## THE UNIVERSITY of EDINBURGH

This thesis has been submitted in fulfilment of the requirements for a postgraduate degree (e.g. PhD, MPhil, DClinPsychol) at the University of Edinburgh. Please note the following terms and conditions of use:

This work is protected by copyright and other intellectual property rights, which are retained by the thesis author, unless otherwise stated.
A copy can be downloaded for personal non-commercial research or study, without prior permission or charge.
This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the author.
The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the author.
When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given.

# Regularized Interior Point Methods for Convex <br> Programming 

Spyridon Pougkakiotis

Doctor of Philosophy
University of Edinburgh
2021

## Declaration

I declare that this thesis was composed by myself and that the work contained therein is my own, except where explicitly stated otherwise in the text. This work has not been submitted for any other degree or professional qualification.
(Spyridon Pougkakiotis)

To my parents,
Christos and Maria.

## Abstract

Interior point methods (IPMs) constitute one of the most important classes of optimization methods, due to their unparalleled robustness, as well as their generality. It is well known that a very large class of convex optimization problems can be solved by means of IPMs, in a polynomial number of iterations. As a result, IPMs are being used to solve problems arising in a plethora of fields, ranging from physics, engineering, and mathematics, to the social sciences, to name just a few. Nevertheless, there remain certain numerical issues that have not yet been addressed. More specifically, the main drawback of IPMs is that the linear algebra task involved is inherently ill-conditioned. At every iteration of the method, one has to solve a (possibly large-scale) linear system of equations (also known as the Newton system), the conditioning of which deteriorates as the IPM converges to an optimal solution. If these linear systems are of very large dimension, prohibiting the use of direct factorization, then iterative schemes may have to be employed. Such schemes are significantly affected by the inherent ill-conditioning within IPMs.

One common approach for improving the aforementioned numerical issues, is to employ regularized IPM variants. Such methods tend to be more robust and numerically stable in practice. Over the last two decades, the theory behind regularization has been significantly advanced. In particular, it is well known that regularized IPM variants can be interpreted as hybrid approaches combining IPMs with the proximal point method. However, it remained unknown whether regularized IPMs retain the polynomial complexity of their non-regularized counterparts. Furthermore, the very important issue of tuning the regularization parameters appropriately, which is also crucial in augmented Lagrangian methods, was not addressed.

In this thesis, we focus on addressing the previous open questions, as well as on creating robust implementations that solve various convex optimization problems. We discuss in detail the effect of regularization, and derive two different regularization strategies; one based on the proximal method of multipliers, and another one based on a Bregman proximal point method. The latter tends to be more efficient, while the former is more robust and has better convergence guarantees. In addition, we discuss the use of iterative linear algebra within the presented algorithms, by proposing some general purpose preconditioning strategies (used to accelerate the iterative schemes) that take advantage of the regularized nature of the systems being solved.

In Chapter 2 we present a dynamic non-diagonal regularization for IPMs. The non-diagonal aspect of this regularization is implicit, since all the off-diagonal el-
ements of the regularization matrices are cancelled out by those elements present in the Newton system, which do not contribute important information in the computation of the Newton direction. Such a regularization, which can be interpreted as the application of a Bregman proximal point method, has multiple goals. The obvious one is to improve the spectral properties of the Newton system solved at each IPM iteration. On the other hand, the regularization matrices introduce sparsity to the aforementioned linear system, allowing for more efficient factorizations. We propose a rule for tuning the regularization dynamically based on the properties of the problem, such that sufficiently large eigenvalues of the non-regularized system are perturbed insignificantly. This alleviates the need of finding specific regularization values through experimentation, which is the most common approach in the literature. We provide perturbation bounds for the eigenvalues of the non-regularized system matrix, and then discuss the spectral properties of the regularized matrix. Finally, we demonstrate the efficiency of the method applied to solve standard small- and medium-scale linear and convex quadratic programming test problems.

