)\|} \\ &= \frac{\|\Pi_{B^\perp}(x^{k+1}-z_P^k)\|}{\|x^{k+1}-z_P^k\|}, \end{aligned}$$

so that

$$\begin{aligned} 1-c_F^2(A^\perp, B^\perp) &\leq \frac{\|x^{k+1}-z_P^k\|^2 - \|\overbrace{\Pi_{B^\perp}(x^{k+1}-z_P^k)}^{\in B^\perp}\|^2}{\|x^{k+1}-z_P^k\|^2} \\ &= \frac{\|\overbrace{\Pi_B(x^{k+1}-z_P^k)}^{\in B}\|^2}{\|x^{k+1}-z_P^k\|^2}. \end{aligned}$$

By observing that the following holds:

$$\begin{aligned} \Pi_B(x^{k+1}-z_P^k) &= \Pi_B(2x^{k+1}-z_P^k) - \Pi_B(x^{k+1}) \\ &= y^{k+1} - \Pi_B(x^{k+1}), \end{aligned}$$

and that $c_F(A^\perp, B^\perp) = c_F(A, B)$ [15, Fact 2.3], we have

$$1-c_F^2(A,B) \leq \frac{\|y^{k+1}-\Pi_B(x^{k+1})\|^2}{\|x^{k+1}-z_P^k\|^2}. \qquad \square$$

We are now ready to derive the linear regularity constant. We have

$$\begin{aligned} \|z^k - z^\star\|^2 &= \|z_P^k - x^\star\|^2 \\ &= \|\underbrace{z_P^k - x^{k+1}}_{\in A^\perp}\|^2 + \|\underbrace{x^{k+1}-x^\star}_{\in A}\|^2 \\ &\leq \frac{1}{1-c_F^2(A,B)} \|\underbrace{y^{k+1}-\Pi_B(x^{k+1})}_{\in B}\|^2 \\ &\quad + \frac{1}{1-c_F^2(A,B)} \|\underbrace{x^{k+1}-\Pi_B(x^{k+1})}_{\in B^\perp}\|^2 \end{aligned}$$

$$= \frac{1}{1 - c_F^2(A,B)} \|y^{k+1} - x^{k+1}\|^2$$
$$= \frac{1}{1 - c_F^2(A,B)} \|z^k - z^{k+1}\|^2.$$

The first equality follows from Lemma 4.22(i), the first inequality from Lemma 4.22 (iii)–(iv), and the last equality follows from (4.15c). This concludes the proof.

4.A.3 Proof of Lemma 4.17

We first provide the following supporting results. We denote by $\operatorname{Fix} T \coloneqq \operatorname{Fix} T_{DR} = \operatorname{Fix} T_{GDR}$.

Lemma 4.23. *Let A and B be closed subspaces of $\mathcal{H}$. Then*

$$c_F(A,B) = \|(\Pi_B - \Pi_{A\cap B})(\Pi_A - \Pi_{A\cap B})\|.$$

Proof. From [15, Fact 2.3] we have the following identity:

$$c_F(A,B) = \|\Pi_B\Pi_A - \Pi_{A\cap B}\|.$$

It is now sufficient to show that the right-hand terms of the above identities are equal. We have

$$\begin{aligned}(\Pi_B - \Pi_{A\cap B})(\Pi_A - \Pi_{A\cap B}) &= \Pi_B\Pi_A - \Pi_B\Pi_{A\cap B} - \Pi_{A\cap B}\Pi_A + \Pi_{A\cap B}^2 \\ &= \Pi_B\Pi_A - \Pi_{A\cap B} - \Pi_{A\cap B} + \Pi_{A\cap B} \\ &= \Pi_B\Pi_A - \Pi_{A\cap B},\end{aligned}$$

where the second equality follows from [15, Eq. (7)]. □

Lemma 4.24. *Let A and B be closed subspaces of $\mathcal{H}$, and T_{DR} the operator given in* (4.14)*. Then*

$$c_F(A,B) \leq \sup_{u\in A\setminus \operatorname{Fix} T} \frac{\operatorname{dist}_{A\cap B}(\Pi_B(u))}{\operatorname{dist}_{A\cap B}(u)}. \tag{4.31}$$

Proof. The characterization of $c_F(A,B)$ in Lemma 4.23 implies

$$c_F(A,B) = \sup_{x\neq 0} \frac{\|(\Pi_B - \Pi_{A\cap B})(\Pi_A - \Pi_{A\cap B})x\|}{\|x\|}. \tag{4.32}$$

If $c_F(A,B) = 0$, then (4.31) holds trivially. We thus assume that $c_F(A,B) > 0$, which implies that for some $x \neq 0$ the numerator in (4.32) is nonzero. We can

therefore disregard $x \in (A \cap B) + A^\perp$ from the above supremum since in that case $(\Pi_A - \Pi_{A\cap B})\, x = 0$ and the numerator in (4.32) is zero. We now have

$$\begin{aligned} c_F(A,B) &= \sup_{x \notin (A\cap B)+A^\perp} \frac{\|(\Pi_B - \Pi_{A\cap B})\,(\Pi_A - \Pi_{A\cap B})\, x\|}{\|x\|} \\ &\leq \sup_{x \notin (A\cap B)+A^\perp} \frac{\|(\Pi_B - \Pi_{A\cap B})(\Pi_A - \Pi_{A\cap B})x\|}{\|(\Pi_A - \Pi_{A\cap B})x\|} \\ &\leq \sup_{u \in A\setminus\{0\}} \frac{\|(\Pi_B - \Pi_{A\cap B})u\|}{\|u\|} \end{aligned}$$

where the second line follows from the nonexpansiveness of projection Π_A and the fact that $0 \in A \cap B$, and the third from the fact that $(\Pi_A - \Pi_{A\cap B})x \in A$. Similarly, we can disregard $u \in \operatorname{Fix} T$ from the above supremum since then $(\Pi_B - \Pi_{A\cap B})\, x = 0$ [15, Prop. 3.6(v)]. We now obtain

$$c_F(A,B) \leq \sup_{u \in A\setminus \operatorname{Fix} T} \frac{\operatorname{dist}_{A\cap B}(\Pi_B(u))}{\operatorname{dist}_{A\cap B}(u)},$$

where we used the fact that $0 \in A \cap B$ and $\Pi_{A\cap B} = \Pi_{A\cap B}\Pi_B$ [63, Thm. 5.14]. □

We are now in a position to prove Lemma 4.17. From the definition of the generalized Douglas-Rachford operator (4.19) we obtain

$$T_{GDR} z^k - z^{k+1} = (1-2\alpha)(z^k - z^{k+1}),$$

where $z^{k+1} = T_{DR} z^k$. For $z^k \in A \setminus \operatorname{Fix} T$ DRS produces the following iterates in the subsequent iteration:

$$\begin{aligned} x^{k+1} &= z^k \\ y^{k+1} &= \Pi_B(z^k) \\ z^{k+1} &= \Pi_B(z^k). \end{aligned}$$

Note also that the following inclusions hold:

$$\begin{aligned} z^k - z^{k+1} &= z^k - \Pi_B(z^k) \in B^\perp \\ z^{k+1} - z^\star &= \Pi_B(z^k) - \Pi_{A\cap B}(z^k) \in B, \end{aligned}$$

where $z^\star := \Pi_{\operatorname{Fix} T}(z^k) = \Pi_{A\cap B}(z^k) = \Pi_{\operatorname{Fix} T}(T_{GDR} z^k)$ [15, Prop. 3.6], [20, Lem. 3.12]. We now have

$$\begin{aligned} T_{GDR} z^k - z^\star &= (T_{GDR} z^k - z^{k+1}) + (z^{k+1} - z^\star) \\ &= (1-2\alpha) \underbrace{(z^k - z^{k+1})}_{\in B^\perp} + \underbrace{(z^{k+1} - z^\star)}_{\in B}, \end{aligned}$$

and therefore

$$\|T_{GDR}z^k - z^\star\|^2 = (1-2\alpha)^2\|z^k - z^{k+1}\|^2 + \|z^{k+1} - z^\star\|^2.$$

Dividing the above equality by $\|z^k - z^\star\|^2 > 0$, and taking the supremum over $z^k \in A \setminus \operatorname{Fix} T$, we obtain

$$\begin{aligned}\sup_{z^k \in A\setminus \operatorname{Fix} T} \frac{\|T_{GDR}z^k - z^\star\|^2}{\|z^k - z^\star\|^2} &= \sup_{z^k \in A\setminus \operatorname{Fix} T}\left((1-2\alpha)^2\frac{\|z^k - z^{k+1}\|^2}{\|z^k - z^\star\|^2} + \frac{\|\Pi_B(z^k) - z^\star\|^2}{\|z^k - z^\star\|^2}\right)\\ &\geq (1-2\alpha)^2(1 - c_F^2(A,B)) + c_F^2(A,B)\\ &= \beta_{GDR}^2,\end{aligned}$$

where we used $z^{k+1} = \Pi_B(z^k)$ in the second line, and the third line follows from Proposition 4.12 and Lemma 4.24. Finally, we have

$$\sup_{z^k \notin \operatorname{Fix} T} \frac{\|T_{GDR}z^k - z^\star\|^2}{\|z^k - z^\star\|^2} \geq \sup_{z^k \in A\setminus \operatorname{Fix} T} \frac{\|T_{GDR}z^k - z^\star\|^2}{\|z^k - z^\star\|^2} \geq \beta_{GDR}^2.$$

This concludes the proof.

5

Regularized Jacobi Algorithm for Convex Optimization

Contents

5.1 Introduction

We consider large-scale optimization problems in which a collection of individual actors (or *agents*) cooperate to minimize some common objective function while incorporating local constraints or additional local utility functions. We consider a decentralized optimization method based on block coordinate descent, an iterative coordinating procedure which has attracted significant attention in large-scale optimization [27, 34, 173].

Solving large-scale optimization problems via an iterative procedure that coordinates among blocks of variables enables the solution of very large problem instances by parallelizing computation across agents. This enables one to overcome computational challenges that would be prohibitive otherwise, without requiring

agents to reveal their local utility functions and constraints to other agents. Due to its pricing mechanism implications, decentralized optimization is also a natural choice for many applications, including demand side management in smart grids, charging coordination for plug-in electric vehicles, coordination of multiple agents in robotic systems etc. [61, 98, 138].

Based on the algorithms outlined in [34], two classes of iterative methods have been employed recently for solving such optimization problems in a decentralized way. The first class includes *block coordinate gradient descent (BCGD)* methods and it requires each agent to perform, at every iteration, a local (proximal) gradient descent step [27, 138]. Under certain regularity assumptions (differentiability of the objective function and Lipschitz continuity of its gradient), and for an appropriately chosen gradient step-size, this method converges to a minimizer of the centralized problem. This class of algorithms includes both sequential [104] and parallel [128, 146] implementations.

The second covers *block coordinate minimization (BCM)* methods, does not assume differentiability of the objective and is based on minimizing the common objective function in each block by fixing variables associated with other agents to their previously computed values. Although BCM methods have a larger per iteration cost than the BCGD methods in the case when there are no local utility functions (constraints) in the problem, or when their proximal operators (projections) have closed-form solutions, in the general case both approaches require solutions of ancillary optimization problems. On the other hand, iterations of BCM methods are numerically more stable than gradient iterations, as observed in [32].

If the block-wise minimizations are done in a cyclic fashion across agents, then the algorithm is known as the Gauss-Seidel algorithm [5, 104, 173]. An alternative implementation, known as the Jacobi algorithm, involves performing the block-wise minimizations in parallel. However, convergence of the Jacobi algorithm is not guaranteed in general, even in the case when the objective function is smooth and convex, unless certain contractiveness properties are satisfied [34, Prop. 2.6 in §3.2 & Prop. 3.10 in §3.3].

The authors in [51] have proposed a regularized Jacobi algorithm wherein, at each iteration, each agent minimizes the weighted sum of the common objective function and a quadratic regularization term penalizing the distance to the previous iterate of the algorithm. A similar regularization has been used in Gauss-Seidel methods [5, 104] which are however not parallelizable. Under certain regularity assumptions, and for an appropriately selected regularization weight, the algorithm converges *in objective value* to the optimal value of the centralized problem [51]. Recently, the authors in [62] have quantified the regularization weighting required to ensure convergence in objective value as a function of the number of agents and other problem parameters. However, convergence of the algorithm *in its iterates* to an optimizer of the centralized problem counterpart was not established, apart from the particular case when the objective function is quadratic.

In this chapter we revisit the algorithm proposed in [51] and establish its convergence under milder conditions. By adopting an analysis based on a power growth condition, which is in turn sufficient for the satisfaction of the so-called Kurdyka-Łojasiewicz property [5, 39], we show that the algorithm's iterates converge under much milder assumptions on the objective function than those used in [34] and [62]. A similar approach was used in [5, 173] to establish convergence of iterates generated by Gauss-Seidel type methods. We also show that the algorithm achieves a linear convergence rate without imposing restrictive strong convexity assumptions on the objective function, in contrast to typical methods in the literature. Our analysis is based on the quadratic growth condition, which is closely related to the so-called error bound property [74, 175] that is used in [128] to establish linear convergence of parallel BCGD methods *in objective value*.

5.2 Problem description and main result

5.2.1 Regularized Jacobi algorithm

We consider the following optimization problem:

$$\min_{\{x_i\}_{i=1}^m} \Big\{ f(x_1, \ldots, x_m) + \sum_{i=1}^m g_i(x_i) \Big\}, \tag{5.1}$$

where $x \coloneqq (x_1, \ldots, x_m) \in \mathbb{R}^n$, $x_i \in \mathbb{R}^{n_i}$ and $n = \sum_{i=1}^m n_i$. To simplify subsequent derivations we define $f(x) \coloneqq f(x_1, \ldots, x_m)$, $g(x) \coloneqq \sum_{i=1}^m g_i(x_i)$ with $\operatorname{dom} g = \operatorname{dom} g_1 \times \ldots \times \operatorname{dom} g_m$, and the combined objective function in (5.1) as

$$h(x) \coloneqq f(x) + g(x).$$

Problems in the form (5.1) can be viewed as multi-agent optimization problems wherein each agent has its own local decision vector x_i and agents cooperate to determine a minimizer of h, which couples the local decision vectors of all agents through the common objective function f. Since the number of agents can be large, solving the problem in a centralized fashion may be computationally intensive. Moreover, even if this were possible from a computational point of view, agents may not be willing to share their local objectives g_i with other agents, since this encodes potentially private information about their local utility functions or constraint sets.

For each $i = 1, \ldots, m$, we let $f_i(\,\cdot\,; x_{-i}) : \mathbb{R}^{n_i} \mapsto \mathbb{R}$ be a function of the decision vector of the i-th block of variables, with the remaining variables $x_{-i} \in \mathbb{R}^{n-n_i}$ treated as a fixed set of parameters, *i.e.*

$$f_i(z_i; x_{-i}) \coloneqq f(x_1, \cdots, x_{i-1}, z_i, x_{i+1}, \cdots, x_m).$$

We wish to solve (5.1) in a decentralized fashion using Algorithm 5.1. At the $(k+1)^{\text{th}}$ iteration of Algorithm 5.1 agent i solves a local optimization problem

Algorithm 5.1 Regularized Jacobi algorithm.

given initial values $x_i^0 \in \operatorname{dom} g_i$ for all $i = 1, \dots, m$, and parameter $c > 0$
repeat
  for $i = 1, \dots, m$ **do**
    $x_i^{k+1} \leftarrow \operatorname{argmin}_{z_i} \left\{ f_i(z_i; x_{-i}^k) + g_i(z_i) + c\|z_i - x_i^k\|^2 \right\}$
  end for
  $k \leftarrow k + 1$
until convergence

accounting for its local function g_i and the function f_i with the parameter vector set to the decisions x_{-i}^k of the other agents from the previous iteration. Moreover in the local cost function an additional term penalizes the squared distance between the optimization variables and their values at the previous iteration x_i^k. The relative importance of the original cost function and the penalty term is regulated by the weight $c > 0$, which should be selected large enough to guarantee convergence [51, 62]. We show in Appendix 5.A.1 that the fixed-points of Algorithm 5.1 coincide with optimal solutions of problem (5.1).

A problem structure equivalent to (5.1) was considered in [62], with the difference that a collection of convex constraints $x_i \in \mathcal{C}_i$ were introduced instead of the functions g_i. We can rewrite this problem in the form of (5.1) by selecting g_i to be the indicator function of a given convex set. On the other hand, problem (5.1) can be written in epigraph form, and thus reformulated in the framework of [62]. The reason that we use the problem structure of (5.1) is twofold. First, some widely used problems such as ℓ_1–regularized least squares are typically posed in the form (5.1). Second, the absence of constraints will ease the convergence analysis of Section 5.3 since many results in the relevant literature use the same problem structure.

5.2.2 Statement of the main result

Before stating the main result we provide some necessary definitions and assumptions. Let $h^\star$ denote the minimum value of problem (5.1). We then have the following definition:

Definition 5.1 (Power-type growth condition)**.** *A function $h : \mathbb{R}^n \mapsto \tilde{\mathbb{R}}$ satisfies a* power-type growth condition *on $[h^\star < h < h^\star + r]$ if there exist $r > 0$, $\gamma > 0$ and $p \geq 1$ such that, for all $x \in [h^\star < h < h^\star + r]$,*

$$h(x) - h^\star \geq \gamma \operatorname{dist}_{\operatorname{argmin} h}(x)^p. \tag{5.2}$$

It should be noted that (5.2) is a very mild condition, since it requires only that the function h is not excessively *flat* in the neighborhood of the set $\operatorname{argmin} h$.

For instance, all polynomial, real-analytic and semi-algebraic functions satisfy this condition [4, 39].

We impose the following standing assumptions on problem (5.1):

Assumption 5.2.

(i) The function f is convex and L–Lipschitz smooth.

(ii) The functions g_i are all proper, convex, and lower semicontinuous.

(iii) The function h is coercive, i.e.

$$\lim_{\|x\|\to\infty} h(x) = +\infty.$$

(iv) The function h exhibits the power-type growth condition of Definition 5.1.

Notice that we do not require differentiability of the functions g_i. Coerciveness of h implies the existence of some $\zeta \in \mathbb{R}$ for which the sublevel set $[h \leq \zeta]$ is nonempty and bounded, which is sufficient to prove existence of a minimizer of h [17, Prop. 11.12 & Thm. 11.9].

We are now in a position to state the main result of the chapter.

Theorem 5.3. *Under Assumption 5.2, if*

$$c > \frac{(m-1)^{3/2}}{2m-1} L, \tag{5.3}$$

then the iterates $\{x^k\}_{k\in\mathbb{N}}$ generated by Algorithm 5.1 converge to a minimizer of problem (5.1)*,* i.e. $\lim_{k\to\infty} x^k = x^\star$*, where* $x^\star \in \operatorname{argmin} h$.

The proof of Theorem 5.3 involves several intermediate statements and is provided in the next section.

5.3 Proof of the main result

Many results on convergence of optimization algorithms establish only convergence in function value [34, 45, 62], without guaranteeing convergence of the iterates $\{x^k\}_{k\in\mathbb{N}}$ as well. Convergence of iterates is straightforward to show when h is strongly convex, or when the operator underlying the iteration update is averaged [17, Prop. 5.15]. The latter condition was used in [62] to establish convergence of the sequence $\{x^k\}_{k\in\mathbb{N}}$ in the special case that f is a convex quadratic function.

In the single-agent case, *i.e.* when $m = 1$, Algorithm 5.1 reduces to the proximal minimization algorithm whose associated fixed-point operator is averaged for any proper, convex, and lower semicontinuous function h. However, in the multi-agent setting the resulting fixed-point operator is not necessarily averaged, which implies that the analysis based on [17, Prop. 5.15] cannot be employed to establish convergence of the sequence $\{x^k\}_{k\in\mathbb{N}}$. To achieve this and prove Theorem 5.3 we exploit the following result, which follows directly from Theorem 14 in [39]:

Theorem 5.4 ([39, Thm. 14])**.** *Consider Assumption 5.2, with* $\operatorname{argmin} h \neq \emptyset$*. Assume that the initial iterate* $x^0 \in \operatorname{dom} g$ *of Algorithm 5.1 satisfies* $h(x^0) < h^\star + r$*, where* r *is as in Definition 5.1. Finally, assume that subsequent iterates generated by Algorithm 5.1 possess the following properties:*

1. *Sufficient decrease condition:*

$$h(x^{k+1}) \leq h(x^k) - a \left\|x^{k+1} - x^k\right\|^2, \tag{5.4}$$

   where $a > 0$*.*

2. *Relative error condition: There exists* $w^{k+1} \in \partial h(x^{k+1})$ *such that*

$$\left\|w^{k+1}\right\| \leq b \left\|x^{k+1} - x^k\right\|, \tag{5.5}$$

   where $b > 0$*.*

Then the sequence $\{x^k\}_{k\in\mathbb{N}}$ *converges to some* $x^\star \in \operatorname{argmin} h$*,* i.e. $\lim_{k\to\infty} x^k = x^\star$*, and for all* $k \geq 1$

$$\left\|x^k - x^\star\right\| \leq b a^{-1} \frac{p}{\gamma^{1/p}} \left(h(x^k) - h^\star\right)^{1/p} + \sqrt{a^{-1}\left(h(x^{k-1}) - h^\star\right)}. \tag{5.6}$$

It should be noted that Theorem 5.4 constitutes a relaxed version of [39, Thm. 14]. Specifically, we could replace the last part of Assumption 5.2 with the KL property and the conclusion of Theorem 5.4 would remain valid.

Notice that, under the assumptions of Theorem 5.4, $\{x^k\}_{k\in\mathbb{N}}$ converges to some $x^\star \in \operatorname{argmin} h$ even if $h(x^0) \geq h^\star + r$. As a consequence of the sufficient decrease condition (5.4) and the fact that the set of fixed-points of Algorithm 5.1 coincides with $\operatorname{argmin} h$ (see Proposition 5.11 in Appendix 5.A.1), $\{h(x^k)\}_{k\in\mathbb{N}}$ converges to $h^\star$ and thus there exists some $k_0 \in \mathbb{N}$ such that $h(x^{k_0}) < h^\star + r$, and hence Theorem 5.4 remains valid if x^k is replaced by x^{k+k_0}.

To prove Theorem 5.3 it suffices to show that, given Assumption 5.2, the iterates generated by Algorithm 5.1 satisfy the *sufficient decrease condition* and the *relative error condition*. To show this we first provide an auxiliary lemma.

Lemma 5.5. *Under Assumption 5.2, for all* $(x, y, z) \in \{\operatorname{dom} g\}^3$*,*

$$\left\| \sum_{i=1}^m \nabla f_i(z_i; x_{-i}) - \sum_{i=1}^m \nabla f_i(z_i; y_{-i}) \right\| \leq \sqrt{m-1} L \left\|x - y\right\|.$$

Proof. The statement follows from [62, Lem. 1]. However, by noticing that $\sum_{i=1}^m \|x_{-i} - y_{-i}\|^2 = (m-1)\left\|x - y\right\|^2$ instead of $m\left\|x - y\right\|^2$, we obtain an improvement in the bound of [62, Lem. 1]. □

We can now show that the sufficient decrease condition is satisfied.