In Chapter 3 we combine an IPM with the proximal method of multipliers (PMM). The resulting algorithm (IP-PMM) is interpreted as a primal-dual regularized IPM, suitable for solving linearly constrained convex quadratic programming problems. We apply few iterations of the interior point method to each sub-problem of the proximal method of multipliers. Once a satisfactory solution of the PMM sub-problem is found, we update the PMM parameters, form a new IPM neighbourhood, and repeat this process. Given this framework, we prove polynomial complexity of the algorithm, under standard assumptions. To our knowledge, this is the first polynomial complexity result for a primal-dual regularized IPM. The algorithm is guided by the use of a single penalty parameter; that of the logarithmic barrier. In other words, we show that IP-PMM inherits the polynomial complexity of IPMs, as well as the strong convexity of the PMM sub-problems. The updates of the penalty parameter are controlled by IPM, and hence are well-tuned, and do not depend on the problem solved. Furthermore, we study the behavior of the method when it is applied to an infeasible problem, and identify a necessary condition for infeasibility. The latter is used to construct an infeasibility detection mechanism. Subsequently, we provide a robust implementation of the presented algorithm and test it over a set of small to large scale linear and convex quadratic programming problems, demonstrating the benefits of using regularization in IPMs as well as the reliability of the approach.

In Chapter 4 we extend IP-PMM to the case of linear semi-definite programming (SDP) problems. In particular, we prove polynomial complexity of the algorithm, under mild assumptions, and without requiring exact computations for the Newton directions. We furthermore provide a necessary condition for lack of strong duality, which can be used as a basis for constructing detection mechanisms for identifying pathological cases within IP-PMM.

In Chapter 5 we present general-purpose preconditioners for regularized Newton systems arising within regularized interior point methods. We discuss positive definite preconditioners, suitable for iterative schemes like the conjugate gradient (CG), or the minimal residual (MINRES) method. We study the spectral properties of the preconditioned systems, and discuss the use of each presented
approach, depending on the properties of the problem under consideration. All preconditioning strategies are numerically tested on various medium- to largescale problems coming from standard test sets, as well as problems arising from partial differential equation (PDE) optimization.

In Chapter 6 we apply specialized regularized IPM variants to problems arising from portfolio optimization, machine learning, image processing, and statistics. Such problems are usually solved by specialized first-order approaches. The efficiency of the proposed regularized IPM variants is confirmed by comparing them against problem-specific state-of-the-art first-order alternatives given in the literature.

Finally, in Chapter 7 we present some conclusions as well as open questions, and possible future research directions.

## Lay Summary

Optimization methods have seen increasing applicability during the era of information. Indeed, these can be used to determine optimal strategies for a plethora of different real-life problems. In light of the availability of computing power, optimization has become ubiquitous during the last three decades.

One of the most general and reliable family of optimization schemes arises from the so-called interior point methods. Such methods have been used in an extremely wide range of applications, originating from mathematical sciences, engineering, social sciences, logistics, medicine or economics, to name just a few. These methods are very popular, since they are able to reliably solve different types of problems very accurately, without the need of a technical background from the user. Indeed, interior point methods are often used as a black-box. While this is very commonly done, it might be possible that traditional interior point methods will struggle when applied to certain problems. This is because standard interior point methods require certain conditions to hold for the mathematical model describing the problem at hand. If the user is not aware of these conditions, and applies such a method for solving a model that does not abide by them, the method might not be able to terminate successfully.

A technique that is commonly used to ensure that interior point methods will provide a meaningful solution even if the provided model does not abide by the standard conditions required by traditional interior point methods, is the socalled regularization. Regularization is a very successful strategy, but it is often used as a heuristic. Indeed, very few works have been dedicated to the study of the effects of regularization when applied in the context of interior point methods.

In light of the previous, in this thesis we study the effects of regularization in the context of interior point methods for a large class of problems that appear in numerous applications areas. In particular, we analyze regularization theoretically while numerically verifying our results and showcasing the impact and importance of regularization in a variety of settings. We provide various black-box as well as specialized interior point solvers that can be utilized to solve problems for which traditional interior point schemes might struggle. Additionally, we demonstrate that the proposed solvers are extremely reliable, and hence can be employed without the need of a technical background from the user. Finally, we argue that regularized interior point methods can very often be more efficient than their non-regularized counterparts. Our observations are numerically verified on real-life problems arising from various applications areas, such as portfolio optimization, machine learning and statistics, medical imaging, as well as optimization over physical systems.

## Acknowledgements

I would like to express my sincere gratitude to my supervisor, Prof. Jacek Gondzio, for his constant guidance during my first steps in research. I would not have been able to continue on this journey without having such an inspiring mentor and teacher, setting the standards I wholeheartedly hope to live up to both in my academic as well as personal life. Thank you for your continuous feedback and support, your experienced suggestions and career advice, and for so many invaluable lessons. I am very grateful for being able to have you as my advisor and mentor for all these years.

I would also like to thank my second supervisor, Dr. John W. Pearson, for a plethora of discussions and fruitful conversations. With your support I was able to step outside of my comfort zone, and explore areas of mathematics that I had never encountered before. I am also grateful to my examination committee, Prof. Raphael Hauser and Dr. Alper Yildirim, for offering their valuable time.

For many stimulating conversations as well as enjoyable moments, I thank various students and members of our research group: Kresimir Mihic, Ivet Galabova, Maximilian Osborne, Filippo Zanetti, Stefano Cipolla, Santolo Leveque, Marion Lemery, Nagisa Sugishita, Tom Byrne, Sarathorn Phusingha, Rodrigo Garcia Nava, Minerva Martin del Campo, and Michael Feldmeier.

I am indebted to the University of Edinburgh and to the Principal's Career Development PhD Scholarship for funding my PhD studies. I would like to thank the school of Mathematics of the University of Edinburgh for funding my MSc studies, and for providing me with an amazing environment. I am very grateful to Iain Dornan and Tatiana Chepelina who were always very helpful whenever I encountered any issue. I am also extremely grateful to Prof. Jacek Gondzio, who financially supported me for an additional year, and alongside the Dr. Laura Wisewell Travel scholarship, funded my conference travel expenses. Being able to travel and join international research conferences was an invaluable experience. I would also like to thank the A.G. Leventis Foundation for the financial support during my PhD studies.

For their very valuable lessons, I would like to thank Dr. Julian Hall, Prof. Luca Bergamaschi, Dr. Angéles Martínez, Prof. Daniela di Serafino, Dr. Valentina De Simone, and Dr. Marco Viola. In addition, I am extremely thankful to Prof. Sergios Theodoridis and Dr. Pantelis Bouboulis for introducing me to mathematical optimization and for helping me start my research career.

This journey would not have been possible without the support of my parents, Christos and Maria, as well as my siblings, Vasiliki and Konstantinos. Thank you for encouraging me in all of my pursuits and for continually supporting me.