Proposition 5.6 (Sufficient decrease condition)**.** *Under Assumption 5.2, if c is chosen according to* (5.3)*, then Algorithm 5.1 converges to the minimum of problem* (5.1) *in value,* i.e. $h(x^k) \to h^\star$*, and for all $k \in \mathbb{N}$ the sufficient decrease condition* (5.4) *is satisfied with*

$$a = m^{-1}\left(c - (m-1)(\sqrt{m-1}L - 2c)\right) > 0. \tag{5.7}$$

Proof. The result follows from [62, Thm. 2], with the Lipschitz constant established in Lemma 5.5. □

Note that the proofs of Lemma 5.5 and Proposition 5.6 do not require last part of Assumption 5.2, which relates to the power-type growth condition of h.

If c is chosen according to Theorem 5.3, then (5.4) implies that $x^{k+1} - x^k \to 0$. To show this, suppose that $x^0 \in \operatorname{dom} h$ and thus $h(x^0)$ is finite. Iterating the inequality (5.4) gives

$$a \sum_{k=0}^{\infty} \|x^{k+1} - x^k\|^2 \leq h(x^0) - h^\star < +\infty,$$

which means that $\|x^{k+1} - x^k\|$ converges to zero. Note however that this does not necessarily imply convergence of the sequence $\{x^k\}_{k\in\mathbb{N}}$.

Proposition 5.7 (Relative error condition)**.** *Consider Algorithm 5.1. Under Assumption 5.2, there exists $w^{k+1} \in \partial h(x^{k+1})$ such that the relative error condition* (5.5) *is satisfied with*

$$b = 2c + \sqrt{m-1}L > 0. \tag{5.8}$$

Proof. Iterate x^{k+1} in Algorithm 5.1 can be characterized via the subdifferential of the associated objective function, *i.e.*

$$0 \in \sum_{i=1}^{m} \nabla f_i(x_i^{k+1}; x_{-i}^k) + \partial g(x^{k+1}) + 2c(x^{k+1} - x^k),$$

which ensures the existence of some $v^{k+1} \in \partial g(x^{k+1})$ such that

$$\begin{aligned}
0 &= \sum_{i=1}^{m} \nabla f_i(x_i^{k+1}; x_{-i}^k) + v^{k+1} + 2c(x^{k+1} - x^k) \\
&= \Big[\sum_{i=1}^{m} \nabla f_i(x_i^{k+1}; x_{-i}^k) - \sum_{i=1}^{m} \nabla f_i(x_i^{k+1}; x_{-i}^{k+1})\Big] \\
&\quad + \sum_{i=1}^{m} \nabla f_i(x_i^{k+1}; x_{-i}^{k+1}) + v^{k+1} + 2c(x^{k+1} - x^k) \\
&= \Big[\sum_{i=1}^{m} \nabla f_i(x_i^{k+1}; x_{-i}^k) - \sum_{i=1}^{m} \nabla f_i(x_i^{k+1}; x_{-i}^{k+1})\Big] \\
&\quad + \nabla f(x^{k+1}) + v^{k+1} + 2c(x^{k+1} - x^k).
\end{aligned}$$

Notice that in the last equality we used the identity $\sum_{i=1}^m \nabla f_i(x_i; x_{-i}) = \nabla f(x)$. Let us now define $w^{k+1} := \nabla f(x^{k+1}) + v^{k+1} \in \partial h(x^{k+1})$. From the above equality we can bound the norm of w^{k+1} as

$$\begin{aligned}\|w^{k+1}\| &= \left\| \sum_{i=1}^m \nabla f_i(x_i^{k+1}; x_{-i}^k) - \sum_{i=1}^m \nabla f_i(x_i^{k+1}; x_{-i}^{k+1}) + 2c(x^{k+1} - x^k) \right\| \\ &\leq \left\| \sum_{i=1}^m \nabla f_i(x_i^{k+1}; x_{-i}^k) - \sum_{i=1}^m \nabla f_i(x_i^{k+1}; x_{-i}^{k+1}) \right\| + 2c\|x^{k+1} - x^k\|.\end{aligned}$$

The last step follows from the triangle inequality, and due to Lemma 5.5 we obtain

$$\|w^{k+1}\| \leq (2c + \sqrt{m-1}L)\, \|x^{k+1} - x^k\|. \qquad \square$$

Propositions 5.6 and 5.7 show that the conditions of Theorem 5.4 are satisfied. As a direct consequence the iterates generated by Algorithm 5.1 converge to some minimizer of (5.1), thus concluding the proof of Theorem 5.3.

5.4 Convergence rate analysis

It is shown in [62] that if f is a strongly convex quadratic function and g_i are indicator functions of convex compact sets, then Algorithm 5.1 converges linearly. We show in this section that Algorithm 5.1 converges linearly under much milder assumptions. In particular, if h satisfies the *quadratic growth condition*, *i.e.* if p in (5.2) is equal to 2, then Algorithm 5.1 admits a linear convergence rate. This property is employed in [129] to establish linear convergence of some first-order methods in a single-agent setting, and is, according to [74, 175] closely related to the *error bound*, which was used in [118, 168] to establish linear convergence of feasible descent methods. Note that the feasible descent methods are not applicable to problem (5.1) since we allow for nondifferentiable objective functions.

Theorem 5.8. *Consider Assumption 5.2, and further assume that power-type growth condition is satisfied with $p = 2$. Let the initial iterate of Algorithm 5.1 be selected such that $h(x^0) < h^\star + r$, where r appears in Definition 5.1. Then the iterates $\{x^k\}_{k\in\mathbb{N}}$ converge to some $x^\star \in \operatorname{argmin} h$, and for all $k \geq 1$*

$$h(x^k) - h^\star \leq \left(\frac{1}{1 + \gamma a b^{-2}}\right)^k \left(h(x^0) - h^\star\right), \tag{5.9}$$

$$\|x^k - x^\star\| \leq M \left(\frac{1}{\sqrt{1 + \gamma a b^{-2}}}\right)^k, \tag{5.10}$$

where

$$M = \left(\frac{2b}{\gamma a} + \frac{1}{\sqrt{a\,(1 + \gamma a b^{-2})}}\right) \sqrt{h(x^0) - h^\star}.$$

Proof. The quadratic growth condition and convexity of h, together with the relative error condition (5.5) imply that for $x^{k+1} \notin \operatorname{argmin} h$, $\bar{x}^{k+1} = \Pi_{\operatorname{argmin} h}(x^{k+1})$ and any $w^{k+1} \in \partial h(x^{k+1})$, we have

$$\begin{aligned} \gamma \operatorname{dist}_{\operatorname{argmin} h}(x^{k+1})^2 &\leq h(x^{k+1}) - h^\star \\ &\leq \left\langle w^{k+1}, x^{k+1} - \bar{x}^{k+1} \right\rangle \\ &\leq \|w^{k+1}\| \, \|x^{k+1} - \bar{x}^{k+1}\| \\ &= \|w^{k+1}\| \operatorname{dist}_{\operatorname{argmin} h}(x^{k+1}) \\ &\leq b \, \|x^{k+1} - x^k\| \operatorname{dist}_{\operatorname{argmin} h}(x^{k+1}). \end{aligned} \tag{5.11}$$

Note that since h is lower semicontinuous, the set $\operatorname{argmin} h$ is closed and thus the projection onto $\operatorname{argmin} h$ is well defined. From the right-hand sides of the first and last inequalities in (5.11) we have

$$h(x^{k+1}) - h^\star \leq b \, \|x^{k+1} - x^k\| \operatorname{dist}_{\operatorname{argmin} h}(x^{k+1}).$$

Dividing the left-hand side of the first inequality and the right-hand side of the last inequality in (5.11) by $\gamma \operatorname{dist}_{\operatorname{argmin} h}(x^{k+1}) > 0$, we obtain

$$\operatorname{dist}_{\operatorname{argmin} h}(x^{k+1}) \leq b\gamma^{-1} \, \|x^{k+1} - x^k\|.$$

Substituting this inequality into the preceding one, we obtain

$$\begin{aligned} h(x^{k+1}) - h^\star &\leq b^2\gamma^{-1} \, \|x^{k+1} - x^k\|^2 \\ &\leq b^2\gamma^{-1}a^{-1} \left(h(x^k) - h(x^{k+1}) \right) \\ &= b^2\gamma^{-1}a^{-1} \left((h(x^k) - h^\star) - (h(x^{k+1}) - h^\star) \right), \end{aligned}$$

where the second inequality follows from the sufficient decrease condition (5.4). Rearranging the terms, we have

$$h(x^{k+1}) - h^\star \leq \frac{1}{1 + \gamma a b^{-2}} \left(h(x^k) - h^\star \right),$$

for all $k \geq 0$, or equivalently

$$h(x^k) - h^\star \leq \left(\frac{1}{1 + \gamma a b^{-2}} \right)^k \left(h(x^0) - h^\star \right),$$

which proves (5.9). Substituting the above inequality into (5.6) we obtain (5.10), which concludes the proof. □

A direct consequence of Theorem 5.8 is that Algorithm 5.1, with c selected as in Theorem 5.3, converges linearly when h satisfies the quadratic growth condition

$$h(x) - h^\star \geq \gamma \operatorname{dist}_{\operatorname{argmin} h}(x)^2. \tag{5.12}$$

This is the case when h is σ–strongly convex, implying that $\operatorname{argmin} h$ is a singleton and h satisfies the quadratic growth condition for all $x \in \operatorname{dom} h$ with $\gamma = \sigma/2$. It is shown in [25, 168] that if $f(x) = v(Ex) + \langle b, x \rangle$ is Lipschitz smooth, with v being a strongly convex function, and g being an indicator function of a convex polyhedral set, then the problem satisfies the quadratic growth condition.

Note that if E does not have full column rank, then f is not strongly convex. In [25, 39] it is shown that a similar bound can be established for the ℓ_1–regularized least-squares problem. Here, we adopt the approach from [39] and show that a similar result can be provided for more general problems in which g can be any polyhedral function. The core idea is to rewrite the problem in epigraph form for which such a property is shown to hold.

We impose the following assumption:

Assumption 5.9.

(i) The function f is defined as $f(x) = v(Ex) + \langle b, x \rangle$, with v being σ_v–strongly convex.

(ii) The component functions g_i are all globally non-negative convex polyhedral functions whose composite epigraph can be represented as

$$\operatorname{epi} g = \{(x, t) \in \mathbb{R}^n \times \mathbb{R} \,|\, Cx + ct \leq d\},$$

where $C \in \mathbb{R}^{p \times n}$, $c \in \mathbb{R}^p$ and $d \in \mathbb{R}^p$.

The conditions of Assumption 5.9 are satisfied when f is quadratic and g_i are indicator functions of convex polyhedral sets, or any polyhedral norms. Note that the dual of a quadratic program (QP) satisfies this assumption. The Lipschitz constant of ∇f, which is required for computing the appropriate parameter c for Algorithm 5.1, can be upper-bounded by $\|E\|^2 L_v$, where $\|E\|$ is the spectral norm of E, and L_v is the Lipschitz constant of ∇v.

Let $x^0 \in \operatorname{dom} h$ be an initial iterate of the algorithm and let $r = h(x^0)$. Since h is coercive, $[h \leq r]$ is a compact set and we can thus define the following quantities:

$$\begin{aligned} D^r &\coloneqq \max_{(x,y) \in [h \leq r]^2} \|x - y\|, \\ D_E^r &\coloneqq \max_{(x,y) \in [h \leq r]^2} \|Ex - Ey\| \leq D \, \|E\|, \\ V^r &\coloneqq \max_{x \in [h \leq r]} \|\nabla v(Ex)\|. \end{aligned}$$

Proposition 5.10. *Let $r = h(x^0)$ and fix any $R > g(x^0) + V^r D_E^r + \|b\| D^r$. Under Assumptions 5.2 and 5.9, for all $x \in [h \leq r]$ we have*

$$h(x) - h^\star \geq \kappa_R^{-1} \operatorname{dist}_{\operatorname{argmin} h}(x)^2,$$

where $\kappa_R > 0$ is some problem dependent constant.

Proof. See Appendix 5.A.2. □

5.5 Conclusions

In this chapter we revisited the regularized Jacobi algorithm proposed in [51], and enhanced its convergence properties. It was shown that iterates generated by the algorithm converge to a minimizer of the centralized problem counterpart, provided that the objective function satisfies a power growth condition. We also established linear convergence of the algorithm when the power growth condition satisfied by the objective function is quadratic.

5.A Proofs

5.A.1 Problem minimizer as an algorithm fixed-point

In this section we show that the set of fixed-points of Algorithm 5.1 coincides with the set of minimizers of problem (5.1). The result follows the same line of argument as [62, §3]; however, the proof is modified to account for the presence of the nondifferentiable terms g_i.

Similarly to [62], we define an operator T such that

$$T(x) = \operatorname*{argmin}_z \left\{ \sum_{i=1}^{m} f_i(z_i; x_{-i}) + g(z) + c\|z - x\|^2 \right\} \tag{5.13}$$

and operators $T_i(y_{-i})$ such that

$$T_i(x_i; y_{-i}) = \operatorname*{argmin}_{z_i} \Big\{ f_i(z_i; y_{-i}) + g_i(z_i) + c\|z_i - x_i\|^2 \Big\},$$

where $y_{-i} \in \mathbb{R}^{n-n_i}$ is treated as a fixed parameter. Observe that we can characterize the operator T via the operators $T_i(x_{-i})$ as follows

$$T(x) = \Big(T_1(x_1; x_{-1}), \ldots, T_m(x_m; x_{-m}) \Big).$$

We define the sets of fixed-points for these operators as

$$\begin{aligned} \operatorname{Fix} T &\coloneqq \{x \,|\, x = T(x)\}, \\ \operatorname{Fix} T_i(y_{-i}) &\coloneqq \{x_i \,|\, x_i = T_i(x_i; y_{-i})\}, \quad i = 1, \ldots, m. \end{aligned}$$

Note that, in the spirit of [152, §5], we treat T as a single valued function $T : \mathbb{R}^n \mapsto \mathbb{R}^n$ since the quadratic term in the right hand side of (5.13) ensures that the set of minimizers is always single-valued, with an identical comment applying to the operators $T_i(y_{-i})$.

We now show that the sets $\operatorname{argmin} h$ and $\operatorname{Fix} T$ coincide.

Proposition 5.11. *If Assumption 5.2 holds, then* $\operatorname{argmin} h = \operatorname{Fix} T$.

Proof. The proof is based on [62, proofs of Prop. 1–3]. We first show that $\operatorname{argmin} h \subseteq \operatorname{Fix} T$. Fix any $x \in \operatorname{argmin} h$. If x minimizes h, then it is also a block-wise minimizer of h at x, *i.e.* for all $i = 1, \ldots, m$, we have

$$x_i \in \operatorname*{argmin}_{z_i} \Big\{ f_i(z_i; x_{-i}) + g_i(z_i) \Big\}. \tag{5.14}$$

Since x_i minimizes both $\Big(f_i(\,\cdot\,; x_{-i}) + g_i \Big)$ and $c\|(\,\cdot\,) - x_i\|^2$, it is also the unique minimizer of their sum, *i.e.*

$$x_i = \operatorname*{argmin}_{z_i} \Big\{ f_i(z_i; x_{-i}) + g_i(z_i) + c\|z_i - x_i\|^2 \Big\},$$

implying that $x_i \in \operatorname{Fix} T_i(x_{-i})$, and thus $x = (x_1, \ldots, x_m)$ is a fixed-point of $T(x) = \big(T_1(x_1; x_{-1}), \ldots, T_m(x_m; x_{-m})\big)$.

We now show that $\operatorname{Fix} T \subseteq \operatorname{argmin} h$. Let $x \in \operatorname{Fix} T$, and thus for all $i = 1, \ldots, m$, $x_i \in \operatorname{Fix} T_i(x_{-i})$, *i.e.*

$$x_i = \operatorname*{argmin}_{z_i} \Big\{ f_i(z_i; x_{-i}) + g_i(z_i) + c\|z_i - x_i\|^2 \Big\}.$$

The above condition implies that x_i is a stationary point, and thus for all $z_i \in \mathbb{R}^{n_i}$ we have

$$\langle \nabla f_i(x_i; x_{-i}), z_i - x_i \rangle + g_i'(x_i, z_i - x_i) + \Big\langle 2c \underbrace{(x_i - x_i)}_{=0}, z_i - x_i \Big\rangle \geq 0,$$

or equivalently for all $d_i \in \mathbb{R}^{n_i}$

$$\langle \nabla f_i(x_i; x_{-i}), d_i \rangle + g_i'(x_i, d_i) \geq 0.$$

Since both f_i and g_i are convex, the above condition implies that x_i is a minimizer of $\big(f_i(\,\cdot\,; x_{-i}) + g_i\big)$. According to [163, Lem. 3.1] differentiability of f and component-wise separability of g imply that any $x = (x_1, \ldots, x_m)$ for which (5.14) holds for all $i = 1, \ldots, m$, is also a minimizer of $(f + g)$, *i.e.* $x \in \operatorname{argmin} h$, thus concluding the proof. □

5.A.2 Proof of Proposition 5.10

We first define the Hoffman constant which will be used in the further analysis.

Lemma 5.12 (Hoffman constant, see *e.g.* [25])**.** *Let X and Y be two convex polyhedra defined as*

$$X = \{x \in \mathbb{R}^n \mid Ax \leq a\}, \quad Y = \{x \in \mathbb{R}^n \mid Ex = e\},$$

where $A \in \mathbb{R}^{m \times n}$, $a \in \mathbb{R}^m$, $E \in \mathbb{R}^{p \times n}$, $e \in \mathbb{R}^p$, and assume that $X \cap Y \neq \emptyset$. Then there exists a constant $\theta = \theta(A, E)$ such that every $x \in X$ satisfies

$$\operatorname{dist}_{X \cap Y}(x) \leq \theta \, \|Ex - e\|.$$

We refer to θ as the Hoffman constant *associated with matrix $[A^T, E^T]^T$.*

Since Algorithm 5.1 generates a non-increasing sequence $\{h(x^k)\}_{k \in \mathbb{N}}$, we have $x^k \in [h \leq r]$ for all k and

$$\begin{aligned} g(x^k) &\leq g(x^0) + f(x^0) - f(x^k) \\ &\leq g(x^0) + v(Ex^0) - v(Ex^k) + \big\langle b, x^0 - x^k \big\rangle \\ &\leq g(x^0) + \|\nabla v(Ex^0)\| \, \|Ex^0 - Ex^k\| + \|b\| \, \|x^0 - x^k\| \\ &\leq g(x^0) + V^r D_E^r + \|b\| D^r, \end{aligned}$$

where the third inequality follows from convexity of v. We conclude that $\operatorname{argmin} h \subseteq [h \leq r] \subset [g \leq R]$, for any fixed $R > g(x^0) + V^r D_E^r + \|b\| D^r$. For such a bound R, we have

$$\begin{aligned}
&\min\left\{v(Ex) + \langle b, x\rangle + g(x) \mid x \in \mathbb{R}^n\right\}\\
=&\min\left\{v(Ex) + \langle b, x\rangle + t \quad \mid (x,t) \in \mathbb{R}^n \times \mathbb{R},\, g(x) \leq R,\, t = g(x)\right\}\\
=&\min\left\{v(Ex) + \langle b, x\rangle + t \quad \mid (x,t) \in \mathbb{R}^n \times \mathbb{R},\, g(x) \leq t,\, t \leq R\right\}\\
=&\min\Big\{\underbrace{v(\tilde{E}\tilde{x}) + \left\langle \tilde{b}, \tilde{x}\right\rangle}_{=:\tilde{h}(\tilde{x})} \quad \mid \tilde{x} \in \tilde{\mathcal{X}} := \{M\tilde{x} \leq \tilde{R}\}\Big\},
\end{aligned} \tag{5.15}$$

where $\tilde{x} = (x,t)$ and

$$\tilde{E} = \begin{bmatrix} E & 0 \end{bmatrix}, \quad \tilde{b} = \begin{bmatrix} b \\ 1 \end{bmatrix}, \quad M = \begin{bmatrix} C & c \\ 0 & 1 \end{bmatrix}, \quad \tilde{R} = \begin{bmatrix} d \\ R \end{bmatrix}.$$

It can be easily seen that $\tilde{x}^\star = (x^\star, t^\star)$ minimizes (5.15) if and only if $x^\star \in \operatorname{argmin} h$ and $t^\star = g(x^\star)$. Using [25, Lem. 2.5], we obtain

$$\operatorname{dist}_{\operatorname{argmin}\tilde{h}}(\tilde{x})^2 \leq \kappa_R\left(\tilde{h}(\tilde{x}) - \tilde{h}^\star\right), \quad \forall \tilde{x} \in \tilde{\mathcal{X}}, \tag{5.16}$$

where $\kappa_R = \theta^2\left(\|\tilde{b}\|\tilde{D}^R + 3\tilde{V}^R\tilde{D}_{\tilde{E}}^R + 2\sigma_v^{-1}\left((\tilde{V}^R)^2 + 1\right)\right)$ with θ being the Hoffman constant associated with matrix $[M^T, \tilde{E}^T, \tilde{b}]^T$ and

$$\begin{aligned}
\tilde{D}^R &:= \max_{(\tilde{x},\tilde{y}) \in \tilde{\mathcal{X}}^2} \|\tilde{x} - \tilde{y}\| \leq \max_{(x,y)\in[g\leq R]^2} \|x - y\| + \max_{(t,s)\in[0,R]^2} \|t - s\| = D^R + R,\\
\tilde{V}^R &:= \max_{\tilde{x}\in\tilde{\mathcal{X}}} \|\nabla v(\tilde{E}\tilde{x})\| = \max_{x\in[g\leq R]} \|\nabla v(Ex)\| = V^R,\\
\tilde{D}_{\tilde{E}}^R &:= \max_{(\tilde{x},\tilde{y}) \in \tilde{\mathcal{X}}^2} \|\tilde{E}\tilde{x} - \tilde{E}\tilde{y}\| = \max_{(x,y)\in[g\leq R]^2} \|Ex - Ey\| = D_E^R.
\end{aligned}$$

Inequality (5.16) implies that for all $x \in [g \leq R]$ and for all $t \in [0, R]$

$$\operatorname{dist}_{\operatorname{argmin} h}(x)^2 + \|t - t^\star\|^2 \leq \kappa_R\left(f(x) + t - f(x^\star) - t^\star\right).$$

Setting $t = g(x)$, we then have

$$\begin{aligned}
\operatorname{dist}_{\operatorname{argmin} h}(x)^2 &\leq \operatorname{dist}_{\operatorname{argmin} h}(x)^2 + \|t - t^\star\|^2\\
&\leq \kappa_R\left(h(x) - h^\star\right).
\end{aligned}$$

This concludes the proof.

6

Operator Splitting Solver for Quadratic Programs

Contents

In this chapter we present a new general-purpose solver for quadratic programs (QPs) based on the algorithm introduced in Chapter 3. The solver supports factorization caching and warm starting, making it particularly efficient for solving multiple instances of a parametric program.

6.1 Introduction

QPs arise in many applications including control [2, 88, 145], signal processing [46, §6.3.3], [123], statistics [48, 105, 106, 161], finance [42, 44, 55, 120], and machine learning [56]. The numerical solution of QP subproblems is also an essential component in nonconvex optimization methods such as sequential quadratic programming (SQP) [134, §18] and mixed-integer optimization using branch-and-bound techniques [28, 84].