## Contents

Abstract ..... 7
1 Introduction ..... 16
1.1 Convex programming ..... 16
1.1.1 A primal-dual interior point method ..... 17
1.1.2 Proximal point methods ..... 19
1.1.3 Regularization in interior point methods ..... 22
1.2 Iterative solution of linear systems ..... 23
1.3 Structure of the thesis ..... 25
1.4 Summary ..... 26
2 Dynamic Regularization in IPMs for Convex QP ..... 28
2.1 Introduction ..... 28
2.2 Exact primal-dual regularization ..... 30
2.2.1 Problem formulation ..... 30
2.2.2 The Newton system ..... 31
2.2.3 The regularization matrices ..... 34
2.3 Spectral analysis ..... 44
2.3.1 Linear programming ..... 44
2.3.2 Quadratic programming ..... 48
2.4 Implementation and numerical results ..... 53
2.4.1 The algorithmic framework ..... 53
2.4.2 Implementation details ..... 53
2.4.3 Numerical results ..... 57
2.5 Conclusions ..... 61
3 An IP-PMM for Convex QP ..... 62
3.1 Introduction ..... 62
3.2 Algorithmic framework ..... 63
3.3 Convergence analysis of IP-PMM ..... 67
3.4 Infeasible problems ..... 83
3.5 Computational experience ..... 86
3.5.1 Implementation details ..... 86
3.5.2 Numerical results ..... 89
3.6 Conclusions ..... 94
4 An IP-PMM for Linear SDP ..... 97
4.1 Introduction ..... 97
4.2 Preliminaries and notation ..... 98
4.2.1 Primal-dual pair of SDP problems ..... 98
4.2.2 A proximal method of multipliers ..... 99
4.2.3 An infeasible interior point method ..... 100
4.2.4 Vectorized format ..... 101
4.3 An interior point-proximal method of multipliers for SDP ..... 102
4.4 Convergence analysis ..... 105
4.5 A sufficient condition for strong duality ..... 117
4.6 Conclusions ..... 120
5 Preconditioning for Regularized IPMs ..... 121
5.1 Introduction ..... 121
5.2 Regularized normal equations ..... 123
5.2.1 A Cholesky-based preconditioner ..... 124
5.2.2 A LDL ${ }^{\top}$-based preconditioner ..... 129
5.2.3 BFGS-like low-rank updates of the preconditioner ..... 130
5.3 Regularized saddle point matrices ..... 131
5.3.1 Block diagonal preconditioners ..... 132
5.3.2 Factorization-based preconditioners ..... 135
5.4 Regularized IPMs: numerical results ..... 136
5.4.1 Linear programming ..... 137
5.4.2 Convex quadratic programming ..... 144
5.5 Conclusions ..... 150
6 IP-PMMs for Sparse Approximation Problems ..... 151
6.1 Introduction ..... 151
6.2 An IP-PMM for convex programming ..... 153
6.2.1 Testing environment ..... 156
6.3 Portfolio selection problem ..... 156
6.3.1 Specialized IP-PMM for quadratic portfolio optimization problems ..... 158
6.3.2 Computational experience ..... 159
6.4 Classification models for functional magnetic resonance imaging data1 ..... 162
6.4.1 Specialized IP-PMM for fused lasso least squares ..... 164
6.4.2 Computational experience ..... 166
6.5 TV-based Poisson image restoration ..... 169
6.5.1 Specialized IP-PMM for image restoration problems ..... 170
6.5.2 Computational experience ..... 172
6.6 Linear classification via regularized logistic regression ..... 175
6.6.1 Computational experience ..... 176
6.7 Conclusions ..... 177
7 Conclusions and Future Directions ..... 179
7.1 Conclusions ..... 179
7.2 Future directions ..... 180
A Convergence Analysis of IP-PMM for SDP: residual proofs ..... 200
A. 1 Proof of Lemma 4.4.2 ..... 200
A. 2 Proof of Lemma 4.4.4 ..... 201
A. 3 Proof of Theorem 4.4.1 ..... 203
A. 4 Proof of Theorem 4.4.3 ..... 203
A. 5 Proof of Theorem 4.5.1 ..... 204

## Notation and Abbreviations

| $\mathcal{A}$ | An index set, unless stated otherwise. |
| :---: | :---: |
| $\|\mathcal{A}\|$ | The cardinality of the set $\mathcal{A}$. |
| $\succeq$ | Loewner ordering inequality. |
| $\mathbb{R}^{n}$ | Space of $n$-dimensional vectors. |
| $\mathbb{R}^{n \times n}$ | Space of real $m \times n$ matrices. |
| $\mathbb{R}_{+}$ | $x \in \mathbb{R}$, such that $x \geq 0$ (non-negative real number). |
| $\mathbb{R}_{+}^{n}$ | $x \in \mathbb{R}^{n}$, such that $x^{j} \in \mathbb{R}_{+}$, for all $j \in\{1, \ldots, n\}$ (nonnegative orthant). |
| $\mathcal{S}^{n}$ | The space of $n \times n$ symmetric matrices |
| $\mathcal{S}_{+}^{n}$ | The space of $n \times n$ symmetric positive semi-definite matrices |
| $\mathcal{S}_{++}^{n}$ | The space of $n \times n$ symmetric positive definite matrices |
| $H^{1}(\Omega)$ | The Sobolev space $W^{1,2}(\Omega)$ |
| $L^{p}(\Omega)$ | The space of measurable functions with support $\Omega$ for which the $p$-th power of the absolute value is Lebesgue integrable |
| $x^{j}$ | The $j$-th component of the column vector $x$ (sub-scripts are reserved for iteration counters or for distinguishing different variables). |
| $A^{(i, j)}$ | The ( $i, j$ )-th element of matrix $A$. |
| $A^{\top}$ | The transpose of $A$. |
| $A^{H} \equiv(\bar{A})^{\top}$ | The conjugate transpose of $A$. |
| $A^{(\mathcal{A}, \mathcal{B})}$ | The ( $i, j$ ) elements of $A$ such that $i \in \mathcal{A}, j \in \mathcal{B}$. |
| $A^{\mathcal{B}}$ | The columns $j$ of $A$ such that $j \in \mathcal{B}$. If $A$ is square, the same notation represents the $(i, j)$ elements of $A$ such that $i, j \in \mathcal{B}$. The distinction is clear from the context. |
| $\operatorname{Diag}(A)$ | The diagonal matrix containing the diagonal elements of $A$. |


| $X \equiv \operatorname{Diag}(x)$ | The diagonal matrix $X$ such that $X^{(i, i)}=x^{i}$. |
| :---: | :---: |
| $\operatorname{Off}(A)$ | The matrix containing the off-diagonal elements of $A$ and zeros in the diagonal. |
| $\\|\cdot\\|$ | Euclidean norm (vector) or Schatten 2-norm (matrix). |
| $\\|\cdot\\|_{x}$ | The (Schatten, for matrix inputs) $x$-norm, $x=1,2, \ldots$ |
| $\\|x\\|_{B}$ | The $B$-norm equal to $\sqrt{x^{\top} B x}$, with $B \in \mathcal{S}_{+}^{n}$. |
| $e_{n}$ | Vector of ones of size $n$. |
| $I_{n}$ | Identity matrix of size $n$. |
| $0_{m, n}$ | Zero matrix of size $m \times n$ (size might be omitted if it is clear). |
| $x_{k}$ | A vector $x$ that depends on iteration $k \geq 0$. |
| $x^{*}$ | Denotes the optimal solution of a problem with unknown $x$. |
| $\sigma(A)$ | The set of singular values of $A$. |
| $\sigma_{\text {max }}(A)\left(\sigma_{\text {min }}(A)\right)$ | Maximum (minimum resp.) singular value of a matrix $A$. |
| $\lambda(A)$ | The set of eigenvalues of $A$. |
| $\lambda_{\text {max }}(A)\left(\lambda_{\text {min }}(A)\right)$ | Maximum (minimum resp.) eigenvalue value of a square matrix $A$. |
| $T(x)=\mathbf{O}(f(x))$ | Assume $T, f: \mathbb{R}_{+} \mapsto \mathbb{R}_{+}$. Then the previous notation means that there exist constants $c>0, x_{0} \geq 0$, such that $T(x) \leq$ $c f(x)$, for all $x \geq x_{0}$. |
| $T(x)=\boldsymbol{\Omega}(f(x))$ | Assume $T, f: \mathbb{R}_{+} \mapsto \mathbb{R}_{+}$. Then the previous notation means that there exist constants $c>0, x_{0} \geq 0$, such that $T(x) \geq$ $c f(x)$, for all $x \geq x_{0}$. |
| $T(x)=\boldsymbol{\Theta}(f(x))$ | This means that $T(x)=\mathbf{O}(f(x))$ and $T(x)=\boldsymbol{\Omega}(f(x))$. |
| $\mu_{k}$ | Barrier parameter of interior point method at iteration $k$. |
| $\tau_{k}$ | Centering parameter of interior point method at iteration $k$. |
| $T_{1} \wedge T_{2}$ | Logical AND between two logical statements $T_{1}$ and $T_{2}$. |
| $q(M)$ | The Rayleigh quotient of a symmetric matrix, defined as $q(M):=\left\{z \in \mathbb{R}\right.$, s.t. $z=\frac{x^{\top} M x}{x^{\top} x}$, for some $\left.x \in \mathbb{R}^{n}, x \neq 0\right\}$. |
| LP, QP, SDP | Linear, convex quadratic, positive semi-definite programming |
| IPM | Interior point method |
| PMM | Proximal method of multipliers |


| IP-PMM | Interior point-proximal method of multipliers |
| :--- | :--- |
| PDE | Partial differential equation |
| fMRI | Functional magnetic resonance imaging |