6.1.1 Related work

Active set methods were the first popular solution methods for QPs [171] obtained as an extension of Dantzig's simplex method for linear programs (LPs) [57]. Active set methods select an active set (*i.e.* a set of binding constraints) and solve the resulting equality constrained QP. The active set is then updated iteratively by adding and dropping constraints [134, §16.5]. These methods can be warm-started to reduce the number of active set recalculations required. However, the major drawback of active set methods is their worst-case complexity that grows exponentially with the number of constraints, since it may be necessary to investigate all possible active sets before reaching the optimal [113]. Modern implementations of active set methods for solving QPs can be found in most commercial solvers such as MOSEK [126] and GUROBI [100], and in the open-source solver qpOASES [83].

Interior-point algorithms gained popularity as methods for solving LPs in polynomial time [91, 110]. These techniques were then extended for solving more general convex problems including QPs [131]. Interior-point methods model the problem constraints as parametrized penalty functions, also referred to as *barrier functions*. A series of unconstrained optimization problems are solved for varying barrier function parameters until the optimal solution is achieved; see [46, §11] and [134, §16.6] for details. Primal-dual interior-point methods, in particular Mehrotra's predictor-corrector method [124], became the algorithms of choice for practical implementation because of their good performance across a wide range of problems [172]. However, these methods are not easily warm-started and do not scale well with the problem dimensions. Interior-point methods are currently the default algorithms in the commercial solvers MOSEK [126] and GUROBI [100], and in the open-source solver OOQP [89].

First-order methods compute a solution of an optimization problem using only first-order information of the problem functions. Operator splitting methods, such as the Douglas-Rachford splitting (DRS) and the alternating direction method of multipliers (ADMM), are a particular class of first-order methods which model an optimization problem as the problem of finding a zero of the sum of monotone operators [80, 115]. In recent years, ADMM has received particular attention because of its very good practical performance [45]. The method was shown to provide modest accuracy solutions in a relatively small number of computationally cheap iterations, and is thus well suited to embedded and large-scale optimization.

6.1.2 Proposed approach

We present a novel general-purpose QP solver based on ADMM. The proposed algorithm employs a novel splitting requiring the solution of a quasi-definite linear system that is solvable for any choice of problem data. We therefore pose no constraints such as strong convexity of the objective function or linear independence of the constraints. Since the linear system solved in each iteration of the algorithm involves a matrix that does not change across iterations, we perform factorization only once at the beginning of the algorithm. In contrast to other first-order methods, our approach not only returns primal and dual solutions when the problem is solvable, but also provides certificates of primal or dual infeasibility otherwise, without resorting to the homogeneous self-dual embedding (HSDE).

Our algorithm can be warm-started from a tentative solution to reduce the number of iterations. Moreover, if problem matrices do not change across multiple problem instances, then the same matrix factorization can be reused, thus reducing the computation time considerably. This feature is particularly useful when solving parametric QPs where only few elements of the problem data change.

We implemented our method in the open-source "Operator Splitting Quadratic Program" (OSQP) solver. OSQP is written in C and can be compiled to be library free. It is robust against noisy problem data, has a very small code footprint, and is suitable for both embedded and large-scale applications.

6.2 Problem description

We consider the following QP:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^T P x + q^T x \\ \text{subject to} & l \leq Ax \leq u, \end{array} \tag{6.1}$$

where $x \in \mathbb{R}^n$ is the optimization variable. The objective function is defined by a matrix $P \in \mathbb{S}^n_+$ and a vector $q \in \mathbb{R}^n$, and the constraints by a matrix $A \in \mathbb{R}^{m \times n}$ and vectors $l \in \tilde{\mathbb{R}}^m$ and $u \in \tilde{\mathbb{R}}^m$ such that $l \leq u$. Linear equality constraints can be enforced by setting $l_i = u_i \in \mathbb{R}$.

6.2.1 Optimality conditions

Since problem (6.1) is a special case of problem (3.2) when $\mathcal{C} = [l, u]$, we can write the optimality conditions as

$$Ax = z, \tag{6.2a}$$

$$Px + q + A^T y = 0, \tag{6.2b}$$

$$l \leq z \leq u, \tag{6.2c}$$
$$y_+^T(z-u) = 0, \tag{6.2d}$$
$$y_-^T(z-l) = 0, \tag{6.2e}$$

where $z \in \mathbb{R}^m$ is an auxiliary optimization variable, $y \in \mathbb{R}^m$ is a Lagrange multiplier associated with the constraint $Ax = z$, $y_+ \coloneqq \max(y, 0)$ and $y_- \coloneqq \min(y, 0)$. If there exist $x \in \mathbb{R}^n$, $z \in \mathbb{R}^m$ and $y \in \mathbb{R}^m$ that satisfy the conditions above, then we say that x is a *primal* and y is a *dual* solution of problem (6.1).

6.2.2 Infeasibility certificate

If there exists some $\bar{y} \in \mathbb{R}^m$ satisfying the following conditions:

$$A^T\bar{y} = 0, \quad u^T\bar{y}_+ + l^T\bar{y}_- < 0, \tag{6.3}$$

then problem (6.1) is primal infeasible. Similarly, if there exists some $\bar{x} \in \mathbb{R}^n$ such that

$$P\bar{x} = 0, \quad q^T\bar{x} < 0, \quad (A\bar{x})_i \begin{cases} = 0 & l_i \in \mathbb{R},\, u_i \in \mathbb{R} \\ \geq 0 & l_i \in \mathbb{R},\, u_i = +\infty \\ \leq 0 & u_i \in \mathbb{R},\, l_i = -\infty \end{cases} \tag{6.4}$$

for all $i = 1, \ldots, m$, then the problem is dual infeasible. We call such $\bar{y}$ and $\bar{x}$ *certificates of primal and dual infeasibility*, respectively. Note that since the constraints in (6.1) are polyhedral, conditions (6.3) and (6.4) are *strong alternatives* for primal and dual feasibility, respectively. We refer the reader to Section 3.2.2 for more details.

6.3 Solution with ADMM

Our method for solving QPs is based on Algorithm 3.1 introduced in Chapter 3. Note that the step 5 of the algorithm involves evaluating the Euclidean projection onto the set $[l, u]$, which has a simple closed-form solution

$$\Pi_{[l,u]}(z) = \max\left(\min(z, u), l\right).$$

We next describe how to solve the step 3 of the algorithm efficiently.

6.3.1 Solving the linear system

The step 3 in Algorithm 3.1 involves solving the following equality constrained QP:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\tilde{x}^T P\tilde{x} + q^T\tilde{x} + \frac{\sigma}{2}\|\tilde{x} - x^k\|_2^2 + \frac{\rho}{2}\|\tilde{z} - z^k + \rho^{-1}y^k\|_2^2 \\ \text{subject to} & A\tilde{x} = \tilde{z}. \end{array}$$

Optimality conditions for this problem are given by

$$\begin{aligned} P\tilde{x} + q + \sigma(\tilde{x} - x^k) + A^T\nu^{k+1} &= 0, \\ \rho(\tilde{z} - z^k) + y^k - \nu &= 0, \\ A\tilde{x} &= \tilde{z}, \end{aligned}$$

where $\nu \in \mathbb{R}^m$ is a Lagrange multiplier associated with the constraint $A\tilde{x} = \tilde{z}$. By eliminating the variable $\tilde{z}$ from the above linear system, it reduces to

$$\begin{bmatrix} P + \sigma I & A^T \\ A & -\rho^{-1}I \end{bmatrix} \begin{bmatrix} \tilde{x} \\ \nu \end{bmatrix} = \begin{bmatrix} \sigma x^k - q \\ z^k - \rho^{-1}y^k \end{bmatrix}, \tag{6.5}$$

where $\tilde{z}$ can then be recovered from

$$\tilde{z} = z^k + \rho^{-1}(\nu - y^k).$$

We refer to the coefficient matrix in (6.5) as the *KKT matrix*. We can solve the linear system in (6.5) using either a *direct* or an *indirect method*.

Direct method

The direct method computes the exact solution of the linear system (6.5) by first computing a factorization of the KKT matrix and then performing forward and backward substitutions. Since the KKT matrix does not depend on the iteration counter k, we perform the factorization only once at the beginning of the algorithm and cache the factors so that we can reuse them in subsequent iterations. This approach is very efficient when the factorization cost is considerably higher than the cost of forward and backward substitutions, so that each ADMM iteration is computed quickly.

The KKT matrix is quasi-definite, *i.e.* it is a 2-by-2 block-symmetric matrix with the $(1,1)$-block positive definite, and the $(2,2)$-block negative definite. It thus always has a well defined LDL^T factorization, with L being a lower triangular matrix with unit diagonal elements, and D a diagonal matrix with nonzero diagonal elements [165]. Note that, once the factorization is carried out, computing the solution of (6.5) can be made division-free by storing D^{-1} instead of D.

For any sparse quasi-definite matrix K, efficient algorithms can be used to compute a permutation matrix P for which the factorization $PKP^T = LDL^T$ results in

a sparse factor L [59]. The LDL^T factorization consists of two steps. First, the sparsity pattern of the factor L is found before performing any numerical operations. Determining this sparsity pattern is known as *symbolic factorization* and requires only the nonzero structure of the matrix K, and not its numerical values. After the symbolic factorization finds the pattern of nonzero entries in L, the numerical values of these elements are computed. This procedure is known as *numerical factorization*. Note that if the nonzero entries in the matrix K change, but the sparsity pattern and quasi-definiteness are preserved, then only the numerical factorization step needs to be performed again and the memory required to store the new factorization does not change.

Indirect method

We can find the solution of (6.5) by solving instead the following linear system:

$$\left(P + \sigma I + \rho A^T A\right) \tilde{x} = \sigma x^k - q + A^T(\rho z^k - y^k),$$

obtained by eliminating ν from (6.5). Note that the coefficient matrix in the above linear system is always positive definite. The linear system can thus be solved with an iterative scheme such as the conjugate gradient method [96, 134]. When the linear system is solved up to some predefined accuracy, we terminate the method. We can also warm start the method using the linear system solution at the previous iteration of ADMM to reduce its computation time. Solving the linear system using an indirect method is beneficial when the dimensions of the problem are very large, making the computational cost of matrix factorization prohibitively expensive.

6.3.2 Final algorithm

The proposed method for solving convex QPs is given in Algorithm 6.1. Scalars $\rho > 0$ and $\sigma > 0$ are the *penalty parameters*, and $\alpha \in (0, 2)$ is the *relaxation parameter*. Steps 4–7 of the algorithm are very easy to evaluate since they involve only vector additions and subtractions, scalar-vector multiplications and the projection operator $\Pi_{[l,u]}$. Moreover, they are component-wise separable and can be easily parallelized. The most computationally intensive step is solving the linear system in step 3.

6.3.3 Convergence and infeasibility detection

Since Algorithm 6.1 is a special case of Algorithm 3.1 when $\mathcal{C} = [l, u]$, we can exploit asymptotic results presented in Section 3.3 and Section 3.4. First, note that the

Algorithm 6.1 OSQP.

given initial values x^0, z^0, y^0 and parameters $\rho > 0$, $\sigma > 0$, $\alpha \in (0, 2)$
repeat
  $(\tilde{x}^{k+1}, \nu^{k+1}) \leftarrow$ solve $\begin{bmatrix} P + \sigma I & A^T \\ A & -\rho^{-1} I \end{bmatrix} \begin{bmatrix} \tilde{x}^{k+1} \\ \nu^{k+1} \end{bmatrix} = \begin{bmatrix} \sigma x^k - q \\ z^k - \rho^{-1} y^k \end{bmatrix}$
  $\tilde{z}^{k+1} \leftarrow z^k + \rho^{-1}(\nu^{k+1} - y^k)$
  $x^{k+1} \leftarrow \alpha \tilde{x}^{k+1} + (1 - \alpha) x^k$
  $z^{k+1} \leftarrow \Pi_{[l,u]} \left(\alpha \tilde{z}^{k+1} + (1 - \alpha) z^k + \rho^{-1} y^k \right)$
  $y^{k+1} \leftarrow y^k + \rho \left(\alpha \tilde{z}^{k+1} + (1 - \alpha) z^k - z^{k+1} \right)$
  $k \leftarrow k + 1$
until termination criterion is satisfied

pair (z^{k+1}, y^{k+1}) obtained from Algorithm 6.1 satisfies the optimality conditions (6.2c)–(6.2e) by construction. Therefore, we define the *primal* and *dual residuals* as

$$r^k_{\text{prim}} := Ax^k - z^k, \tag{6.6}$$
$$r^k_{\text{dual}} := Px^k + q + A^T y^k, \tag{6.7}$$

which quantify the level of suboptimality of an iterate (x^k, z^k, y^k).

According to Proposition 3.9, if problem (6.1) is solvable, then the primal and dual residuals converge to zero as $k \to \infty$. Alternatively, according to Theorem 3.10, if the problem is unsolvable, then the sequences $\{\delta y^k\}_{k \in \mathbb{N}}$ and $\{\delta x^k\}_{k \in \mathbb{N}}$ converge to certificates of primal and dual infeasibility, respectively.

6.3.4 Termination criteria

We can define termination criteria for Algorithm 6.1 so that the iterations stop when either a primal-dual solution or a certificate of primal or dual infeasibility is found with some predefined accuracy.

A reasonable termination criterion for detecting optimality is that the norms of the residuals r^k_{prim} and r^k_{dual} are smaller than some tolerance levels $\varepsilon_{\text{prim}} > 0$ and $\varepsilon_{\text{dual}} > 0$ [45], *i.e.*

$$\|r^k_{\text{prim}}\|_\infty \leq \varepsilon_{\text{prim}}, \quad \|r^k_{\text{dual}}\|_\infty \leq \varepsilon_{\text{dual}}. \tag{6.8}$$

We set the tolerance levels as

$$\varepsilon_{\text{prim}} := \varepsilon_{\text{abs}} + \varepsilon_{\text{rel}} \max\{\|Ax^k\|_\infty, \|z^k\|_\infty\}$$
$$\varepsilon_{\text{dual}} := \varepsilon_{\text{abs}} + \varepsilon_{\text{rel}} \max\{\|Px^k\|_\infty, \|A^T y^k\|_\infty, \|q\|_\infty\},$$

where $\varepsilon_{\text{abs}} > 0$ and $\varepsilon_{\text{rel}} > 0$ are *absolute* and *relative tolerances*, respectively.

If $\|\delta y^k\|_\infty > \varepsilon_{\text{pinf}}$ for some tolerance level $\varepsilon_{\text{pinf}} > 0$, then we define the following criterion for detecting primal infeasibility:

$$\|A^T \delta\hat{y}^k\|_\infty \leq \varepsilon_{\text{pinf}}, \quad u^T(\delta\hat{y}^k)_+ + l^T(\delta\hat{y}^k)_- \leq \varepsilon_{\text{pinf}},$$

where $\delta\hat{y}^k \coloneqq \delta y^k / \|\delta y^k\|_\infty$.

Similarly, if $\|\delta x^k\|_\infty > \varepsilon_{\text{dinf}}$ for some tolerance level $\varepsilon_{\text{dinf}} > 0$, then we detect dual infeasibility using the following criterion:

$$\|P\delta\hat{x}^k\|_\infty \leq \varepsilon_{\text{dinf}}, \quad q^T\delta\hat{x}^k \leq \varepsilon_{\text{dinf}}, \quad (A\delta\hat{x}^k)_i \begin{cases} \in [-\varepsilon_{\text{dinf}}, \varepsilon_{\text{dinf}}] & l_i \in \mathbb{R},\ u_i \in \mathbb{R} \\ \geq -\varepsilon_{\text{dinf}} & l_i \in \mathbb{R},\ u_i = +\infty \\ \leq \varepsilon_{\text{dinf}} & u_i \in \mathbb{R},\ l_i = -\infty \end{cases}$$

for $i = 1, \ldots, m$, where $\delta\hat{x}^k \coloneqq \delta x^k / \|\delta x^k\|_\infty$.

6.4 Data preconditioning

A known weakness of first-order methods is their inability to deal effectively with ill-conditioned problems, and the convergence rate can vary significantly when data are badly scaled. Preconditioning is a common heuristic aiming to reduce the number of iterations of first-order methods [134, §5], [31, 90, 93, 141]. The optimal choice of preconditioners has been studied for at least two decades and remains an active research area [111, §2], [99, §10]. For example, the optimal diagonal preconditioner required to minimize the condition number of a matrix can be found exactly by solving a semidefinite program (SDP) [43], which is computationally more expensive than solving the original QP.

In order to keep the preconditioning procedure simple, we instead use a simple heuristic called *matrix equilibration* [47, 66, 85]. Our goal is to rescale the problem data represented by the following matrix:

$$M \coloneqq \begin{bmatrix} P & A^T \\ A & 0 \end{bmatrix}.$$

In particular, we perform *symmetric matrix equilibration* by computing the diagonal matrix $S \in \mathbb{S}^{n+m}_{++}$ so that all rows of the matrix SMS have equal norms. In addition, we normalize the objective function by multiplying it by a scalar $c > 0$.

We can write matrix S as

$$S = \begin{bmatrix} D & \\ & E \end{bmatrix},$$

where $D \in \mathbb{S}^n_{++}$ and $E \in \mathbb{S}^m_{++}$ are both diagonal. It is possible to show that finding such a scaling matrix S can be cast as a convex optimization problem

Algorithm 6.2 Modified Ruiz equilibration.

initialize $c = 1$, $D = I$, $E = I$, $\delta = 0$, $\bar{P} = P$, $\bar{q} = q$, $\bar{A} = A$
while $\|1 - \delta\|_\infty > \varepsilon_{\text{equil}}$ **do**
  $M \leftarrow \begin{bmatrix} \bar{P} & \bar{A}^T \\ \bar{A} & 0 \end{bmatrix}$
  for $i = 1, \ldots, n + m$ **do**
    $\delta_i \leftarrow 1/\sqrt{\|M_i\|_\infty}$
  end for
  $\begin{bmatrix} D \\ E \end{bmatrix} \leftarrow \text{diag}(\delta) \begin{bmatrix} D \\ E \end{bmatrix}$
  $\bar{P} \leftarrow DPD, \quad \bar{q} \leftarrow Dq, \quad \bar{A} \leftarrow EAD$ ▷ Matrix equilibration
  $\gamma \leftarrow 1/\max\{\text{mean}(\|\bar{P}_i\|_\infty), \|\bar{q}\|_\infty\}$
  $\bar{P} \leftarrow \gamma \bar{P}, \quad \bar{q} \leftarrow \gamma \bar{q}, \quad c \leftarrow \gamma c$ ▷ Cost scaling
end while
return D, E, c

[7]. However, it is computationally more convenient to solve this problem with heuristic iterative methods, rather than continuous optimization algorithms. We refer the reader to [47] for more details on matrix equilibration.

Preconditioning effectively modifies problem (6.1) into the following:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\bar{x}^T \bar{P} \bar{x} + \bar{q}^T \bar{x} \\ \text{subject to} & \bar{l} \leq \bar{A}\bar{x} \leq \bar{u}, \end{array} \tag{6.9}$$

where the optimization variables are $\bar{x} = D^{-1}x$, $\bar{z} = Ez$ and $\bar{y} = cE^{-1}y$, respectively, and the problem data are

$$\bar{P} = cDPD, \quad \bar{q} = cDq, \quad \bar{A} = EAD, \quad \bar{l} = El, \quad \bar{u} = Eu.$$

6.4.1 Ruiz equilibration

We apply a variation of the Ruiz equilibration [153]. This technique was shown to converge faster than other methods such as the Sinkhorn-Knopp equilibration [156]. The steps of the method are outlined in Algorithm 6.2 and differ from the original Ruiz algorithm by adding a cost scaling step that takes into account very large values of the cost, and thus controls the norm of the dual variables.

The first part of the algorithm is the usual Ruiz equilibration step. Since M is symmetric, we focus only on the rows M_i and apply the scaling to both sides of M. The second part is the cost scaling step. The scalar γ is the current cost normalization coefficient taking into account maximum between the average norm of rows of $\bar{P}$ and the norm of $\bar{q}$.

6.4.2 Unscaled termination criteria

Although we scale problem (6.1) in the form (6.9), we would still like to apply the stopping criteria defined in Section 6.3.4 to the unscaled problem. The primal and dual residuals in (6.6) and (6.7) can be rewritten in terms of the scaled problem as

$$r^k_{\text{prim}} = E^{-1}\bar{r}^k_{\text{prim}} = E^{-1}(\bar{A}\bar{x}^k - \bar{z}^k),$$
$$r^k_{\text{dual}} = c^{-1}D^{-1}\bar{r}^k_{\text{dual}} = c^{-1}D^{-1}(\bar{P}\bar{x}^k + \bar{q} + \bar{A}^T\bar{y}^k),$$

and the tolerance levels as

$$\varepsilon_{\text{prim}} = \varepsilon_{\text{abs}} + \varepsilon_{\text{rel}} \max\{\|E^{-1}\bar{A}\bar{x}^k\|_\infty, \|E^{-1}\bar{z}^k\|_\infty\}$$
$$\varepsilon_{\text{dual}} = \varepsilon_{\text{abs}} + \varepsilon_{\text{rel}}\, c^{-1} \max\{\|D^{-1}\bar{P}\bar{x}^k\|_\infty, \|D^{-1}\bar{A}^T\bar{y}^k\|_\infty, \|D^{-1}\bar{q}\|_\infty\}.$$

Termination criteria for detecting primal and dual infeasibility remain as in Section 6.3.4, but with $\delta\hat{y}^k \coloneqq E\delta\bar{y}^k/\|E\delta\bar{y}^k\|_\infty$ and $\delta\hat{x}^k \coloneqq D\delta\bar{x}^k/\|D\delta\bar{x}^k\|_\infty$.

6.5 Parameter selection

The choice of parameters (ρ, σ, α) in Algorithm 6.1 is a key determinant of the number of iterations required to satisfy a termination criterion. Unfortunately, it is still an open research question how to select suitable ADMM parameters [90, 93]. After extensive numerical testing on millions of problem instances and a wide range of dimensions, we found rules for selecting the algorithm parameters that usually improve the convergence rate, and which we present in the sequel.

Choosing σ and α. The parameter σ ensures existence of a unique solution of the linear system in the step 3 of Algorithm 6.1, even when P is not positive definite. However, numerical experiments show that small values of σ make the algorithm converge faster. We choose σ big enough to preserve numerical stability without slowing down the algorithm. We set the default value to $\sigma = 10^{-6}$.

The relaxation parameter α in the range $[1.5, 1.8]$ has empirically shown to improve the convergence rate [78, 81]. We set the default value to $\alpha = 1.6$.

6.5.1 Choosing ρ

The most crucial parameter of the algorithm is the step-size ρ. Numerical testing showed that having different values of ρ for different types of constraints can improve the algorithm's performance considerably. For this reason, without altering the algorithm steps, we choose $\rho \in \mathbb{S}^m_{++}$ to be a diagonal matrix with positive diagonal elements ρ_i.