## Chapter 1

## Introduction

In this thesis we study regularized interior point methods (IPMs) for the solution of large-scale convex programming problems. In particular, we are concerned with the underlying theory of such algorithms as well as their efficient implementation. More specifically, we discuss two regularization strategies, the interpretation of which is drawn from the theory of proximal point methods. When such regularization strategies are incorporated within standard IPM schemes, one can show that the solution of the original problem is retrieved (without having to resort to finding a perturbed solution), while the associated linear algebra tasks, that constitute the computational bottleneck of IPMs, are simplified. In certain cases, we show that the polynomial complexity of IPMs can be maintained when the regularization is tuned appropriately. The latter was an open problem that was resolved for the first time as a byproduct of this study.

The biggest part of this thesis is focused on convex quadratic programming problems. Solution methods of such problems can serve as the main tool for the solution of general nonlinear convex problems. Nevertheless, polynomial convergence of a regularized IPM scheme is shown for linear positive semi-definite problems, in order to stress the generality of the approach. Global or local convergence analyses of such methods for general nonlinear (possibly non-convex) problems are not given in this thesis, as they have appeared before in the literature, and the reader is referred to the appropriate references for further details. Throughout all the chapters, we provide extensive numerical results on real-life applications, to showcase the efficiency and the robustness of the proposed methodologies. We consider a wide range of applications, such as problems arising from portfolio optimization, image processing, machine learning, as well as partial differential equation (PDE) optimization.

### 1.1 Convex programming

In this thesis, we consider problems of the following form

$$
\begin{equation*}
\min _{x} f(x), \quad \text { s.t. } A x=b, \quad x \geq 0_{n} \tag{CP}
\end{equation*}
$$

with $x \in \mathbb{R}^{n}, A \in \mathbb{R}^{m \times n}$, where $f: \mathbb{R}^{n} \rightarrow \mathbb{R}$ is a twice continuously differentiable convex function. Without loss of generality we assume that $m \leq n$. We can form the optimaity conditions of (CP) by introducing the Lagrangian function, using $y \in \mathbb{R}^{m}$ and $z \in \mathbb{R}^{n}, z \geq 0_{n}$, as the Lagrange multipliers for the equality and inequality constraints, respectively. Hence, we obtain:

$$
\begin{equation*}
\mathcal{L}(x, y, z):=f(x)-y^{\top}(A x-b)-z^{\top} x . \tag{1.1}
\end{equation*}
$$

Using the latter function, one can formulate the first-order optimality conditions (known as Karush-Kuhn-Tucker (KKT) conditions) for this problem. In particular, we define the vector $w=\left(x^{\top}, y^{\top}, z^{\top}\right)^{\top}$, and compute the gradient of $\mathcal{L}(w)$. Using $\nabla_{w} \mathcal{L}(w)$, as well as the complementarity conditions, we may define a function $F(w): \mathbb{R}^{2 n+m} \mapsto \mathbb{R}^{2 n+m}$, using which, we write the KKT conditions as follows:

$$
F(w):=\left[\begin{array}{c}
\nabla_{x} f(x)-A^{\top} y-z  \tag{1.2}\\
A x-b \\
X Z e_{n}
\end{array}\right]=\left[\begin{array}{c}
0_{n} \\
0_{m} \\
0_{n}
\end{array}\right], \quad(x, z) \geq\left(0_{n}, 0_{n}\right),
$$

where $e_{n}$ denotes the vector of ones of size $n$, while $X, Z \in \mathbb{R}^{n \times n}$ denote the diagonal matrices satisfying $X^{(i, i)}=x^{i}$ and $Z^{(i, i)}=z^{i}$, for all $i \in\{1, \ldots, n\}$.