For a specific problem, we can achieve the optimal convergence rate of the algorithm by setting elements of matrix ρ as $\rho_i = \infty$ for active, and $\rho_i = 0$ for inactive constraints [90, §IV.D]. Unfortunately, we cannot know which constraints are active and which are inactive at the optimal solution before solving the problem. However, this rule suggests, and numerical tests confirm, that having a higher values of the parameter for equality constraints improves convergence rate of the algorithm. We thus define ρ as follows

$$\rho = \operatorname{diag}(\rho_1, \ldots, \rho_m), \qquad \rho_i = \begin{cases} \bar{\rho} & l_i \neq u_i \\ 10^3 \bar{\rho} & \text{otherwise,} \end{cases}$$

where $\bar{\rho} > 0$. In this way we assign a high value to the step-size parameter related to the equality constraints since they are always active at the solution.

Having a fixed value of $\bar{\rho}$ does not provide satisfactory performance of the algorithm across different problem classes. To compensate for this issue, we adopt an adaptive scheme which updates $\bar{\rho}$ during the iterations based on the ratio between norms of primal and dual residuals. Introducing a "feedback" in the algorithm makes ADMM more robust to ill-conditioned problems, as observed in [45, 101, 170, 174]. Contrary to the adaptation approaches in the literature where the update increases or decreases the value of the step-size by a fixed factor, we adopt the following rule:

$$\bar{\rho}^{k+1} \leftarrow \bar{\rho}^k \sqrt{\frac{\|\bar{r}^k_{\text{prim}}\|_\infty / \max\{\|\bar{A}\bar{x}^k\|_\infty, \|\bar{z}^k\|_\infty\}}{\|\bar{r}^k_{\text{dual}}\|_\infty / \max\{\|\bar{P}\bar{x}^k\|_\infty, \|\bar{A}^T\bar{y}^k\|_\infty, \|\bar{q}\|_\infty\}}}.$$

In other words we update $\bar{\rho}$ using the square root of the ratio between the scaled residuals normalized by the magnitudes of the relative parts of the tolerances. Note that the convergence results of Section 6.3.3 hold as long as the sequence $\{\bar{\rho}^k\}_{k\in\mathbb{N}}$ is lower- and upper-bounded by some positive constants. We set the initial value as $\bar{\rho}^0 = 0.1$.

The proposed rule makes our algorithm much more robust with only a few parameter updates, usually 1 or 2. The parameter update changes the KKT matrix in (6.5) and, if we use a direct linear system solver, we need to perform a new numerical factorization of the matrix. Since this operation can be computationally expensive, we perform the adaptation only when it is really necessary. In particular, we allow an update if the algorithm running time since the previous update is greater than a certain percentage of the factorization time (nominally 40 %) and if the new parameter is sufficiently different from the current, *e.g.* 5 times larger or smaller. Note that in case of an indirect method, this rule allows for more frequent changes of ρ since there is no need to re-factor the KKT matrix, and thus the parameter update is computationally much cheaper.

6.6 Solution polishing

Operator splitting methods are typically used for obtaining solutions of low or medium accuracy. However, we can often guess which constraints are active from an approximate primal-dual solution, and then obtain a high accuracy solution by solving an additional linear system.

Given a dual solution y of the problem, we define the sets of lower- and upper-active constraints

$$\begin{aligned}\mathcal{L} &\coloneqq \{i \in \{1, \ldots, m\} \,|\, y_i < 0\}\,,\\ \mathcal{U} &\coloneqq \{i \in \{1, \ldots, m\} \,|\, y_i > 0\}\,.\end{aligned}$$

According to (6.2d)–(6.2e) we have $z_{\mathcal{L}} = l_{\mathcal{L}}$ and $z_{\mathcal{U}} = u_{\mathcal{U}}$, where $l_{\mathcal{L}}$ denotes the vector composed of elements of l corresponding to the indices in $\mathcal{L}$. Similarly, we denote by $A_{\mathcal{L}}$ the matrix composed of rows of A corresponding to the indices in $\mathcal{L}$.

If the sets of active constraints are known *a priori*, then a primal-dual solution (x, y, z) can be found by solving the following linear system:

$$\begin{bmatrix} P & A_{\mathcal{L}}^T & A_{\mathcal{U}}^T \\ A_{\mathcal{L}} & & \\ A_{\mathcal{U}} & & \end{bmatrix} \begin{bmatrix} x \\ y_{\mathcal{L}} \\ y_{\mathcal{U}} \end{bmatrix} = \begin{bmatrix} -q \\ l_{\mathcal{L}} \\ u_{\mathcal{U}} \end{bmatrix}, \tag{6.10}$$

$$y_i = 0, \quad i \notin (\mathcal{L} \cup \mathcal{U}), \tag{6.11}$$

$$z = Ax. \tag{6.12}$$

We can then apply the aforementioned procedure to obtain a candidate solution (x, y, z). If the tuple (x, y, z) satisfies optimality conditions (6.2), then our guess is correct and (x, y) is a primal-dual solution of problem (6.1). This approach is referred to as *solution polishing*. Note that the dimension of the linear system in (6.10) is usually much smaller than the one in (6.5) because the number of active constraints at optimality is at most n for non-degenerate QPs.

However, the linear system (6.10) is not necessarily solvable even when the sets of active constraints $\mathcal{L}$ and $\mathcal{U}$ are correctly identified. This can happen, *e.g.* if the solution is degenerate, *i.e.* if it has one or more redundant active constraints. We can make the solution polishing procedure more robust by solving instead the following linear system:

$$\begin{bmatrix} P + \delta I & A_{\mathcal{L}}^T & A_{\mathcal{U}}^T \\ A_{\mathcal{L}} & -\delta I & \\ A_{\mathcal{U}} & & -\delta I \end{bmatrix} \begin{bmatrix} \hat{x} \\ \hat{y}_{\mathcal{L}} \\ \hat{y}_{\mathcal{U}} \end{bmatrix} = \begin{bmatrix} -q \\ l_{\mathcal{L}} \\ u_{\mathcal{U}} \end{bmatrix}, \tag{6.13}$$

where $\delta > 0$ is a regularization parameter with value typically around 10^{-6}. Since the regularized matrix in (6.13) is quasi-definite, the linear system is always solvable.

By using regularization, we actually solve a perturbed linear system and thus introduce a small error to the polished solution. If we denote by K and $(K + \Delta K)$ the matrices in (6.10) and (6.13), respectively, then we can represent the two linear systems as $Kt = g$ and $(K + \Delta K)\hat{t} = g$. To compensate for this error, we apply an *iterative refinement* procedure [76], *i.e.* we solve iteratively

$$\Delta\hat{t}^{k+1} \leftarrow (K + \Delta K)^{-1}\left(g - K\hat{t}^{k}\right) \tag{6.14a}$$

$$\hat{t}^{k+1} \leftarrow \hat{t}^{k} + \Delta\hat{t}^{k+1}. \tag{6.14b}$$

The sequence $\{\hat{t}^k\}_{k\in\mathbb{N}}$ converges to the true solution t, provided that it exists. Observe that the iterative refinement requires only forward and backward substitutions, and not another matrix factorization.

6.7 Parametric programs

In application areas such as control, statistics, finance, and SQP, problem (6.1) is solved repeatedly for varying data, and is thus referred to as a *parametric program*. In such cases we can exploit the problem structure and reduce the overall computation time if multiple instances of the problem are solved.

We make a distinction between two cases depending on which of the problem data are to be treated as parameters. We assume that the problem dimensions n and m, and the sparsity patterns of matrices P and A are fixed.

Vectors as parameters. If the vectors q, l and u are the only parameters of the problem, then the KKT matrix in Algorithm 6.1 does not change across different instances of the parametric program, provided that we do not update parameters σ and ρ. Thus, if the linear system in the algorithm is solved using a direct method, we perform factorization when solving the first instance of the problem, and then reuse it across all subsequent instances. Since the matrix factorization is a computationally intensive operation, we can reduce significantly the computation times to solve subsequent problem instances. This class of problems arises very frequently in many applications including linear model predictive control (MPC) and moving horizon estimation (MHE) [2, 145], lasso [48, 161], and portfolio optimization [44, 120].

Matrices and vectors as parameters. In the case in which the values (but not the locations) of nonzero entries of matrices P and A are parameters, we need to re-factor the KKT matrix in Algorithm 6.1 between the subsequent instances of the parametric program. However, since the sparsity pattern of the matrix does not change, we only need to perform the *numerical factorization* while reusing the *symbolic factorization* from the previous problem instance. This results in a modest reduction in the computation time. This class of problems encompasses several applications such as nonlinear MPC and MHE [68], and SQP [134].

Warm starting. In contrast to interior-point methods, we can provide to our algorithm an initial guess of both primal and dual solutions. When a series of similar optimization problems is solved, the solutions across problem instances are often similar. Since the algorithm's running time depends largely on the distance between the algorithm's initial iterate and the problem's solution set, one can set a solution of the previous problem instance as the initial iterate in the next instance. This strategy is known as *warm starting* and often improves running times of iterative optimization algorithms [103].

6.8 Code generation for embedded systems

In recent years, convex optimization has increasingly been applied on embedded systems where data are processed in real time and on low-cost computational platforms [46, §1], [148]. Current applications include, *e.g.* MPC [36], real-time signal processing [60, 123], and onboard trajectory planning in space missions [35, 155].

Real-time applications of embedded optimization impose special requirements on the solvers used [122]. First, embedded solvers must be reliable even in the presence of poor quality data, and should avoid exceptions caused by division by zero or memory faults caused by dynamic memory allocation. Second, the solver should be implementable on low-cost embedded platforms with very limited computational and memory resources. In particular, solvers should have very small compiled footprint, should consist only of basic algebraic operations, and should not be linked to any external libraries, which also makes the solver easily verifiable. Finally, real-time applications typically require that the solver is fast and able to correctly identify infeasible problems.

On the other hand, optimization problems arising in embedded applications have certain features that can be exploited when designing an embedded solver [122]. First, embedded optimization is typically applied to the repeated solution of parametrized problems in which the problem data, but not its dimensions or sparsity pattern, change between problem instances. For such problems, the solver initialization and some part of its computations can be performed offline during the solver design phase. Second, requirements on the solution accuracy in embedded applications are often moderate because of noise in the data and arbitrariness of the objective function. As an example, the authors in [169] show that acceptable control performance of an MPC controller is achievable even when using a very low accuracy solver. Finally, in embedded applications one can typically assume that problems are reasonably scaled.

6.8.1 Related work

In some cases the solution of a parametrized convex optimization problem can be precomputed offline using multi-parametric programming techniques [30, 162]. However, the memory required for storing such solutions grows exponentially with the problem dimensions, making this approach applicable only to small problems.

Over the last decade tools for generating custom online solvers for parametric problems have attracted increasing attention. CVXGEN [122] is a code generation software tool for small-scale parametric QPs. The generated solver is fast and reliable, but its main disadvantage is that the code size grows rapidly with the problem dimensions. This issue is overcome in FORCES [70, 72] where the size of the compiled code is broadly constant with respect to the problem dimensions. In HPMPC [86] tailored solvers for MPC are combined with high-performance optimized libraries for linear algebra. ECOS [50, 71] and Bsocp [75] are embedded solvers for a wider class of second-order cone programs (SOCPs). All of the aforementioned solvers are based on primal-dual interior point methods that are tailored for their specific problem classes. A known limitation of these methods is that they cannot use the warm starting technique, which is one of the dominant acceleration factors in applications such as MPC [103].

In contrast, qpOASES [83] is based on a parametric active-set method which can effectively use *a priori* information to speed-up computation of a QP solution. On the other hand, since qpOASES is based on dense linear algebra it cannot exploit sparsity in the problem data. Moreover, the computational complexity of active-set methods grows exponentially with the number of constraints.

FiOrdOs [164] uses first-order gradient methods as the basis for the embedded solvers it generates. In the case of a general QP, the methods require a Lipschitz constant of the gradient of the objective function in order to compute the step-size. Alternatively, FiOrdOs implements an adaptive rule for the step-size selection, but it requires a new matrix factorization each time the step-size is updated. QPgen [93, 94] uses optimal preconditioning of the problem data that can improve performance of first-order methods considerably. The main disadvantage of FiOrdOs and QPgen is their inability to detect infeasible problems.

6.9 OSQP

We have implemented our proposed approach in the "Operator Splitting Quadratic Program" (OSQP) solver, an open-source software package written in C language. OSQP solves QPs of the form (6.1), and makes no assumptions on the problem data other than convexity. OSQP is available online at

`http://osqp.readthedocs.io`.

Users can use OSQP from C, C++, Fortran, Python, Matlab, Julia and Rust, and via parsers such as CVXPY [1, 65], JuMP [77], and YALMIP [116].

To exploit sparsity in the data, OSQP stores matrices in Compressed-Sparse-Column (CSC) format [59]. OSQP supports multiple linear system solvers including Suitesparse LDL [3, 58] and MKL Pardiso [107].

The default values for the OSQP termination tolerance levels described in Section 6.3.4 are

$$\varepsilon_{\text{abs}} = \varepsilon_{\text{rel}} = 10^{-3}, \quad \varepsilon_{\text{pinf}} = \varepsilon_{\text{dinf}} = 10^{-4}.$$

The default step-size parameter σ and the relaxation parameter α are set to

$$\sigma = 10^{-6}, \quad \alpha = 1.6,$$

while ρ is automatically chosen by default as described in Section 6.5, with optional user override. For more details on the solver settings, we refer the reader to the solver documentation on the main website.

OSQP is able to generate tailored C code that compiles into a solver for a user-specified parametric QP. If the vectors q, l and u in problem (6.1) are the only parameters, then the KKT matrix in Algorithm 6.1 does not change across different instances of the parametric program. The matrix is then pre-factored offline and only backward and forward substitutions are performed during code execution. This enables a significant reduction of the code footprint. If the diagonal matrix D^{-1} is stored instead of D, then the resulting algorithm is division-free. Moreover, using warm starting makes the generated solvers extremely fast. For more details, we refer the reader to [13].

6.10 Numerical tests

We benchmarked OSQP against the open-source interior-point solver ECOS [71], the open-source active-set solver qpOASES [83], and the commercial interior-point solvers GUROBI [100] and MOSEK [126]. We executed the OSQP solver with default settings and solution polishing disabled.

Note that the solution returned by the other solvers is with high accuracy while OSQP returns a lower accuracy solution. Hence, runtime benchmarks are not completely fair since OSQP might take more time than interior point methods if high accuracy is required. On the other hand, we used the direct single-threaded linear system solver SuiteSparse LDL [3, 58] and very simple linear algebra, while other solvers such as GUROBI and MOSEK use advanced multi-threaded linear system solvers and custom linear algebra routines.

We say that a primal-dual solution $(x^\star, y^\star)$ returned by each solver is optimal if the following conditions are satisfied with $\varepsilon_{\text{abs}} = \varepsilon_{\text{rel}} = 10^{-3}$:

$$\|(Ax^\star - u)_+ + (Ax^\star - l)_-\|_\infty \leq \varepsilon_{\text{prim}}, \quad \|Px^\star + q + A^T y^\star\|_\infty \leq \varepsilon_{\text{dual}},$$
$$\left\| \min\left(y_+^\star, |u - Ax^\star|\right) \right\|_\infty \leq \varepsilon_{\text{slack}}, \quad \left\| \min\left(-y_-^\star, |Ax^\star - l|\right) \right\|_\infty \leq \varepsilon_{\text{slack}},$$

where $\varepsilon_{\text{prim}}$ and $\varepsilon_{\text{dual}}$ are defined in Section 6.3.4, and $\varepsilon_{\text{slack}} = \varepsilon_{\text{abs}} + \varepsilon_{\text{rel}}\|Ax^\star\|_\infty$. All the experiments were carried out on a system with 32 2.2 GHz cores and 512 GB of RAM, running Linux. The code for all numerical examples is available online at

`https://github.com/oxfordcontrol/osqp_benchmarks`.

Performance profiles. We make use of the performance profiles [69] to compare the timings of various solvers. We define the performance ratio as

$$r_{p,s} := \frac{t_{p,s}}{\min_s t_{p,s}},$$

where $t_{p,s}$ is the time it takes for solver s to solve problem instance p. If solver s fails to solve problem p, we set $r_{p,s} = \infty$. The performance profile plots the function $f_s : \mathbb{R} \mapsto [0, 1]$ defined as

$$f_s(\tau) := n_p^{-1} \sum_p \mathcal{I}_{\leq \tau}(r_{p,s}),$$

where $\mathcal{I}_{\leq \tau}(r_{p,s}) = 1$ if $r_{p,s} \leq \tau$, or 0 otherwise, and n_p is the number of problems considered. The value $f_s(\tau)$ corresponds to the fraction of problems solved within a factor τ of the best solver.

6.10.1 Benchmark problems

We considered QPs in the form (6.1) from 7 problem classes ranging from standard random problems to applications in the areas of control, finance and machine learning. For each problem class, we generated 10 different instances for 20 dimensions giving a total of 1400 problems. We describe data generation for each problem class in Appendix 6.A. All instances were obtained from realistic non-trivial random data. Throughout all the problem classes, n ranges between 10^1 and 10^4, m between 10^2 and 10^5, and the number of nonzeros in matrices P and A (that we denote by N) between 10^2 and 10^8.

Computation times. We show in Figure 6.1 the computation times across all the problem classes for OSQP and GUROBI. Each problem class is represented using a different marker. OSQP shows to be competitive or even faster than GUROBI for several problem classes.

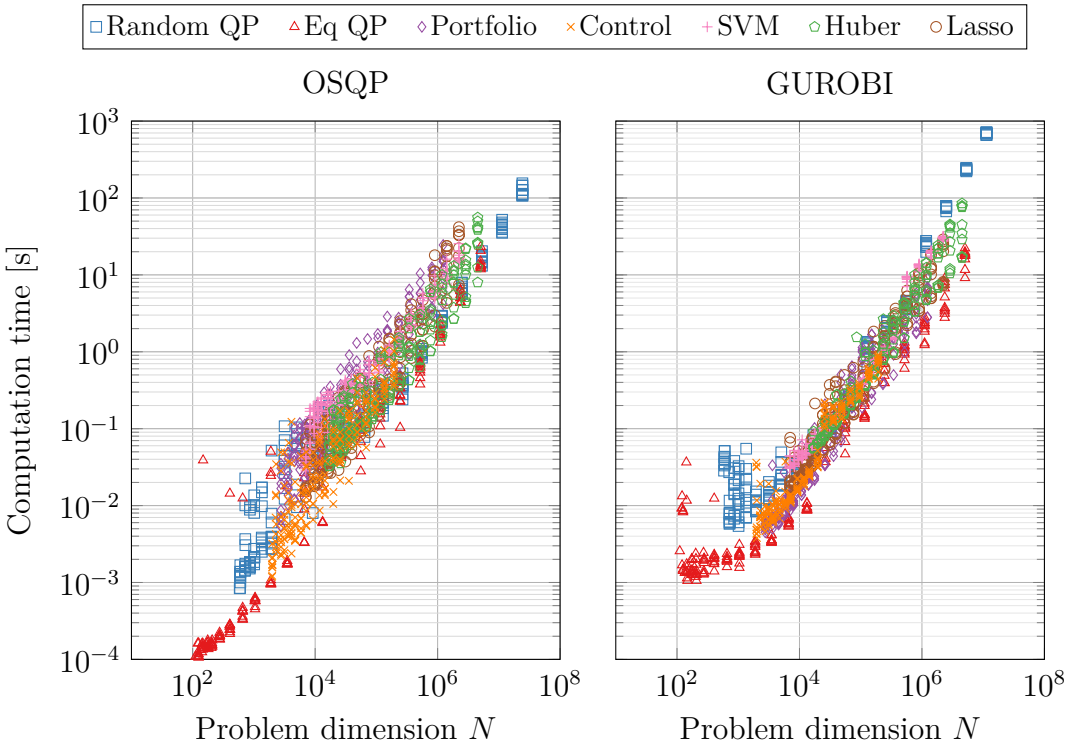


Figure 6.1: Computation times vs problem dimensions for OSQP (left) and GUROBI (right) for the 7 benchmark problem classes. N denotes the overall number of nonzero entries in matrices P and A.

Failure rates. Figure 6.2 shows the failure rates for all the solvers across the benchmark problems.

Performance profiles. Figure 6.3 compares the performance profiles of all the solvers tested. OSQP outperforms other solvers and has performance very close to the one of GUROBI. ECOS and MOSEK perform similarly even though ECOS is a single-threaded solver. qpOASES has a clear disadvantage compared to the other solvers when dealing with large problems because it cannot exploit sparsity in the problem matrices and the number of active set combinations becomes extremely large for large problem dimensions.

6.10.2 Polishing

We ran the same benchmark problems with the OSQP solver with solution polishing enabled. Polishing succeeded in 59 % of the times providing a high-accuracy solution with a median of 1.18× computation time compared to the OSQP solution with polishing disabled. When polishing succeeds, the solution is as accurate, or even more accurate, than the one obtained with any other

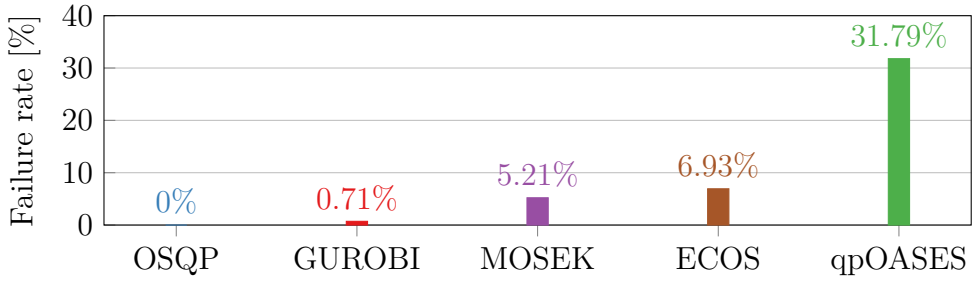


Figure 6.2: Failure rates for the 7 benchmark problem classes.

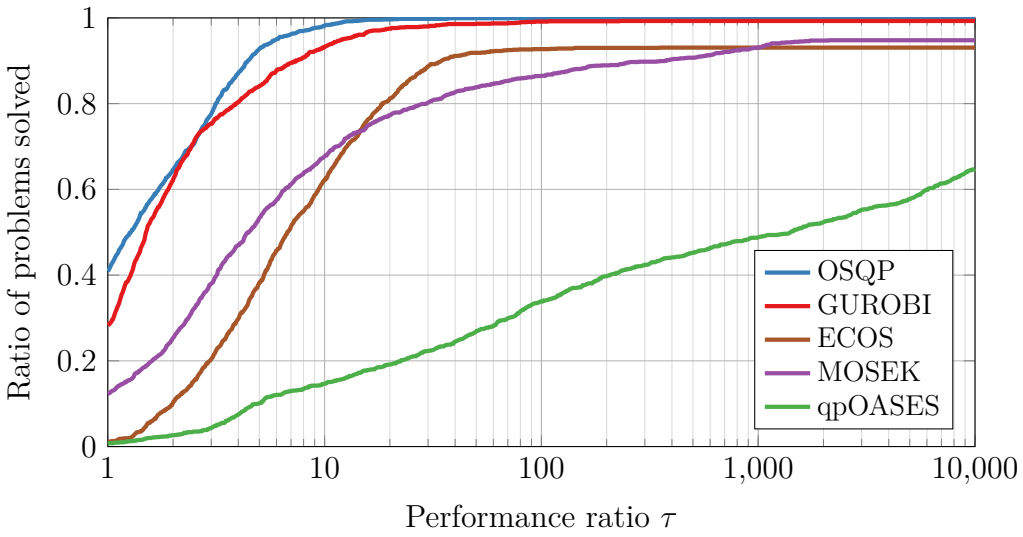


Figure 6.3: Performance profiles for the 7 benchmark problem classes.

solver. Note that by decreasing the tolerances ε_{abs} and ε_{rel} we can increase the percentage of times solution polishing succeeds.