### 1.1.1 A primal-dual interior point method

Primal-dual interior point methods are popular for simultaneously solving (CP) and its dual. As indicated in the name, primal-dual IPMs act on both primal and dual variables. There are numerous variants of IPMs and the reader is referred to [78] for an extended literature review. In this thesis, infeasible primaldual IPMs are discussed. Such methods are called infeasible because they allow intermediate iterates of the method to be infeasible for the primal-dual problems under consideration, in contrast to feasible IPMs, which require intermediate iterates to be strictly feasible.

Interior point methods handle the non-negativity constraints of the problem with logarithmic barriers in the objective. That is, at each iteration, we choose a barrier parameter $\mu$ and form the logarithmic barrier primal problem:

$$
\begin{equation*}
\min _{x}\left(f(x)-\mu \sum_{j=1}^{n} \ln x^{j}\right) \text { s.t. } A x=b, \tag{1.3}
\end{equation*}
$$

in which non-negativity constraints $x>0_{n}$ are implicit. We form the KKT optimality conditions of problem (1.3), by introducing the Lagrangian of the primal barrier problem:

$$
\mathcal{L}_{\mu}^{\mathrm{IPM}}(x, y):=f(x)-y^{\top}(A x-b)-\mu \sum_{j=1}^{n} \ln x^{j} .
$$

Equating the gradient of the previous function to zero, gives the following conditions:

$$
\begin{aligned}
\nabla_{x} \mathcal{L}_{\mu}^{\mathrm{IPM}}(x, y) & =\nabla_{x} f(x)-A^{\top} y-\mu X^{-1} e_{n} & & =0_{n} \\
\nabla_{y} \mathcal{L}_{\mu}^{\mathrm{IPM}}(x, y) & =A x-b & & =0_{m}
\end{aligned}
$$

Using the variable $z=\mu X^{-1} e_{n}$, the final conditions read as follows:

$$
\begin{aligned}
\nabla_{x} f(x)-A^{\top} y-z & =0_{n}, \\
A x-b & =0_{m}, \\
X Z e_{n}-\mu e_{n} & =0_{n} .
\end{aligned}
$$

At each IPM iteration, we want to approximately solve the following nonlinear system:

$$
F_{\tau, \mu}^{\mathrm{IPM}}(w):=\left[\begin{array}{c}
\nabla_{x} f(x)-A^{\top} y-z  \tag{1.4}\\
b-A x \\
X Z e_{n}-\tau \mu e_{n}
\end{array}\right]=\left[\begin{array}{c}
0_{n} \\
0_{m} \\
0_{n}
\end{array}\right],
$$

where $F_{\tau, \mu}^{\mathrm{IPM}}(w)=0_{2 n+m}$ is a slightly perturbed form of the previously presented optimality conditions. In particular, $\tau \in(0,1)$ is a centering parameter which determines how fast $\mu$ will be forced to decrease at the next IPM iteration. For $\tau=1$, we recover the barrier optimality conditions, while for $\tau=0$ we recover the initial problem's optimality conditions given in (1.2). The efficiency of the method depends heavily on the choice of $\tau$. In fact, various improvements of the traditional IPM schemes have been proposed in the literature, which solve the previous system for multiple carefully chosen values of the centering parameter $\tau$ and of the right hand side, at each IPM iteration. These are the so called predictor-corrector schemes, proposed for the first time in [119]. Various extensions of such methods have been proposed and analyzed in the literature (e.g. see [77, 122, 165] and references therein). However, for simplicity of exposition, we will follow the traditional approach, when theoretically analyzing the methods proposed in this thesis. In this case, $\tau$ is chosen heuristically at each iteration, and the previous system is solved only once. Nevertheless, predictor-corrector schemes will be incorporated in most implementations of the proposed methods.