6.10.3 Warm starting and factorization caching

To show the benefits of warm starting and factorization caching, we solved a sequence of QPs with the data varying according to some parameters.

Lasso. We solved a lasso problem described in Appendix 6.A.5 with varying parameter λ in order to obtain a series of regressors with different levels of sparsity. We solved one problem instance with $n = 50$ features, $m = 5000$ data points, and λ logarithmically spaced taking 100 values between $\lambda_{\max} = \|A^T b\|_\infty$ and $0.01\lambda_{\max}$. Since the parameter enters only in the linear part of the cost, we can reuse the matrix factorization and enable warm starting to reduce the computation time as discussed in Section 6.7.

Model predictive control. In MPC, we solve the optimal control problem described in Appendix 6.A.3 at each time step to compute an optimal input sequence over the prediction horizon. We apply only the first input to the system and propagate the state to the next time step. The whole procedure then repeats with an updated initial state x_{init}. We solved the control problem with $n_x = 10$ states, $n_u = 5$ inputs, horizon length $T = 10$, and 100 simulation steps. The initial state of the simulation is uniformly distributed and constrained to be within the feasible region, *i.e.* $x_{\text{init}} \sim \mathcal{U}(-0.5\bar{x}, 0.5\bar{x})$. Since the parameters only enter in the constraint bounds, we can reuse the matrix factorization and enable warm starting to reduce the overall computation time.

Portfolio back-test. We consider the portfolio optimization problem described in Appendix 6.A.4 with $n = 3000$ assets and $k = 100$ factors.

We run a 4 years back-test to compute the optimal assets investment depending on varying expected returns and factor models [42]. We solved 240 QPs per year giving a total of 960 QPs. Each month we solved 20 QPs corresponding to the trading days. Every day, we updated the expected returns μ by randomly generating another vector with $\mu_i \sim 0.9\hat{\mu}_i + \mathcal{N}(0, 0.1)$, where $\hat{\mu}_i$ comes from the previous expected returns. The risk model was updated every month by updating the nonzero entries of D and F according to $D_{ii} \sim 0.9\hat{D}_{ii} + \mathcal{U}[0, 0.1\sqrt{k}]$ and $F_{ij} \sim 0.9\hat{F}_{ij} + \mathcal{N}(0, 0.1)$ where $\hat{D}_{ii}$ and $\hat{F}_{ij}$ come from the previous risk model.

Since μ only enters in the linear part of the cost, we can reuse the matrix factorization and enable warm starting. Since the sparsity patterns of D and F do not change during the monthly updates, we can reuse the symbolic factorization and exploit warm starting to speed-up the computations.

Results. We show the results in Figure 6.4. For the lasso example, warm starting and factorization caching bring an average improvement in computation time of 8.5× going from 255.9 ms to 29.7 ms. In the MPC example, warm starting brings 2.8× improvement in average, from 22.8 ms to 8.0 ms. In the portfolio example, we obtain an average improvement of 6.2×, from 200.6 ms to 32.2 ms. Depending on the problem type and size, warm starting and factorization caching improve the performance considerably allowing a solution in few tens of milliseconds.

6.10.4 Maros-Mészáros problems

We consider the 138 problems from the Maros-Mészáros test set of hard QPs [121]. We compared the OSQP solver against GUROBI and MOSEK against all the problems in the set. We decided to exclude ECOS because it showed numerical issues for several problems in the test set. We also excluded qpOASES because it could not solve most of the problems.

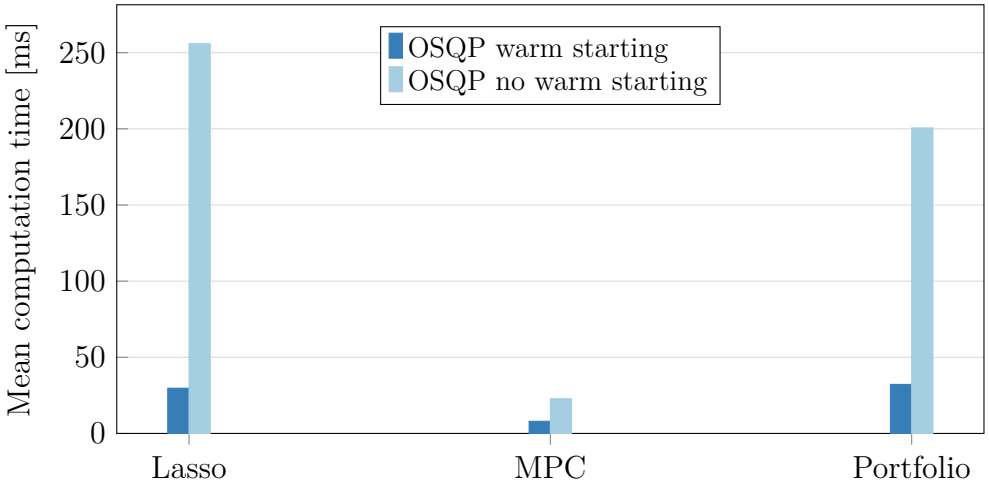


Figure 6.4: OSQP warm starting and factorization caching benchmarks.

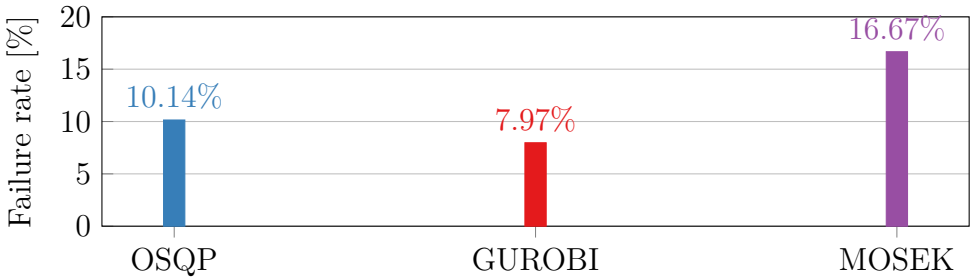


Figure 6.5: Failure rates for the Maros-Mészáros test set.

Failure rates and performance profiles. The failure rates are shown in Figure 6.5 and the performance profiles in Figure 6.6. OSQP shows competitive or even better performance than GUROBI in around 85 % of the cases. Its failure rate is 10.14 %. GUROBI solves the largest number of problems even though fails in 7.97 % of the cases. MOSEK is the slowest solver with the failure rate of 16.67 %.

6.11 Conclusions

We presented a general-purpose solver for QPs based on ADMM, employing a novel operator splitting technique that requires the solution of a quasi-definite linear system with the same coefficient matrix in each iteration. Our algorithm is very robust, placing no requirements on the problem data such as positive definiteness of the objective function or linear independence of the constraint functions. It is division-free once an initial matrix factorization is carried out, making it suitable for real-time applications in embedded systems. In addition, OSQP is the first operator splitting QP solver able to reliably detect primal

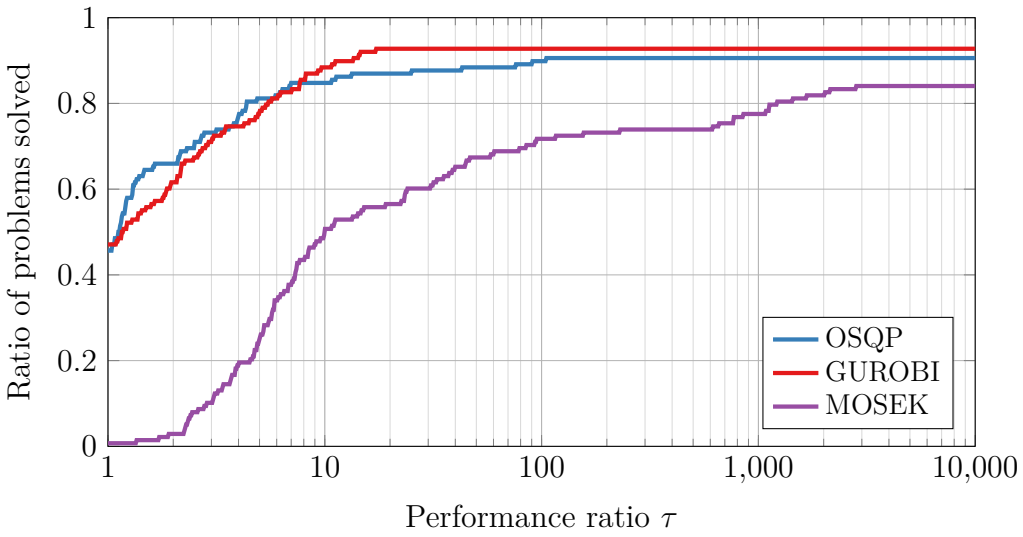


Figure 6.6: Performance profiles for Maros-Mészáros problems.

and dual infeasible problems. The method also supports factorization caching and warm starting, making it particularly efficient when solving parametrized problems arising in finance, control, and machine learning. Our open-source C implementation OSQP has a small footprint, can be compiled to be library-free, and has been extensively tested on many problem instances from a wide variety of application areas. Timing and failure rate tests show great improvements over state-of-the-art academic and commercial QP solvers.

6.A Benchmark problem classes

6.A.1 Random QP

Consider the following QP:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^T P x + q^T x \\ \text{subject to} & l \leq Ax \leq u. \end{array}$$

Problem instances. The number of variables and constraints in our problem instances are n and $m = 10n$. We generate a matrix $P = MM^T$ where $M \in \mathbb{R}^{n\times n}$ has 50 % nonzero entries generated as $M_{ij} \sim \mathcal{N}(0,1)$. Matrix $A \in \mathbb{R}^{m\times n}$ also has 50 % nonzero entries generated as $A_{ij} \sim \mathcal{N}(0,1)$. The linear part of the cost is normally distributed, *i.e.* $q_i \sim \mathcal{N}(0,1)$. We generate the constraint bounds as $l_i \sim \mathcal{U}[-1,0]$ and $u_i \sim \mathcal{U}[0,1]$.

6.A.2 Equality constrained QP

Consider the following equality constrained QP:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^T P x + q^T x \\ \text{subject to} & Ax = b. \end{array}$$

This problem can be rewritten in the form (6.1) by setting $l = u = b$.

Problem instances. The number of variables and constraints in our problem instances are n and $m = \lfloor n/2 \rfloor$. We generated random matrix $P = MM^T$ where $M \in \mathbb{R}^{n\times n}$ has 50 % nonzero entries generated as $M_{ij} \sim \mathcal{N}(0,1)$. Similarly, we form $A \in \mathbb{R}^{m\times n}$ with 50% nonzero entries generated as $A_{ij} \sim \mathcal{N}(0,1)$. The elements of vectors q and b are drawn from $\mathcal{N}(0,1)$.

Iterative refinement interpretation. Solution to the above problem can be found directly by solving the following linear system:

$$\begin{bmatrix} P & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x \\ \nu \end{bmatrix} = \begin{bmatrix} -q \\ b \end{bmatrix}. \tag{6.15}$$

If we set $\alpha = 1$, $y^0 = b$ and use Algorithm 6.1 to solve the problem, it boils down to the following iteration:

$$\begin{bmatrix} x^{k+1} \\ \nu^{k+1} \end{bmatrix} = \begin{bmatrix} x^k \\ \nu^k \end{bmatrix} + \begin{bmatrix} P + \sigma I & A^T \\ A & -\rho^{-1} I \end{bmatrix}^{-1} \left(\begin{bmatrix} -q \\ b \end{bmatrix} - \begin{bmatrix} P & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} x^k \\ \nu^k \end{bmatrix} \right),$$

which has the form (6.14a) with $g = (-q, b)$ and $\hat{t}^k = (x^k, \nu^k)$. This means that Algorithm 6.1 applied to an equality constrained QP is equivalent to performing the iterative refinement to the linear system in (6.15).

6.A.3 Optimal control

We consider the problem of controlling a constrained linear time-invariant dynamical system. To achieve this, we solve the following optimization problem [41]:

$$\begin{array}{ll} \text{minimize} & x_N^T Q_T x_T + \sum_{t=0}^{T-1} x_t^T Q x_t + u_t^T R u_t \\ \text{subject to} & x_{t+1} = Ax_t + Bu_t \\ & x_t \in X,\ u_t \in U \\ & x_0 = x_{\text{init}}. \end{array} \tag{6.16}$$

The states $x_t \in \mathbb{R}^{n_x}$ and the inputs $u_t \in \mathbb{R}^{n_u}$ are subject to polyhedral constraint sets $X \subseteq \mathbb{R}^{n_x}$ and $U \subseteq \mathbb{R}^{n_u}$, respectively. The prediction horizon is $T \in \mathbb{N}$, and the initial state is $x_{\text{init}} \in X$. Matrices $Q \in \mathbb{S}_+^{n_x}$ and $R \in \mathbb{S}_{++}^{n_u}$ define the state and input costs at each stage of the horizon, and $Q_T \in \mathbb{S}_+^{n_x}$ defines the final stage cost. By defining the new variable $z = (x_0, \ldots, x_T, u_0, \ldots, u_{T-1})$, problem (6.16) can be reformulated in the form (6.1) with a total of $(T+1)n_x + Tn_u$ variables.

Problem instances. We define linear time-invariant systems with n_x states and $n_u = \lfloor n_x/2 \rfloor$ inputs, and set the prediction horizon to $T = 10$. We generate the state matrix as $A = I + \Delta$ where the entries of Δ are generated as $\Delta_{ij} \sim \mathcal{N}(0, 0.01)$. We enforce the eigenvalues of A to be stable. Entries of the input matrix B are generated as $B_{ij} \sim \mathcal{N}(0, 1)$. The state cost matrix is diagonal with 70 % nonzero diagonal elements generated as $Q_{ii} \sim \mathcal{U}[0, 10]$. We chose the input cost matrix as $R = 0.1I$. The terminal cost matrix Q_T is chosen by solving a discrete algebraic Riccati equation [41]. We generated state and input constraints as

$$X = \{x_t \in \mathbb{R}^{n_x} \mid -\bar{x} \leq x_t \leq \bar{x}\}, \qquad U = \{u_t \in \mathbb{R}^{n_u} \mid -\bar{u} \leq u_t \leq \bar{u}\},$$

where $\bar{x}_i \sim \mathcal{U}[1, 2]$ and $\bar{u}_i \sim \mathcal{U}[0, 0.1]$. The initial state is uniformly distributed with $x_{\text{init}} \sim \mathcal{U}[-0.5\bar{x}, 0.5\bar{x}]$.

6.A.4 Portfolio optimization

Portfolio optimization is a problem arising in finance that seeks to allocate assets in a way that maximizes risk adjusted return [42, 44, 120], [46, §4.4.1],

$$\begin{array}{ll} \text{maximize} & \mu^T x - \gamma(x^T \Sigma x) \\ \text{subject to} & \mathbf{1}^T x = 1, \quad x \geq 0, \end{array}$$

where the variable $x \in \mathbb{R}^n$ represents the portfolio, $\mu \in \mathbb{R}^n$ the vector of expected returns, $\gamma > 0$ the risk aversion parameter, and $\Sigma \in \mathbb{S}_+^n$ the asset return covariance.

A common assumption is the k-factor risk model [54], where the return covariance matrix is the sum of a diagonal and a matrix of rank $k < n$, *i.e.*

$$\Sigma = D + FF^T,$$

where $F \in \mathbb{R}^{n\times k}$ is the factor loading matrix, and $D \in \mathbb{S}^n_+$ is a diagonal matrix describing the asset-specific risk.

We can reformulate the problem above in the form (6.1) with linear part of the cost depending on parameter γ,

$$\begin{array}{ll} \text{minimize} & x^T D x + y^T y - \gamma^{-1}\mu^T x \\ \text{subject to} & \mathbf{1}^T x = 1, \quad x \geq 0 \\ & y = F^T x \end{array} \tag{6.17}$$

Problem instances. We generate instances of portfolio optimization problem for increasing number of factors k and number of assets $n = 100k$. The matrix F has 50 % nonzero entries generated as $F_{ij} \sim \mathcal{N}(0,1)$, and diagonal elements of D are generated as $D_{ii} \sim \mathcal{U}[0, \sqrt{k}]$. The expected return vector is generated as $\mu_i \sim \mathcal{N}(0,1)$. We set $\gamma = 1$.

6.A.5 Lasso

We seek a sparse solution of the regressor selection problem, which is in general a hard combinatorial problem [46, §6.3]. The *least absolute shrinkage and selection operator (lasso)* is a popular heuristic for promoting sparsity of the solution by adding an ℓ_1–regularization term in the objective [48, 161]. The problem is

$$\text{minimize} \quad \|Ax - b\|_2^2 + \lambda\|x\|_1,$$

where $x \in \mathbb{R}^n$ is the vector of parameters, $A \in \mathbb{R}^{m\times n}$ is the data matrix whose columns are potential regressors, $b \in \mathbb{R}^m$ is the vector of measurements that is to be fit by a subset of regressors, and $\lambda > 0$ is the weighting parameter.

We reformulate the problem above as the following QP:

$$\begin{array}{ll} \text{minimize} & y^T y + \lambda \mathbf{1}^T t \\ \text{subject to} & y = Ax - b \\ & -t \leq x \leq t. \end{array}$$

Problem instances. Matrix A has 50 % nonzero entries generated as $A_{ij} \sim \mathcal{N}(0,1)$. Vector $\hat{x}$ also has 50 % nonzero elements drawn from $\mathcal{N}(0, 1/n)$. We then set $b = A\hat{x} + \varepsilon$, where ε represents a vector of noise with elements drawn from $\mathcal{N}(0,1)$. We generate the problem instances with varying number of parameters n and $m = 100n$ data points. The parameter λ is chosen as $\|A^T b\|_\infty/5$.

6.A.6 Huber fitting

Huber fitting or *robust least-squares* performs linear regression under the assumption that there are outliers in the data [105, 106]. The problem is

$$\text{minimize} \quad \sum_{i=1}^{m} \phi_{\text{hub}}(a_i^T x - b_i), \tag{6.18}$$

with the Huber penalty function $\phi_{\text{hub}} : \mathbb{R} \mapsto \mathbb{R}_+$ defined as

$$\phi_{\text{hub}}(u) := \begin{cases} u^2 & |u| \leq M \\ M(2|u| - M) & |u| > M. \end{cases}$$

Problem (6.18) is equivalent to the following QP [46, p.190]:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}u^T u + M\mathbf{1}^T v \\ \text{subject to} & -u - v \leq Ax - b \leq u + v \\ & 0 \leq u \leq M\mathbf{1} \\ & v \geq 0. \end{array}$$

Problem instances. Matrix A has 50 % nonzero entries generated as $A_{ij} \sim \mathcal{N}(0, 1)$. To construct $b \in \mathbb{R}^m$ we first generate a vector $v \in \mathbb{R}^n$ with $v_i \sim \mathcal{N}(0, 1/n)$, and a noise vector $\varepsilon \in \mathbb{R}^m$ with

$$\varepsilon_i \sim \begin{cases} \mathcal{N}(0, 1/4) & \text{with probability } p = 0.95 \\ \mathcal{U}[0, 10] & \text{otherwise.} \end{cases}$$

We then set $b = Av + \varepsilon$. For each problem instance we choose $m = 10n$ and $M = 1$.

6.A.7 Support vector machine

Support vector machine problem seeks a hyperplane that approximately separates two sets of points in the feature space [56]. The problem can be formulated as

$$\text{minimize} \quad x^T x + \lambda \sum_{i=1}^{m} \max(0, b_i a_i^T x + 1),$$

where $b_i \in \{-1, +1\}$ is a set label, a_i is a vector of features for the i-th point, and $\lambda > 0$ is the weighting parameter. The problem can be reformulated as the following QP:

$$\begin{array}{ll} \text{minimize} & x^T x + \lambda \mathbf{1}^T t \\ \text{subject to} & t \geq \text{diag}(b)Ax + \mathbf{1} \\ & t \geq 0, \end{array}$$

where $\text{diag}(b)$ denotes the diagonal matrix with elements of b on its diagonal.

Problem instances. We choose the vector b so that

$$b_i = \begin{cases} +1 & i \leq m/2 \\ -1 & \text{otherwise.} \end{cases}$$

Matrix A has 50 % nonzero entries generated as

$$A_{ij} \sim \begin{cases} \mathcal{N}(+1/n, 1/n) & i \leq m/2 \\ \mathcal{N}(-1/n, 1/n) & \text{otherwise.} \end{cases}$$

We set $\lambda = 1$.

7

Convex Problems with Cardinality Constraints

Contents

In the previous chapters, we considered solving convex optimization problems using operator splitting methods. Properties of the related fixed-point operators allowed us to use powerful operator theory to analyze asymptotic behavior of these methods. Although many of these properties do not hold when considering nonconvex optimization problems, operator splitting methods can be powerful in tackling such problems as well.

7.1 Introduction

In this chapter we consider the following optimization problem:

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & \operatorname{card}(x) \leq k, \quad x \in \mathcal{B}, \end{array} \tag{7.1}$$

where $f : \mathbb{R}^n \mapsto \tilde{\mathbb{R}}$ is a differentiable convex function, $k \in \mathbb{N}$ is a given positive integer, and $\mathcal{B} = [l\mathbf{1}, u\mathbf{1}]$ with $l \leq 0$ and $u \geq 0$. Such problems arise in a number of engineering applications, such as compression [73], sparse regression [125], image processing [119], and sparse controller design [114].

The cardinality constraint in (7.1) is used to impose sparsity of some degree on the optimal solution. However, such a constraint is nonconvex and the resulting optimization problem is known to be $\mathcal{NP}$-hard in general. A widely used heuristic approach for enforcing sparsity of the solution is to add an ℓ_1–regularization term in the objective [161], *i.e.* to solve the following convex relaxation of problem (7.1):

$$\begin{array}{ll} \text{minimize} & f(x) + \gamma \|x\|_1 \\ \text{subject to} & x \in \mathcal{B}, \end{array} \tag{7.2}$$

for different values of the weighting parameter $\gamma \geq 0$, and to find a γ sufficiently large to obtain a solution in which the original cardinality constraint is satisfied. However, any such solution to (7.2) is not guaranteed to be optimal for the original problem (7.1).

This approach to obtaining a sparse solution has received significant attention, because the resulting convex problem can be solved very efficiently. Since the problem (7.2) has special structure, operator splitting methods can be used to obtain a solution when the dimension of the optimization variable is very large [45]. A related approach is to replace the ℓ_1–norm in (7.2) by a weighted ℓ_1–norm, and to then solve a sequence of weighted ℓ_1 optimization problems wherein the weights are updated based on previously obtained solutions [48].

However, the quality of a solution obtained by relaxing the nonconvex constraint is not always satisfactory. One alternative approach is to tackle the nonconvex problem (7.1) directly. The exact minimization of such a nonconvex problem is possible using techniques such as branch-and-bound, but the approach is only applicable to problems of relatively low dimension. The authors in [24, 37] proposed a projected gradient method for solving a variant of problem (7.1) that does not consider the constraint $x \in \mathcal{B}$, and established convergence of the algorithm to a stationary point under some regularity assumptions on the objective function. The authors in [67] use the alternating direction method of multipliers (ADMM) to solve problem (7.1), with additional heuristic steps that usually improve the quality of the solution obtained. However, this method is not guaranteed to converge and the authors propose simply to terminate the algorithm after some predefined number of iterations.

In this chapter we reformulate problem (7.1) by replacing the ℓ_1–regularization term in (7.2) by a function that penalizes only the $(n - k)$ smallest elements in magnitude of x, rather than all the elements of x. We then apply the proximal gradient method (PGM) to the reformulated problem, and show that under some regularity assumptions on the objective function our algorithm converges to a stationary point. We show that, given a suitable selection of the weighting parameter, our algorithm is equivalent to the projected gradient method for solving problem (7.1). However, we propose an update scheme of the weighting parameter which often yields a better solution than when the parameter is held fixed.

7.2 Problem reformulation

We make the following standing assumptions about the cardinality constrained optimization problem (7.1):

Assumption 7.1.

(i) The function f is convex and L_f–Lipschitz smooth.

(ii) The function f is lower-bounded on $\mathcal{B}$, i.e. *there exists some $\zeta \in \mathbb{R}$ such that $f(x) \geq \zeta$ for all $x \in \mathcal{B}$.*

Notice that, given Assumption 7.1(i), a sufficient condition for f to be lower-bounded on $\mathcal{B}$ is that $\mathcal{B}$ is bounded.

In this section we will show how to reformulate problem (7.1) in a form suitable for applying operator splitting methods. Observe that the cardinality constraint can be equivalently written as $\|x\|_1 = \|x\|_{[k]}$ [102]. Since $\|x\|_1 \geq \|x\|_{[k]}$ is always true, the cardinality constraint in (7.1) can be replaced by $\|x\|_1 - \|x\|_{[k]} \leq 0$. By representing the resulting problem in the Lagrangian form, we obtain

$$\begin{array}{ll} \text{minimize} & f(x) + \gamma\, \varphi_{[k]}(x) \\ \text{subject to} & x \in \mathcal{B}, \end{array} \tag{7.3}$$

where $\varphi_{[k]}(x) \coloneqq \|x\|_1 - \|x\|_{[k]}$, and $\gamma \geq 0$ is a weighting parameter. Observe that, compared to problem (7.2), in problem (7.3) only the $(n-k)$ smallest elements in magnitude of x are penalized, rather than all the elements of x.

We show in the sequel that, for an appropriately selected parameter γ, stationary points of problems (7.1) and (7.3) coincide. By introducing the following functions:

$$\begin{aligned} h_{\mathcal{P}}(x) &\coloneqq f(x) + \mathcal{I}_{\mathcal{B}}(x) + \mathcal{I}_{\mathrm{card}\leq k}(x), \\ h_{\mathcal{R}}(x) &\coloneqq f(x) + \mathcal{I}_{\mathcal{B}}(x) + \gamma\, \varphi_{[k]}(x), \end{aligned}$$

where $\mathcal{I}_{\mathrm{card}\leq k}$ is the indicator function of the set $\{x \in \mathbb{R}^n \mid \mathrm{card}(x) \leq k\}$, we can characterize stationary points of problems (7.1) and (7.3) in a way that they are equivalent to stationary points of functions $h_{\mathcal{P}}$ and $h_{\mathcal{R}}$, respectively.

Theorem 7.2. *If $x^\star \in \mathcal{B}$ is a stationary point of problem* (7.3) *with $\gamma > \|\nabla f(x^\star)\|_\infty$, then $x^\star$ is a stationary point of problem* (7.1)*, and* vice versa.

Proof. 1) Suppose that $x^\star$ is a stationary point of problem (7.1), and thus $\varphi_{[k]}(x^\star) = 0$. We assume that $x^\star \neq 0$. Let $\mathcal{J}$ be the set of indices i for which $x_i^\star \neq 0$, and $d \in \mathbb{R}^n$ any vector such that $\|d\|_\infty < \min_{i\in\mathcal{J}} |x_i^\star|/2 =: \varepsilon$. Then the set of indices of the k largest elements in magnitude of $(x^\star + d)$ is equivalent to

$\mathcal{J}$, and thus $\varphi_{[k]}(x^\star + d) = \sum_{j\notin\mathcal{J}} |d_j|$. Due to the convexity of f and $\mathcal{I}_\mathcal{B}$, and separability of $\mathcal{I}_\mathcal{B}$, we have

$$\begin{aligned} h_\mathcal{R}(x^\star + d) - h_\mathcal{R}(x^\star) &\geq \langle \nabla f(x^\star), d\rangle + \mathcal{I}'_\mathcal{B}(x^\star, d) + \gamma \sum_{j\notin\mathcal{J}} |d_j| \\ &= \sum_{i\in\mathcal{J}} \Big(\nabla_i f(x^\star)\, d_i + \mathcal{I}'_\mathcal{B}(x^\star_i, d_i)\Big) + \\ &\quad \sum_{j\notin\mathcal{J}} \Big(\nabla_j f(x^\star)\, d_j + \mathcal{I}'_\mathcal{B}(x^\star_j, d_j)\Big) + \gamma \sum_{j\notin\mathcal{J}} |d_j|. \end{aligned}$$

Observe that $\mathcal{I}'_\mathcal{B}(x^\star_j, d_j)$ is 0 if $(x^\star_j + d_j) \in \mathcal{B}$, and ∞ otherwise. In particular, we always have $\mathcal{I}'_\mathcal{B}(x^\star_j, d_j) \geq 0$. On the other hand, since $x^\star$ is a stationary point of problem (7.1), this means that moving x from $x^\star$ along coordinates $i \in \mathcal{J}$ either leads to infeasibility or does not improve f, *i.e.* the first sum in the above equality is nonnegative. Therefore, the above inequality reduces to

$$\begin{aligned} h_\mathcal{R}(x^\star + d) - h_\mathcal{R}(x^\star) &\geq \sum_{j\notin\mathcal{J}} \nabla_j f(x^\star)\, d_j + \gamma \sum_{j\notin\mathcal{J}} |d_j| \\ &\geq -\max_{j\notin\mathcal{J}} |\nabla_j f(x^\star)| \sum_{j\notin\mathcal{J}} |d_j| + \gamma \sum_{j\notin\mathcal{J}} |d_j| \\ &= \Big(\gamma - \max_{j\notin\mathcal{J}} |\nabla_j f(x^\star)|\Big) \sum_{j\notin\mathcal{J}} |d_j|. \end{aligned}$$

Provided that $\gamma > \|\nabla f(x^\star)\|_\infty \geq \max_{j\notin\mathcal{J}} |\nabla f_j(x^\star)|$, the above inequality implies

$$h_\mathcal{R}(x^\star + d) \geq h_\mathcal{R}(x^\star), \tag{7.4}$$

for all $d \in \mathbb{R}^n$ such that $\|d\|_\infty < \varepsilon$, *i.e.* $x^\star$ is a local minimizer of (7.3) [33], and thus its stationary point.

If $x^\star = 0$, then the cardinality constraint is not active at the solution implying that 0 is a global minimizer of f over $\mathcal{B}$. Therefore, for all $x \in \mathcal{B}$,

$$h_\mathcal{R}(0) = f(0) \leq f(x) \leq f(x) + \mathcal{I}_\mathcal{B}(x) + \varphi_{[k]}(x) = h_\mathcal{R}(x),$$

that is, 0 is a global minimizer of $h_\mathcal{R}$. This concludes the first direction of the proof.

2) Suppose that $x^\star$ is a stationary point of problem (7.3) with $\gamma > \|\nabla f_i(x^\star)\|_\infty$. We first show that this implies $\varphi_{[k]}(x^\star) = 0$. Assume that $\varphi_{[k]}(x^\star) \neq 0$ which means that the $(k+1)^{\text{th}}$ largest element in magnitude of $x^\star$, whose index is denoted by i, is not equal zero. Then the directional derivative of $h_\mathcal{R}$ at $x^\star$ in the direction $d = (0, \cdots, 0, -x^\star_i, 0, \cdots, 0)$ is

$$\begin{aligned} h'_\mathcal{R}(x^\star, d) &= \langle \nabla f(x^\star), d\rangle + \mathcal{I}'_\mathcal{B}(x^\star, d) + \gamma\, \varphi'_{[k]}(x^\star, d) \\ &= -\nabla_i f(x^\star)\, x^\star_i - \gamma\, |x^\star_i| \\ &\leq (|\nabla f_i(x^\star)| - \gamma)\, |x^\star_i| \\ &< 0, \end{aligned}$$

where we used the fact that $\mathcal{I}'_{\mathcal{B}}(x_i^\star, -x_i^\star) = 0$ coming from $0 \in \mathcal{B}$, $\gamma > |\nabla f_i(x^\star)|$, and $|x_i^\star| > 0$. In the sense of Definition 2.19, the above inequality means that $x^\star$ is not a stationary point of (7.3), which is a contradiction.x

Observe that if $d \in \mathbb{R}^n$ points in a feasible direction from $x^\star$ with respect to the cardinality constraints, then $\mathcal{I}'_{\text{card}\leq k}(x^\star, d) = \gamma\, \varphi'_{[k]}(x^\star, d) = 0$, and otherwise $\mathcal{I}'_{\text{card}\leq k}(x^\star, d) = \infty$ and $\gamma\, \varphi'_{[k]}(x^\star, d) \leq \gamma\, \|d\|_{[k]} < \infty$. We now have $\mathcal{I}'_{\text{card}\leq k}(x^\star, d) \geq \gamma\, \varphi'_{[k]}(x^\star, d)$ for all d, and therefore

$$\begin{aligned} h'_{\mathcal{P}}(x^\star, d) &= \langle \nabla f(x^\star), d\rangle + \mathcal{I}'_{\mathcal{B}}(x^\star, d) + \mathcal{I}'_{\text{card}\leq k}(x^\star, d) \\ &\geq \langle \nabla f(x^\star), d\rangle + \mathcal{I}'_{\mathcal{B}}(x^\star, d) + \gamma\, \varphi'_{[k]}(x^\star, d) \\ &= h'_{\mathcal{R}}(x^\star, d) \\ &\geq 0, \end{aligned}$$

The above inequality means that $x^\star$ is a stationary point of problem (7.1). This concludes the proof. □

7.3 Solution method

Operator splitting methods were originally designed for solving optimization problems in the form

$$\text{minimize} \quad f(x) + g(x), \tag{7.5}$$

where both f and g are convex. However, they are sometimes used as heuristics in nonconvex optimization [24, 45, 67, 114]. The advantage of these methods is that the functions f and g can be tackled separately. For instance, in PGM the function f is tackled through its gradient, and g through its proximal operator. In the case when both f and g are convex, and ∇f is Lipschitz continuous, the method converges to a global minimum [17]. Another method that can be used for solving problem (7.5) is ADMM, which uses the proximal operators of both f and g. This method does not require any differentiability assumptions on the functions and, provided that both f and g are convex, and that problem (7.5) is solvable, the method always converges to a global minimum [17].

If we set

$$g(x; \gamma, l, u) \coloneqq \mathcal{I}_{\mathcal{B}}(x) + \gamma\, \varphi_{[k]}(x), \tag{7.6}$$

where (γ, l, u) are the function parameters, then problem (7.3) can be represented in the form (7.5). In order to use proximal methods for solving problem (7.3) efficiently, the proximal operator of the function g should be easy to evaluate. We will next show how to evaluate the proximal operator of g, and that PGM can be used for computing a stationary point of problem (7.3).

Note that the authors in [97] use the same reformulation to tackle problem (7.1), but solve it using a different approach. In particular, the authors represent the objective in problem (7.3) as a difference of convex functions (*i.e.* a *DC function)* and apply so-called DC algorithms to the problem.

7.3.1 Evaluating the proximal operator

The authors in [24] show that projection of $x \in \mathbb{R}^n$ onto the subset of $\mathbb{R}^n$ with cardinality at most k can be obtained by setting its $(n-k)$ smallest elements in magnitude to zero. Since the choice of these elements does not have to be unique, neither is the projection on the cardinality constraint. This corresponds to a projection onto the constraint set in (7.1) when setting $l = -\infty$ and $u = \infty$. In the case that $-l = u = M > 0$, the projection can be obtained by additional clipping of the elements with magnitude larger than M, as shown in [67]. It is easy to show that in the case when $l = 0$ and $u = M > 0$, the projection can be obtained by first setting all the nonnegative elements to zero, and then performing projection as in the case when $-l = u = M > 0$.

Similarly, we can evaluate the proximal operator of function g given in (7.6). We denote by $\{i_s(x)\}_{s=1}^n$ a permutation of indices of $x \in \mathbb{R}^n$ such that

$$|x_{i_1(x)}| \geq \cdots \geq |x_{i_n(x)}|.$$

The proximal operator of $g(x; \gamma, -M, M)$ has the form

$$\left(\operatorname{prox}_g(x; \gamma, -M, M)\right)_{i_s(x)} = \begin{cases} \operatorname{sat}_{[-M,M]}\left(x_{i_s(x)}\right) & s \leq k \\ \operatorname{sat}_{[-M,M]}\left(\operatorname{sth}_\gamma(x_{i_s(x)})\right) & s > k. \end{cases}$$

and the proximal operator of $g(x; \gamma, 0, M)$ is

$$\left(\operatorname{prox}_g(x; \gamma, 0, M)\right)_{i_s(x_+)} = \begin{cases} \operatorname{sat}_{[0,M]}\left(x_{i_s(x_+)}\right) & s \leq k \\ \operatorname{sat}_{[0,M]}\left(\operatorname{sth}_\gamma(x_{i_s(x_+)})\right) & s > k \end{cases}$$

In order to evaluate the proximal operators of $g(x; \gamma, -M, M)$ and $g(x; \gamma, 0, M)$, we must first sort in magnitude the elements of x and x_+, respectively. Observe that when $\gamma = \infty$, the proximal operator of g is equivalent to projection onto the constraint set in (7.1). This implies that PGM for solving problem (7.3) generalizes the projected gradient method for solving problem (7.1) for which the convergence to a stationary point was established in [24]. We will next show that we can extend this result to any $\gamma \geq 0$ in PGM.

7.3.2 Convergence of PGM

PGM described in Algorithm 7.1 is known to converge when f and g are convex, f is L_f–Lipschitz smooth, and $L > L_f/2$ [130, 154]. The authors in [24] establish convergence of PGM when g is an indicator function of a general closed set, provided that $L > L_f$. We will use similar arguments to establish convergence of PGM for an arbitrary nonconvex function g. We first require a supporting lemma:

Algorithm 7.1 PGM for problem (7.5).

1: **given** initial value $x_0 \in \mathcal{B}$ and parameter $L > 0$
2: **repeat**
3: $\quad y^{t+1} \leftarrow x^t - L^{-1}\nabla f(x^t)$
4: $\quad x^{t+1} \leftarrow \operatorname{prox}_{g/L}\left(y^{t+1}\right)$
5: $\quad k \leftarrow k + 1$
6: **until** convergence

Lemma 7.3. *Any fixed-point $x^\star$ of Algorithm 7.1,* i.e. *a point satisfying*

$$x^\star \in \operatorname{prox}_{g/L}\left(x^\star - L^{-1}\nabla f(x^\star)\right),$$

is a stationary point of problem (7.5).

Proof. From the definition of a proximal operator, $x^\star$ must satisfy

$$x^\star \in \operatorname*{argmin}_z \left\{ g(z) + \tfrac{L}{2}\left\| z - \left(x^\star - L^{-1}\nabla f(x^\star)\right)\right\|^2 \right\}. \tag{7.7}$$

Since every minimizer is a stationary point, the above inclusion implies

$$g'(x^\star, d) + \langle \nabla f(x^\star), d\rangle \geq 0, \quad \forall d \in \mathbb{R}^n,$$

which means that $x^\star$ is also a stationary point of problem (7.5). □

Theorem 7.4. *Let $\{x^t\}_{t\in\mathbb{N}}$ be a sequence of iterates generated by Algorithm 7.1 for solving problem* (7.5). *Suppose that $(f + g)$ is lower-bounded, f is convex and L_f–Lipschitz smooth, and $L > L_f$. Then the sequence $\{f(x^t)+g(x^t)\}_{t\in\mathbb{N}}$ converges and any accumulation point of $\{x^t\}_{t\in\mathbb{N}}$ is a stationary point of problem* (7.5).

Proof. Let $h(x) \coloneqq f(x) + g(x)$ and

$$h_L(z; x) \coloneqq f(x) + \langle \nabla f(x), z - x\rangle + \tfrac{L}{2}\|z - x\|^2 + g(z). \tag{7.8}$$

We first show that Algorithm 7.1 generates a non-increasing sequence $\{h(x^t)\}_{t\in\mathbb{N}}$. Lipschitz smoothness of f implies that $h_{L_f}(z; x)$ is an upper bound on $h(z)$ [130], *i.e.* for all $x \in \mathbb{R}^n$ and $z \in \mathbb{R}^n$,

$$h_{L_f}(z; x) \geq h(z). \tag{7.9}$$

Similar to (7.7), x^{t+1} can be characterized as

$$x^{t+1} \in \operatorname*{argmin}_z \left\{ g(z) + \tfrac{L}{2}\left\| z - \left(x^t - L^{-1}\nabla f(x^t)\right)\right\|^2 \right\}. \tag{7.10}$$

Setting $x = x^t$ in (7.8) and rearranging the terms, we get

$$h_L(z; x^t) = g(z) + \tfrac{L}{2}\left\| z - \left(x^t - L^{-1}\nabla f(x^t)\right)\right\|^2 + f(x^t) - (2L)^{-1}\|\nabla f(x^t)\|^2.$$

Since the last two summands in the above equality do not depend on z, inclusion (7.10) is equivalent to

$$x^{t+1} \in \operatorname*{argmin}_z h_L(z; x^t),$$

which implies

$$h_L(x^{t+1}; x^t) \leq h_L(x^t; x^t) = h(x^t). \tag{7.11}$$

Setting $x = x^t$ and $z = x^{t+1}$ in (7.9), and combining with (7.11), we obtain

$$\begin{aligned} h(x^t) - h(x^{t+1}) &\geq h_L(x^{t+1}; x^t) - h_{L_f}(x^{t+1}; x^t) \\ &= \frac{L - L_f}{2} \|x^{t+1} - x^t\|^2. \end{aligned} \tag{7.12}$$

The last inequality shows that $\{h(x^t)\}_{t\in\mathbb{N}}$ is a strictly decreasing sequence but for the case when $x^{t+1} = x^t$, for which x^t is a fixed-point of Algorithm 7.1 and, according to Lemma 7.3, a stationary point of problem (7.5). Lower-boundedness of h together with monotonicity of $\{h(x^t)\}_{t\in\mathbb{N}}$ implies convergence by the *monotone convergence theorem.* □

Corollary 7.5. *Suppose that Assumption 7.1 holds, $L > L_f$, and let $\{x^t\}_{t\in\mathbb{N}}$ be a sequence of iterates generated by Algorithm 7.1 for solving problem* (7.3)*. Then the sequence $\{h_{\mathcal{R}}(x^t)\}_{t\in\mathbb{N}}$ converges and any accumulation point of $\{x^t\}_{t\in\mathbb{N}}$ is a stationary point of problem* (7.3)*.*

7.3.3 Termination criterion

As shown in the previous subsection, the Algorithm 7.1 for solving problem (7.3) always produces a monotonically decreasing sequence $\{h_{\mathcal{R}}(x^t)\}_{t\in\mathbb{N}}$ that eventually converges to some value $h^\star_{\mathcal{R}}$. Since we do not know $h^\star_{\mathcal{R}}$, a reasonable termination criterion is that the difference in objective value in subsequent iterations is small relative to the objective value, *i.e.*

$$h_{\mathcal{R}}(x^t) - h_{\mathcal{R}}(x^{t+1}) \leq \varepsilon \, |h_{\mathcal{R}}(x^{t+1})|,$$

where $\varepsilon \in \mathbb{R}_+$ is the optimality tolerance. Notice from (7.12) that this condition implies that $\|x^{t+1} - x^t\|$ is also small.

7.3.4 Heuristic for updating the weighting parameter

In Section 7.2 we showed that stationary points of problems (7.1) and (7.3) coincide provided that $\gamma > \|\nabla f(x^\star)\|_\infty$. Although $\nabla f(x^\star)$ is not known prior to the algorithm runtime, in some cases we can find an upper bound on $\|\nabla f(x^\star)\|_\infty$ over $\mathcal{B}$, and use it in order to select an appropriate γ. However, such a selection rule usually results in a relatively large value of γ, and consequently in equivalence between our proposed method and the projected gradient method for solving problem (7.1).

We thus propose a heuristic for updating the weighting parameter γ at each iteration. After y^{t+1} is computed, we update the weighting parameter in each iteration according to the following rule:

$$\gamma^{t+1} = \|\nabla f(y^{t+1})\|_\infty. \tag{7.13}$$

According to Theorem 7.2, this selection rule does not guarantee that a limit point of the algorithm (if it exists) satisfies the original cardinality constraint. In order to obtain a vector that satisfies the cardinality constraint, it is sufficient to project the obtained vector onto the constraint set in problem (7.1). We will show in the next section that this strategy usually results in higher quality solutions.

7.4 Numerical results

We consider the following sparse least-squares problem:

$$\begin{array}{ll} \text{minimize} & \|Ax - b\|_2^2 \\ \text{subject to} & \text{card}(x) \leq k, \quad \|x\|_\infty \leq M, \end{array} \tag{7.14}$$

with decision variable $x \in \mathbb{R}^n$ and problem data $A \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $M \in \mathbb{R}_+$, and $k \in \mathbb{N}$.

This problem arises, for instance, in compressed sensing where a sparse vector x must be recovered from linear measurements $Ax = b$ [24]. Observe that problem (7.14) satisfies Assumption 7.1 with $\zeta = 0$ and $L_f = 2\,\|A^TA\|_2$ where $\|\cdot\|_2$ denotes the matrix spectral norm.

We will show that for problem (7.14) the algorithm proposed in the previous section, denoted here by prox-PGM, produces solutions of a higher quality than other algorithms reported in the literature. We describe these algorithms in Appendix 7.A.

The data are generated as described in [67, §6.1], *i.e.* $A \in \mathbb{R}^{m\times 2m}$ with i.i.d. $\mathcal{N}(0,1)$ entries, $b = A\hat{x} + v$ with $\hat{x}$ drawn uniformly at random from the set of vectors satisfying $\text{card}(x) \leq \lfloor m/5 \rfloor$ and $\|x\|_\infty \leq M = 1$, and $v \in \mathbb{R}^m$ being a noise vector drawn from $\mathcal{N}(0, \sigma^2 I)$. We set $\sigma = \|A\hat{x}\|/(20\sqrt{m})$ so that the signal-to-noise ratio is near 20.

For each value of m we generate 100 instances of the problem. Since the quality of the solutions obtained depends on the initial point (except for the Lasso method), for each problem instance we run the algorithms from 10 initial points drawn from $\mathcal{N}(0, I)$, and keep the solution with the smallest objective value. The resulting solution is then *polished* in the sense that we replace the cardinality constraint in (7.14) with the sparsity pattern of the solution, and solve the resulting convex optimization problem to obtain a final solution. Although the authors in [67] propose additional heuristic steps such as *neighbor search* that

usually improves quality of the solutions obtained, we do not include these heuristics in our implementations. Note that these heuristics are not restricted to a specific algorithm, but can be used in any of the proposed methods that solve nonconvex problems.

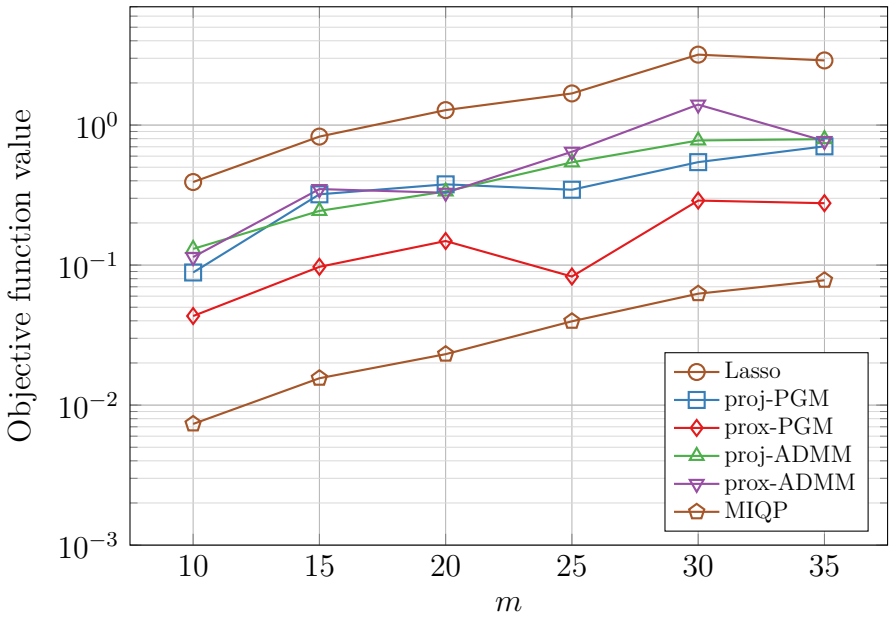
To make the comparison fair, we run all the algorithms for solving nonconvex problems for not more than 100 iterations. We use GUROBI [100] to solve convex quadratic programs (QPs) arising in Lasso and solution polishing, and mixed-integer quadratic programs (MIQPs).

The numerical results obtained are shown in Figure 7.1. For each value of m we show the average value of the objective function over the 100 generated instances. The exact solutions are obtained for small values of m by solving MIQPs. Solutions obtained with the Lasso approach have values of the objective function around one order of magnitude larger than solutions obtained by prox-PGM. Also, prox-PGM consistently outperforms all the other methods by at least a factor of 2 (relative to the exact solution for $m \leq 35$ and relative to zero for $m \geq 50$) for all values of m.

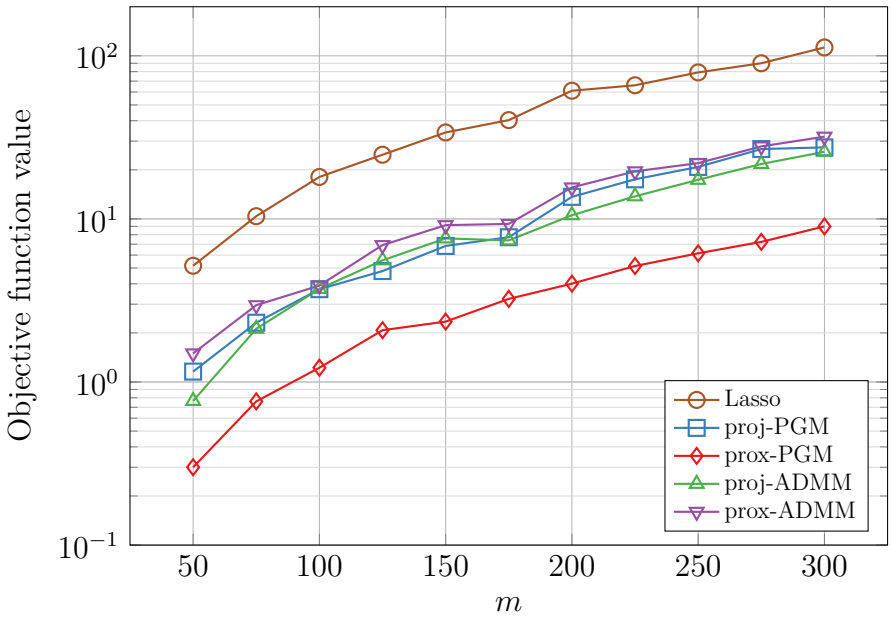
Figure 7.2 shows the average times for solving one instance of the problem. The time required to solve MIQPs grows rapidly with m and the approach is applicable only for solving small problems. On the other hand, runtimes of the operator splitting methods scale much better with the problem dimensions making them suitable for solving large-scale problems.

7.5 Conclusions

In this chapter we proposed a method for minimizing a convex differentiable function subject to sparsity constraints. We showed that, under suitably selected weighting parameter of a reformulated problem, PGM converges to a stationary point of the original problem. We also proposed a heuristic that updates the weighting parameter in each iteration of the algorithm. The performance of our method was compared to other methods proposed in the literature for solving such problems. Our method consistently outperforms all the other solution methods considered in this chapter by more than a factor of 2.



(a) $10 \leq m \leq 35$



(b) $50 \leq m \leq 300$

Figure 7.1: The average objective function values at solutions found by various algorithms for 100 instances of problem (7.14).

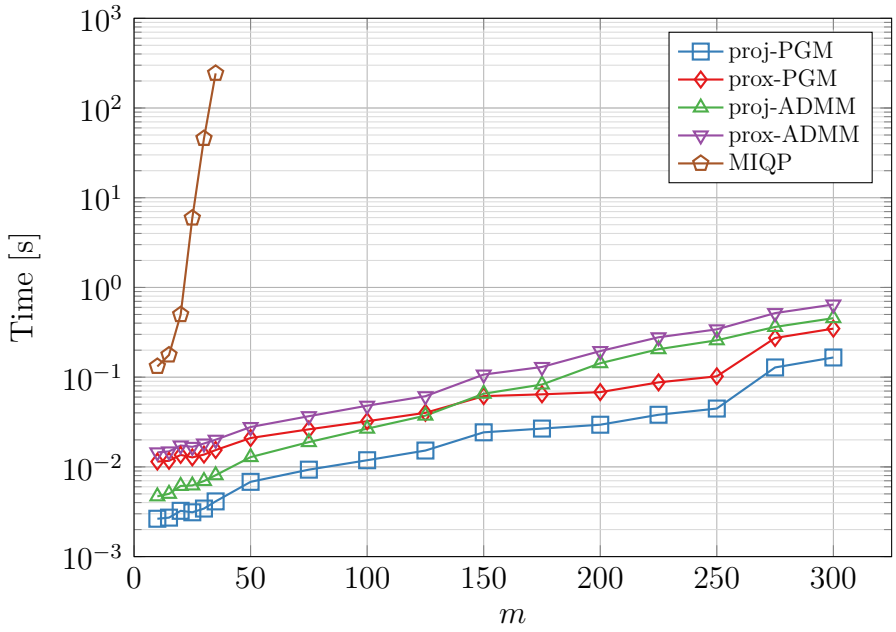


Figure 7.2: The average runtimes of various algorithms for solving 100 instances of problem (7.14).

7.A Tested algorithms

Lasso

The least absolute shrinkage and selection operator (Lasso) is a well-known heuristic for solving problem (7.14) and is based on a convex relaxation as in problem (7.2). We solve the problem for different values of the weighting parameter γ, and keep the solution obtained with the smallest value of the parameter for which the cardinality constraint is satisfied.

proj-PGM

The projected gradient method was used by [24] for solving a variant of problem (7.14), and is equivalent to Algorithm 7.1 when $\gamma = \infty$. The method converges to a stationary point, and the same termination criterion can be used as proposed for the prox-PGM.

proj-ADMM

A method proposed by [67] for solving problem (7.14) is equivalent to Algorithm 7.2, when $\gamma = \infty$. We set the algorithm parameter $\rho = L$. The method does not

Algorithm 7.2 ADMM for problem (7.5).

1: **given** initial values $z^0 \in \mathcal{B}$ and $u^0 = 0$, and parameter $\rho > 0$
2: **for** $t = 0 : N - 1$ **do**
3: $\quad x^{t+1} \leftarrow \text{prox}_{f/\rho}(z^t - u^t)$
4: $\quad z^{t+1} \leftarrow \text{prox}_{g/\rho}(x^{t+1} + u^t)$
5: $\quad u^{t+1} \leftarrow u^t + x^{t+1} - z^{t+1}$
6: **end for**

necessarily converge, and we terminate the algorithm after some fixed number of iterations is reached, as proposed by [67]. Here, we stop the algorithm after 100 iterations. Notice that evaluating the proximal operator of f requires solving a linear system, which is computationally more expensive than evaluating the gradient of f. However, since the left-hand side of the linear system does not change, we can factor the matrix once and use the cached factorization in the subsequent iterations.

prox-ADMM

As Algorithm 7.1 can be used for solving both problems (7.1) and (7.3), we can use Algorithm 7.2 also for solving problem (7.3). We denote this method by prox-ADMM and it can be seen as a generalization of proj-ADMM. We use the same strategy as in (7.13) to update the weighting parameter of the problem, with y^{t+1} replaced by x^{t+1}. Other implementation details are as in proj-ADMM.

8 Conclusions

Contents

8.1 Contributions of this dissertation

The main focus of this dissertation is the analysis and implementation of operator splitting methods for solving convex optimization problems. Specific contributions are as follows:

Infeasibility detection

Characterization of an operator splitting method as the fixed-point iteration of some nonlinear operator allows us to derive many interesting results for this important class of optimization methods. For instance, asymptotic behavior of the fixed-point iteration of an averaged operator has been studied for several decades. In the case when the fixed-point set of such an operator is empty, the iterates of its fixed-point iteration diverge, but the difference between consecutive iterates converge to a constant vector [18].

Using this result, we showed in Chapter 3 not only that the differences between consecutive primal and dual iterates of ADMM converge to some constant vectors

when the underlying problem is unsolvable, but also that these vectors are actually certificates of primal and/or dual infeasibility. Based on this result, we propose reliable termination criteria for detecting primal and dual infeasibility in ADMM.

The relevance of this result is that ADMM can provide infeasibility certificates without reformulating the problem using HSDE. This is especially important for a QP since solving an equivalent conic program is not efficient from a computational point of view.

Linear convergence rate

A known limitation of operator splitting methods is that they can sometimes converge very slowly. Linear convergence of an operator splitting method, meaning that some suboptimality measure of its iterates is upper-bounded by an exponentially decreasing sequence, can be established only in some special cases. In Chapters 4 and 5 we derive generalized conditions that ensure linear convergence of operator splitting methods in both single- and multi-agent case.

Fixed-point iteration. Using the aforementioned characterization of operator splitting methods, we also analyzed their convergence rate. If the fixed-point set of some averaged operator is nonempty, then its fixed-point iteration converges to a fixed-point but the convergence rate is sublinear in general. We showed that linear regularity of such operators is a necessary and sufficient condition for linear convergence of the associated fixed-point iteration, and that a bound on the linear convergence rate using this analysis is actually tight. This analysis does not require restrictive assumptions such as contractiveness of the operator.

Decentralized optimization. If operator splitting methods are used in multi-agent setting, then the associated fixed-point operator is not necessarily non-expansive. In order to establish linear convergence in such setting we adopted analysis based on the quadratic growth of the objective function, which does not require that the function is strongly convex.

Implementation of optimization methods

Operator splitting methods have recently attracted huge interest in the research community. In order to foster using the recent developments in real-world applications, there is an obvious need for fast and reliable software. Although there has been considerable progress in this regard [85, 136], a reliable and scalable general-purpose QP solver is still missing.

In Chapter 6 we introduced OSQP, a general-purpose QP solver based on ADMM. The solver is very robust and has been numerically tested on millions of problem

instances and a wide range of dimensions. It is the first operator-splitting QP solver that can reliably detect primal and dual infeasibility. OSQP is also able to generate customized solvers for given parametric QPs, which often arise in embedded applications.

Nonconvex optimization

Operator splitting methods are sometimes used for solving nonconvex optimization problems. We showed in Chapter 7 that PGM for minimizing the sum of a Lipschitz smooth convex function and any nonconvex function finds a stationary point of the problem. We used this method for solving convex problems with cardinality constraints. Numerical experiments showed that our method outperforms a common approach based on convex relaxation of the problem by an order of magnitude.

8.2 Directions for future research

Finally, we outline some possible directions for future research.

Superlinear convergence rate

The key motivation of this work is that operator splitting methods converge, at best, with a linear rate, which can be prohibitively slow in applications requiring high accuracy solutions. The authors in [160] have recently proposed a Newton-type scheme for finding fixed-points of nonexpansive operators, which robustifies operator splitting methods overcoming their sensitivity to ill-conditioning and, under certain regularity assumptions, converges superlinearly. It would be interesting to explore different schemes with which operator splitting methods attain superlinear convergence rate.

Computational methods

The ADMM algorithm introduced in Chapter 3 is the core of the OSQP solver. However, the same algorithm can be used to solve conic optimization problems including SOCPs and SDPs. A central motivation for this work is that state-of-the-art interior-point methods for solving SDPs do not scale well with the problem dimensions. It might be possible to find more efficient ways to evaluate projection onto a semidefinite cone than computing the eigenvalue decomposition of a matrix in each iteration without using the decompositions from previous iterations.

In order to improve practical performance of an optimization method, we could employ an acceleration scheme. While the Nesterov's acceleration has been widely

used in the optimization community, *Anderson's acceleration* seems to have been underexploited despite its considerable success in some other research areas [167]. This acceleration scheme has already been implemented in SCS [136], and it would be interesting to test its performance with OSQP.

Low-rank optimization

Notions such as order, complexity or dimensionality can often be expressed via the rank of an appropriate matrix, and the problem of low-rank approximation arises in many research areas including data analysis, image compression, and model order reduction.

In Chapter 7 we reformulated the nonconvex cardinality constraint using the difference of the ℓ_1–norm and the largest–k norm. It is possible to reformulate in a similar way the rank constraint via the difference of the spectral and the *Ky-Fan* k–norm [102]. The results from Chapter 7 could thus be extended to a more general optimization problem involving rank constraints.

References

[1] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd. "A rewriting system for convex optimization problems". In: *Journal of Control and Decision* 5.1 (2018), pp. 42–60.

[2] F. Allgöwer, T. A. Badgwell, J. S. Qin, J. B. Rawlings, and S. J. Wright. "Nonlinear Predictive Control and Moving Horizon Estimation – An Introductory Overview". In: *Advances in Control*. Springer London, 1999, pp. 391–449.

[3] P. R. Amestoy, T. A. Davis, and I. S. Duff. "Algorithm 837: AMD, an approximate minimum degree ordering algorithm". In: *ACM Transactions on Mathematical Software* 30.3 (2004), pp. 381–388.

[4] H. Attouch, J. Bolte, P. Redont, and A. Soubeyran. "Proximal alternating minimization and projection methods for nonconvex problems: an approach based on the Kurdyka-Łojasiewicz inequality". In: *Mathematics of Operations Research* 35.2 (2010), pp. 438–457.

[5] H. Attouch, J. Bolte, and B. F. Svaiter. "Convergence of descent methods for semi-algebraic and tame problems: proximal algorithms, forward-backward splitting, and regularized Gauss-Seidel methods". In: *Mathematical Programming* 137.1-2 (2013), pp. 91–129.

[6] J. B. Baillon, R. E. Bruck, and S. Reich. "On the asymptotic behavior of nonexpansive mappings and semigroups in Banach spaces". In: *Houston Journal of Mathematics* 4.1 (1978), pp. 1–9.

[7] H. Balakrishnan, I. Hwang, and C. J. Tomlin. "Polynomial approximation algorithms for belief matrix maintenance in identity management". In: *IEEE Conference on Decision and Control (CDC)*. 2004, pp. 4874–4879.

[8] G. Banjac and P. Goulart. "A novel approach for solving convex problems with cardinality constraints". In: *IFAC World Congress*. 2017, pp. 13182–13187.

[9] G. Banjac and P. Goulart. "Global linear convergence in operator splitting methods". In: *IEEE Conference on Decision and Control (CDC)*. 2016, pp. 233–238.

[10] G. Banjac and P. Goulart. "Tight global linear convergence rate bounds for operator splitting methods". In: *IEEE Transactions on Automatic Control (To appear)* (2018).

[11] G. Banjac, P. Goulart, B. Stellato, and S. Boyd. "Infeasibility detection in the alternating direction method of multipliers for convex optimization". In: *optimization-online.org* (2017).

[12] G. Banjac, K. Margellos, and P. Goulart. "On the convergence of a regularized Jacobi algorithm for convex optimization". In: *IEEE Transactions on Automatic Control* 63.4 (2018), pp. 1113–1119.

[13] G. Banjac, B. Stellato, N. Moehle, P. Goulart, A. Bemporad, and S. Boyd. "Embedded code generation using the OSQP solver". In: *IEEE Conference on Decision and Control (CDC)*. 2017, pp. 1906–1911.

[14] H. H. Bauschke, J. Y. Bello Cruz, T. T. A. Nghia, H. M. Phan, and X. Wang. "Optimal rates of linear convergence of relaxed alternating projections and generalized Douglas-Rachford methods for two subspaces". In: *Numerical Algorithms* 73.1 (2016), pp. 33–76.

[15] H. H. Bauschke, J. Y. Bello Cruz, T. T. A. Nghia, H. M. Phan, and X. Wang. "The rate of linear convergence of the Douglas-Rachford algorithm for subspaces is the cosine of the Friedrichs angle". In: *Journal of Approximation Theory* 185 (2014), pp. 63–79.

[16] H. H. Bauschke and J. M. Borwein. "On projection algorithms for solving convex feasibility problems". In: *SIAM Review* 38.3 (1996), pp. 367–426.

[17] H. H. Bauschke and P. L. Combettes. *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. Springer Science & Business Media, 2011.

[18] H. H. Bauschke, P. L. Combettes, and D. R. Luke. "Finding best approximation pairs relative to two closed convex sets in Hilbert spaces". In: *Journal of Approximation Theory* 127.2 (2004), pp. 178–192.

[19] H. H. Bauschke, M. N. Dao, and W. M. Moursi. "The Douglas-Rachford algorithm in the affine-convex case". In: *Operations Research Letters* 44.3 (2016), pp. 379–382.

[20] H. H. Bauschke, F. Deutsch, H. Hundal, and S.-H. Park. "Accelerating the convergence of the method of alternating projections". In: *Transactions of the American Mathematical Society* 355.9 (2003), pp. 3433–3461.

[21] H. H. Bauschke and W. M. Moursi. "On the Douglas-Rachford algorithm". In: *Mathematical Programming* 164.1 (2017), pp. 263–284.

[22] H. H. Bauschke and W. M. Moursi. "The Douglas-Rachford algorithm for two (not necessarily intersecting) affine subspaces". In: *SIAM Journal on Optimization* 26.2 (2016), pp. 968–985.

[23] H. H. Bauschke, D. Noll, and H. M. Phan. "Linear and strong convergence of algorithms involving averaged nonexpansive operators". In: *Journal of Mathematical Analysis and Applications* 421.1 (2015), pp. 1–20.

[24] A. Beck and Y. C. Eldar. "Sparsity constrained nonlinear optimization: optimality conditions and algorithms". In: *SIAM Journal on Optimization* 23.3 (2013), pp. 1480–1509.

[25] A. Beck and S. Shtern. "Linearly convergent away-step conditional gradient for non-strongly convex functions". In: *Mathematical Programming* 164.1-2 (2017), pp. 1–27.

[26] A. Beck and M. Teboulle. “Fast gradient-based algorithms for constrained total variation image denoising and deblurring problems”. In: *IEEE Transactions on Image Processing* 18.11 (2009), pp. 2419–2434.

[27] A. Beck and L. Tetruashvili. “On the convergence of block coordinate descent type methods”. In: *SIAM Journal on Optimization* 23.4 (2013), pp. 2037–2060.

[28] P. Belotti, C. Kirches, S. Leyffer, J. Linderoth, J. Luedtke, and A. Mahajan. “Mixed-integer nonlinear optimization”. In: *Acta Numerica* 22 (2013), pp. 1–131.

[29] A. Bemporad, F. Borrelli, and M. Morari. “Min-max control of constrained uncertain discrete-time linear systems”. In: *IEEE Transactions on Automatic Control* 48.9 (2003), pp. 1600–1606.

[30] A. Bemporad, F. Borrelli, and M. Morari. “Model predictive control based on linear programming – the explicit solution”. In: *IEEE Transactions on Automatic Control* 47.12 (2002), pp. 1974–1985.

[31] M. Benzi. “Preconditioning techniques for large linear systems: a survey”. In: *Journal of Computational Physics* 182.2 (2002), pp. 418–477.

[32] D. P. Bertsekas. “Incremental proximal methods for large scale convex optimization”. In: *Mathematical Programming* 129.2 (2011), pp. 163–195.

[33] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1995.

[34] D. P. Bertsekas and J. N. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.

[35] L. Blackmore, B. Açikmeşe, and D. P. Scharf. “Minimum-landing-error powered-descent guidance for Mars landing using convex optimization”. In: *Journal of Guidance, Control, and Dynamics* 33.4 (2010), pp. 1161–1171.

[36] L. G. Bleris and M. V. Kothare. “Real-time implementation of model predictive control”. In: *American Control Conference (ACC)*. 2005, pp. 4166–4171.

[37] T. Blumensath and M. E. Davies. “Iterative thresholding for sparse approximations”. In: *Journal of Fourier Analysis and Applications* 14.5-6 (2008), pp. 629–654.

[38] D. Boley. “Local linear convergence of the alternating direction method of multipliers on quadratic or linear programs”. In: *SIAM Journal on Optimization* 23.4 (2013), pp. 2183–2207.

[39] J. Bolte, T. P. Nguyen, J. Peypouquet, and B. W. Suter. “From error bounds to the complexity of first-order descent methods for convex functions”. In: *Mathematical Programming* 165.2 (2017), pp. 471–507.

[40] B. Borchers. “SDPLIB 1.2, a library of semidefinite programming test problems”. In: *Optimization Methods and Software* 11.1 (1999), pp. 683–690.

[41] F. Borrelli, A. Bemporad, and M. Morari. *Predictive Control for Linear and Hybrid Systems*. Cambridge University Press, 2017.

[42] S. Boyd, E. Busseti, S. Diamond, R. N. Kahn, K. Koh, P. Nystrup, and J. Speth. “Multi-period trading via convex optimization”. In: *Foundations and Trends in Optimization* 3.1 (2017), pp. 1–76.

[43] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. Society for Industrial and Applied Mathematics, 1994.

[44] S. Boyd, M. T. Mueller, B. O'Donoghue, and Y. Wang. "Performance bounds and suboptimal policies for multi–period investment". In: *Foundations and Trends in Optimization* 1.1 (2014), pp. 1–72.

[45] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. "Distributed optimization and statistical learning via the alternating direction method of multipliers". In: *Foundations and Trends in Machine Learning* 3.1 (2011), pp. 1–122.

[46] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

[47] A. Bradley. "Algorithms for the equilibration of matrices and their application to limited-memory quasi-Newton methods". PhD thesis. Stanford University, 2010.

[48] E. Candès, M. Wakin, and S. Boyd. "Enhancing sparsity by reweighted ℓ_1 minimization". In: *Journal of Fourier Analysis and Applications* 14.5 (2008), pp. 877–905.

[49] A. Cegielski. "Application of quasi-nonexpansive operators to an iterative method for variational inequality". In: *SIAM Journal on Optimization* 25.4 (2015), pp. 2165–2181.

[50] E. Chu, N. Parikh, A. Domahidi, and S. Boyd. "Code generation for embedded second-order cone programming". In: *European Control Conference (ECC)*. 2013, pp. 1547–1552.

[51] G. Cohen. "Optimization by decomposition and coordination: a unified approach". In: *IEEE Transactions on Automatic Control* 23.2 (1978), pp. 222–232.

[52] P. L. Combettes and J.-C. Pesquet. "Proximal splitting methods in signal processing". In: *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*. Springer New York, 2011, pp. 185–212.

[53] P. L. Combettes and V. R. Wajs. "Signal recovery by proximal forward-backward splitting". In: *Multiscale Modeling & Simulation* 4.4 (2005), pp. 1168–1200.

[54] G. Connor and R. A. Korajczyk. "The arbitrage pricing theory and multifactor models of asset returns". In: *Finance*. Vol. 9. Handbooks in Operations Research and Management Science. 1995, pp. 87–144.

[55] G. Cornuejols and R. Tütüncü. *Optimization Methods in Finance*. Mathematics, Finance and Risk. Cambridge University Press, 2006.

[56] C. Cortes and V. Vapnik. "Support-vector networks". In: *Machine Learning* 20.3 (1995), pp. 273–297.

[57] G. B. Dantzig. *Linear programming and extensions*. Princeton University Press, 1963.

[58] T. A. Davis. "Algorithm 849: a concise sparse Cholesky factorization package". In: *ACM Transactions on Mathematical Software* 31.4 (2005), pp. 587–591.

[59] T. A. Davis. *Direct Methods for Sparse Linear Systems*. Society for Industrial and Applied Mathematics, 2006.

[60] B. Defraene, T. Van Waterschoot, H. J. Ferreau, M. Diehl, and M. Moonen. "Real-time perception-based clipping of audio signals using convex optimization". In: *IEEE Transactions on Audio, Speech and Language Processing* 20.10 (2012), pp. 2657–2671.

[61] L. Deori, K. Margellos, and M. Prandini. "On decentralized convex optimization in a multi-agent setting with separable constraints and its application to optimal charging of electric vehicles". In: *IEEE Conference on Decision and Control (CDC)*. 2016, pp. 6044–6049.

[62] L. Deori, K. Margellos, and M. Prandini. "Regularized Jacobi iteration for decentralized convex quadratic optimization with separable constraints". In: *IEEE Transactions on Control Systems Technology (To appear)* (2018).

[63] F. Deutsch. *Best Approximation in Inner Product Spaces*. 1st. Springer-Verlag, 2001.

[64] F. Deutsch and H. Hundal. "The rate of convergence of Dykstra's cyclic projections algorithm: the polyhedral case". In: *Numerical Functional Analysis and Optimization* 15.5-6 (1994), pp. 537–565.

[65] S. Diamond and S. Boyd. "CVXPY: a Python-embedded modeling language for convex optimization". In: *Journal of Machine Learning Research* 17.83 (2016), pp. 1–5.

[66] S. Diamond and S. Boyd. "Stochastic matrix-free equilibration". In: *Journal of Optimization Theory and Applications* 172.2 (2017), pp. 436–454.

[67] S. Diamond, R. Takapoui, and S. Boyd. "A general system for heuristic minimization of convex functions over non-convex sets". In: *Optimization Methods and Software* 33.1 (2018), pp. 165–193.

[68] M. Diehl, H. J. Ferreau, and N. Haverbeke. "Efficient Numerical Methods for Nonlinear MPC and Moving Horizon Estimation". In: *Nonlinear Model Predictive Control: Towards New Challenging Applications*. Springer Berlin Heidelberg, 2009, pp. 391–417.

[69] E. D. Dolan and J. J. Moré. "Benchmarking optimization software with performance profiles". In: *Mathematical Programming* 91.2 (2002), pp. 201–213.

[70] A. Domahidi. *FORCES: Fast optimization for real-time control on embedded systems*. 2012. URL: `http://forces.ethz.ch`.

[71] A. Domahidi, E. Chu, and S. Boyd. "ECOS: an SOCP solver for embedded systems". In: *European Control Conference (ECC)*. 2013, pp. 3071–3076.

[72] A. Domahidi, A. U. Zgraggen, M. N. Zeilinger, M. Morari, and C. N. Jones. "Efficient interior point methods for multistage problems arising in receding horizon control". In: *IEEE Conference on Decision and Control (CDC)*. 2012, pp. 668–674.

[73] D. L. Donoho. “Compressed sensing”. In: *IEEE Transactions on Information Theory* 52.4 (2006), pp. 1289–1306.

[74] D. Drusvyatskiy and A. S. Lewis. “Error bounds, quadratic growth, and linear convergence of proximal methods”. In: *Mathematics of Operations Research (To appear)* (2018).

[75] D. Dueri, J. Zhang, and B. Açikmeşe. “Automated custom code generation for embedded, real-time second order cone programming”. In: *IFAC World Congress*. 2014, pp. 1605–1612.

[76] I. S. Duff, A. M. Erisman, and J. K. Reid. *Direct methods for sparse matrices*. Oxford University Press, 1989.

[77] I. Dunning, J. Huchette, and M. Lubin. “JuMP: a modeling language for mathematical optimization”. In: *SIAM Review* 59.2 (2017), pp. 295–320.

[78] J. Eckstein. “Parallel alternating direction multiplier decomposition of convex programs”. In: *Journal of Optimization Theory and Applications* 80.1 (1994), pp. 39–62.

[79] J. Eckstein and D. P. Bertsekas. *An alternating direction method for linear programming*. available: `http://dspace.mit.edu/handle/1721.1/3197`, 1990.

[80] J. Eckstein and D. P. Bertsekas. “On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators”. In: *Mathematical Programming* 55.1 (1992), pp. 293–318.

[81] J. Eckstein and M. C. Ferris. “Operator-splitting methods for monotone affine variational inequalities, with a parallel application to optimal control”. In: *INFORMS Journal on Computing* 10.2 (1998), pp. 218–235.

[82] M. Fält and P. Giselsson. “Optimal convergence rates for generalized alternating projections”. In: *IEEE Conference on Decision and Control (CDC)*. 2017, pp. 2268–2274.

[83] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl. “qpOASES: a parametric active-set algorithm for quadratic programming”. In: *Mathematical Programming Computation* 6.4 (2014), pp. 327–363.

[84] R. Fletcher and S. Leyffer. “Numerical experience with lower bounds for MIQP branch-and-bound”. In: *SIAM Journal on Optimization* 8.2 (1998), pp. 604–616.

[85] C. Fougner and S. Boyd. “Parameter selection and pre-conditioning for a graph form solver”. In: *Technical report, Stanford University* (2015), pp. 1–28.

[86] G. Frison, H. H. B. Sorensen, B. Dammann, and J. B. Jorgensen. “High-performance small-scale solvers for linear model predictive control”. In: *European Control Conference (ECC)*. 2014, pp. 128–133.

[87] D. Gabay. “Applications of the method of multipliers to variational inequalities”. In: *Studies in Mathematics and its Applications* 15.C (1983), pp. 299–331.

[88] C. E. García, D. M. Prett, and M. Morari. “Model predictive control: theory and practice – a survey”. In: *Automatica* 25.3 (1989), pp. 335–348.

[89] E. M. Gertz and S. J. Wright. "Object-oriented software for quadratic programming". In: *ACM Transactions on Mathematical Software* 29.1 (2003), pp. 58–81.

[90] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson. "Optimal parameter selection for the alternating direction method of multipliers (ADMM): quadratic problems". In: *IEEE Transactions on Automatic Control* 60.3 (2015), pp. 644–658.

[91] P. E. Gill, W. Murray, M. A. Saunders, J. A. Tomlin, and M. H. Wright. "On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method". In: *Mathematical Programming* 36.2 (1986), pp. 183–209.

[92] P. Giselsson. "Tight global linear convergence rate bounds for Douglas-Rachford splitting". In: *Journal of Fixed Point Theory and Applications* 19.4 (2017), pp. 2241–2270.

[93] P. Giselsson and S. Boyd. "Linear convergence and metric selection for Douglas-Rachford splitting and ADMM". In: *IEEE Transactions on Automatic Control* 62.2 (2017), pp. 532–544.

[94] P. Giselsson and S. Boyd. "Metric selection in fast dual forward–backward splitting". In: *Automatica* 62 (2015), pp. 1–10.

[95] P. Giselsson, M. Fält, and S. Boyd. "Line search for averaged operator iteration". In: *IEEE Conference on Decision and Control (CDC)*. 2016, pp. 1015–1022.

[96] G. H. Golub and C. F. Van Loan. *Matrix Computations*. 3rd ed. Johns Hopkins University Press, 1996.

[97] J. Gotoh, A. Takeda, and K. Tono. "DC formulations and algorithms for sparse optimization problems". In: *Mathematical Programming* (2017).

[98] S. Grammatico, F. Parise, M. Colombino, and J. Lygeros. "Decentralized convergence to Nash equilibria in constrained deterministic mean field control". In: *IEEE Transactions on Automatic Control* 61.11 (2016), pp. 3315–3329.

[99] A. Greenbaum. *Iterative Methods for Solving Linear Systems*. Society for Industrial and Applied Mathematics, 1997.

[100] Gurobi Optimization Inc. *Gurobi Optimizer Reference Manual*. 2016. URL: `http://www.gurobi.com`.

[101] B. S. He, H. Yang, and S. L. Wang. "Alternating direction method with self-adaptive penalty parameters for monotone variational inequalities". In: *Journal of Optimization Theory and Applications* 106.2 (2000), pp. 337–356.

[102] A. Hempel and P. Goulart. "A novel method for modelling cardinality and rank constraints". In: *IEEE Conference on Decision and Control (CDC)*. 2014, pp. 4322–4327.

[103] M. Herceg, C. N. Jones, and M. Morari. "Dominant speed factors of active set methods for fast MPC". In: *Optimal Control Applications and Methods* 36.5 (2015), pp. 608–627.

[104] M. Hong, X. Wang, M. Razaviyayn, and Z.-Q. Luo. "Iteration complexity analysis of block coordinate descent methods". In: *Mathematical Programming* 163.1-2 (2017), pp. 85–114.

[105] P. J. Huber. "Robust estimation of a location parameter". In: *The Annals of Mathematical Statistics* 35.1 (1964), pp. 73–101.

[106] P. J. Huber. *Robust Statistics*. John Wiley & Sons, 1981.

[107] Intel Corporation. *Intel Math Kernel Library. User's Guide*. Intel Corporation, 2017.

[108] F. Iutzeler, P. Bianchi, P. Ciblat, and W. Hachem. "Explicit convergence rate of a distributed alternating direction method of multipliers". In: *IEEE Transactions on Automatic Control* 61.4 (2016), pp. 892–904.

[109] J. Jerez, P. Goulart, S. Richter, G. Constantinides, E. Kerrigan, and M. Morari. "Embedded online optimization for model predictive control at megahertz rates". In: *IEEE Transactions on Automatic Control* 59.12 (2014), pp. 3238–3251.

[110] N. Karmarkar. "A new polynomial-time algorithm for linear programming". In: *Combinatorica* 4.4 (1984), pp. 373–395.

[111] C. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. Society for Industrial and Applied Mathematics, 1995.

[112] E. C. Kerrigan and J. M. Maciejowski. "Feedback min-max model predictive control using a single linear program: robust stability and the explicit solution". In: *International Journal of Robust and Nonlinear Control* 14.4 (2004), pp. 395–413.

[113] V. Klee and G. Minty. "How good is the simplex algorithm". In: *Technical Report, University of Washington* (1970).

[114] F. Lin, M. Fardad, and M. R. Jovanović. "Design of optimal sparse feedback gains via the alternating direction method of multipliers". In: *IEEE Transactions on Automatic Control* 58.9 (2013), pp. 2426–2431.

[115] P. L. Lions and B. Mercier. "Splitting algorithms for the sum of two nonlinear operators". In: *SIAM Journal on Numerical Analysis* 16.6 (1979), pp. 964–979.

[116] J. Löfberg. "YALMIP: a toolbox for modeling and optimization in MATLAB". In: *IEEE International Conference on Robotics and Automation*. 2004, pp. 284–289.

[117] D. R. Luke, N. H. Thao, and M. K. Tam. "Implicit error bounds for Picard iterations on Hilbert spaces". In: *Vietnam Journal of Mathematics* 46.2 (2018), pp. 243–258.

[118] Z.-Q. Luo and P. Tseng. "On the linear convergence of descent methods for convex essentially smooth minimization". In: *SIAM Journal on Control and Optimization* 30.2 (1992), pp. 408–425.

[119] J. Mairal, F. Bach, and J. Ponce. "Sparse modeling for image and vision processing". In: *Foundations and Trends in Computer Graphics and Vision* 8.2–3 (2014), pp. 85–283.

[120] H. Markowitz. "Portfolio selection". In: *The Journal of Finance* 7.1 (1952), pp. 77–91.

[121] I. Maros and C. Mészáros. "A repository of convex quadratic programming problems". In: *Optimization Methods and Software* 11.1–4 (1999), pp. 671–681.

[122] J. Mattingley and S. Boyd. "CVXGEN: a code generator for embedded convex optimization". In: *Optimization and Engineering* 13.1 (2012), pp. 1–27.

[123] J. Mattingley and S. Boyd. "Real-time convex optimization in signal processing". In: *IEEE Signal Processing Magazine* 27.3 (2010), pp. 50–61.

[124] S. Mehrotra. "On the implementation of a primal-dual interior point method". In: *SIAM Journal on Optimization* 2.4 (1992), pp. 575–601.

[125] A. Miller. *Subset Selection in Regression*. Chapman & Hall. 2002.

[126] MOSEK ApS. *The MOSEK optimization toolbox for MATLAB manual. Version 8.0 (Revision 57).* 2017. URL: `https://www.mosek.com`.

[127] V. V. Naik and A. Bemporad. "Embedded mixed-integer quadratic optimization using accelerated dual gradient projection". In: *IFAC World Congress*. 2017, pp. 10723–10728.

[128] I. Necoara and D. Clipici. "Parallel random coordinate descent method for composite minimization: convergence analysis and error bounds". In: *SIAM Journal on Optimization* 26.1 (2016), pp. 197–226.

[129] I. Necoara, Yu. Nesterov, and F. Glineur. "Linear convergence of first order methods for non-strongly convex optimization". In: *Mathematical Programming (To appear)* (2018).

[130] Yu. Nesterov. *Introductory lectures on convex optimization: A basic course*. Applied optimization. Kluwer Academic Publishers, 2004.

[131] Yu. Nesterov and A. Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. Society for Industrial and Applied Mathematics, 1994.

[132] J. von Neumann. *Functional Operators, Vol. II*. Princeton University Press, 1950.

[133] R. Nishihara, S. Jegelka, and M. I. Jordan. "On the convergence rate of decomposable submodular function minimization". In: *International Conference on Neural Information Processing Systems (NIPS)*. 2014, pp. 640–648.

[134] J. Nocedal and S. J. Wright. *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer, 2006.

[135] B. O'Donoghue and E. Candès. "Adaptive restart for accelerated gradient schemes". In: *Foundations of Computational Mathematics* 15.3 (2015), pp. 715–732.

[136] B. O'Donoghue, E. Chu, N. Parikh, and S. Boyd. "Conic optimization via operator splitting and homogeneous self-dual embedding". In: *Journal of Optimization Theory and Applications* 169.3 (2016), pp. 1042–1068.

[137] B. O'Donoghue, G. Stathopoulos, and S. Boyd. "A splitting method for optimal control". In: *IEEE Transactions on Control Systems Technology* 21.6 (2013), pp. 2432–2442.

[138] D. Paccagnan, M. Kamgarpour, and J. Lygeros. "On aggregative and mean field games with applications to electricity markets". In: *European Control Conference (ECC)*. 2016, pp. 196–201.

[139] N. Parikh and S. Boyd. "Proximal algorithms". In: *Foundations and Trends in Optimization* 1.3 (2013), pp. 123–231.

[140] A. Pazy. "Asymptotic behavior of contractions in Hilbert space". In: *Israel Journal of Mathematics* 9.2 (1971), pp. 235–240.

[141] T. Pock and A. Chambolle. "Diagonal preconditioning for first order primal-dual algorithms in convex optimization". In: *International Conference on Computer Vision*. 2011, pp. 1762–1769.

[142] A. U. Raghunathan and S. Di Cairano. "Infeasibility detection in alternating direction method of multipliers for convex quadratic programs". In: *IEEE Conference on Decision and Control (CDC)*. 2014, pp. 5819–5824.

[143] M. V. Ramana. "An exact duality theory for semidefinite programming and its complexity implications". In: *Mathematical Programming* 77.1 (1997), pp. 129–162.

[144] C. V. Rao and J. B. Rawlings. "Linear programming and model predictive control". In: *Journal of Process Control* 10.2–3 (2000), pp. 283–289.

[145] J. B. Rawlings and D. Q. Mayne. *Model Predictive Control: Theory and Design*. Nob Hill Publishing, 2009.

[146] P. Richtárik and M. Takáč. "Parallel coordinate descent methods for big data optimization". In: *Mathematical Programming* 156.1–2 (2016), pp. 433–484.

[147] S. Richter. "Computational Complexity Certification of Gradient Methods for Real-Time Model Predictive Control". PhD thesis. ETH-Zürich, 2012.

[148] S. Richter, C. N. Jones, and M. Morari. "Certification aspects of the fast gradient method for solving the dual of parametric convex programs". In: *Mathematical Methods of Operations Research* 77.3 (2013), pp. 305–321.

[149] S. Richter, C. N. Jones, and M. Morari. "Computational complexity certification for real-time MPC with input constraints based on the fast gradient method". In: *IEEE Transactions on Automatic Control* 57.6 (2012), pp. 1391–1403.

[150] R. T. Rockafellar. "Characterization of the subdifferentials of convex functions". In: *Pacific Journal of Mathematics* 17.3 (1966), pp. 497–510.

[151] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, USA, 1970.

[152] R. T. Rockafellar and R. J.-B. Wets. *Variational Analysis*. Springer-Verlag, 1998.

[153] D. Ruiz. "A scaling algorithm to equilibrate both rows and columns norms in matrices". In: *Technical report, Rutherford Appleton Laboratory* (2001), pp. 1–21.

[154] E. Ryu and S. Boyd. "Primer on monotone operator methods". In: *International Journal of Applied and Computational Mathematics* 15.1 (2016), pp. 3–43.

[155] D. P. Scharf, B. Açikmeşe, D. Dueri, J. Benito, and J. Casoliva. "Implementation and experimental demonstration of onboard powered-descent guidance". In: *Journal of Guidance, Control, and Dynamics* 40.2 (2017), pp. 213–229.

[156] R. Sinkhorn and P. Knopp. "Concerning nonnegative matrices and doubly stochastic matrices". In: *Pacific Journal of Mathematics* 21.2 (1967), pp. 343–348.

[157] G. Stathopoulos, H. Shukla, A. Szucs, Y. Pu, and C. Jones. "Operator splitting methods in control". In: *Foundations and Trends in Systems and Control* 3.3 (2016), pp. 249–362.

[158] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. "OSQP: an operator splitting solver for quadratic programs". In: *arXiv:1711.08013* (2017).

[159] W. Su, S. Boyd, and E. Candès. "A differential equation for modeling Nesterov's accelerated gradient method: theory and insights". In: *Journal of Machine Learning Research* 17 (2016), pp. 1–43.

[160] A. Themelis and P. Patrinos. "SuperMann: a superlinearly convergent algorithm for finding fixed points of nonexpansive operators". In: *arXiv:1609.06955* (2016).

[161] R. Tibshirani. "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society: Series B* 58.1 (1996), pp. 267–288.

[162] P. Tøndel, T. A. Johansen, and A. Bemporad. "An algorithm for multi-parametric quadratic programming and explicit MPC solutions". In: *Automatica* 39.3 (2003), pp. 489–497.

[163] P. Tseng. "Convergence of a block coordinate descent method for nondifferentiable minimization". In: *Journal of Optimization Theory and Applications* 109.3 (2001), pp. 475–494.

[164] F. Ullmann and S. Richter. *FiOrdOs: Code generation for first-order methods, Version 2.0*. 2014. URL: `http://fiordos.ethz.ch`.

[165] R. Vanderbei. "Symmetric quasi-definite matrices". In: *SIAM Journal on Optimization* 5.1 (1995), pp. 100–113.

[166] A. Wächter and L. T. Biegler. "On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming". In: *Mathematical Programming* 106.1 (2006), pp. 25–57.

[167] H. F. Walker and P. Ni. "Anderson acceleration for fixed-point iterations". In: *SIAM Journal on Numerical Analysis* 49.4 (2011), pp. 1715–1735.

[168] P.-W. Wang and C.-J. Lin. "Iteration complexity of feasible descent methods for convex optimization". In: *Journal of Machine Learning Research* 15 (2014), pp. 1523–1548.

[169] Y. Wang and S. Boyd. "Fast model predictive control using online optimization". In: *IEEE Transactions on Control Systems Technology* 18.2 (2010), pp. 267–278.

[170] B. Wohlberg. "ADMM penalty parameter selection by residual balancing". In: *arXiv:1704.06209* (2017).

[171] P. Wolfe. "The simplex method for quadratic programming". In: *Econometrica* 27.3 (1959), pp. 382–398.

[172] S. J. Wright. *Primal-Dual Interior-Point Methods*. Society for Industrial and Applied Mathematics, 1997.

[173] Y. Xu and W. Yin. "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion". In: *SIAM Journal on Imaging Sciences* 6.3 (2013), pp. 1758–1789.

[174] Z. Xu, M. A. Figueiredo, and T. Goldstein. "Adaptive ADMM with spectral penalty parameter selection". In: *International Conference on Artificial Intelligence and Statistics (AISTATS)*. 2017, pp. 718–727.

[175] H. Zhang. "The restricted strong convexity revisited: analysis of equivalence to error bound and quadratic growth". In: *Optimization Letters* 11.4 (2017), pp. 817–833.

[176] Y. Zheng, G. Fantuzzi, A. Papachristodoulou, P. Goulart, and A. Wynn. "Fast ADMM for semidefinite programs with chordal sparsity". In: *American Control Conference (ACC)*. 2017, pp. 3335–3340.