In order to approximately solve the system $F_{\tau, \mu}^{\mathrm{IPM}}(w)=0_{2 n+m}$ for each value of $\mu$, the most common approach is to apply Newton method. Newton method is favored for systems of this form, due to the self-concordance of the function $\ln (\cdot)$. Briefly, a $p$-self-concordant function $g(x): C \mapsto \mathbb{R}$, where $C \subset \mathbb{R}^{n}$ is a nonempty convex set and $p>0$, is a three times continuously differentiable convex function that satisfies

$$
\left|\nabla_{x}^{3} g(x)[h, h, h]\right| \leq 2 p^{-\frac{1}{2}}\left(\nabla_{x}^{2} g(x)[h, h]\right)^{\frac{3}{2}}
$$

for all $x \in C$ and for all $h$ such that $x+h \in C$. In essence, a self-concordant function is always well approximated by a quadratic model, because the approximation error is bounded by its Hessian raised to the power $\frac{3}{2}$. For more details on this subject, the interested reader is referred to [128, Chapter 2].

At the beginning of the $k$-th iteration of the IPM, for $k \geq 0$, we have available
an iterate $w_{k}=\left(x_{k}^{\top}, y_{k}^{\top}, z_{k}^{\top}\right)^{\top}$, and a barrier parameter $\mu_{k}$, defined as $\mu_{k}=$ $\left(x_{k}^{\top} z_{k}\right) / n$. We choose a value for the centering parameter $\tau_{k} \in(0,1)$ and form the Jacobian of $F_{\tau_{k}, \mu_{k}}^{\mathrm{IPM}}(\cdot)$, evaluated at $w_{k}$. Then, the Newton direction is determined by solving the following system of equations:

$$
\left[\begin{array}{ccc}
-\nabla_{x}^{2} f\left(x_{k}\right) & A^{\top} & I  \tag{1.5}\\
A & 0_{m, m} & 0_{m, n} \\
Z_{k} & 0_{n, m} & X_{k}
\end{array}\right]\left[\begin{array}{c}
\Delta x_{k} \\
\Delta y_{k} \\
\Delta z_{k}
\end{array}\right]=-\left[\begin{array}{c}
-\nabla_{x} f\left(x_{k}\right)+A^{\top} y_{k}+z_{k} \\
A x_{k}-b \\
X_{k} Z_{k} e_{n}-\tau_{k} \mu_{k} e_{n}
\end{array}\right] .
$$

Notice that as $\mu_{k} \rightarrow 0$, the optimal solution of (1.3) converges to the optimal solution of (CP) and its dual. Polynomial convergence of such methods, for various classes of problems, has been established multiple times in the literature, even in cases where system (1.5) is solved approximately (see for example, among others, [181] for linear programming, [186] for linear complementarity problems, and [190] for linear positive semi-definite programming).

### 1.1.2 Proximal point methods

## Primal proximal point method

One possible method for solving the primal problem (CP), is the so called proximal point method. Given an arbitrary starting point $x_{0}$, the $k$-th iteration of the method is summarized by the following minimization problem:

$$
x_{k+1}=\arg \min _{x}\left\{f(x)+\frac{\mu_{k}}{2}\left\|x-x_{k}\right\|_{2}^{2}, \quad \text { s.t. } A x=b, \quad x \geq 0_{n}\right\}
$$

where $\left\{\mu_{k}\right\}$ is some non-increasing sequence of positive penalty parameters, and $x_{k}$ is the current estimate for an optimal solution of (CP). The use of $\mu_{k}$ is not a mistake; as will become clearer later, we intend to use the barrier parameter $\mu_{k}$ to control both the IPM and the proximal method. Notice that such an algorithm is not practical, since we have to solve a sequence of sub-problems of similar difficulty to that of (CP). Nevertheless, the proximal method contributes the term $\mu_{k} I_{n}$ to the Hessian of the objective, and hence the sub-problems are strongly convex. This method is known to achieve a linear convergence rate (and possibly superlinear if $\mu_{k} \rightarrow 0$ at a suitable rate), as shown in [75, 146, 147], among others, even in the case where the minimization sub-problems are solved approximately. Various extensions of this method have been proposed in the literature. For example, one could employ different penalty functions other than the typical 2 -norm penalty presented previously (e.g. see [39, 64, 95]).

Despite the fact that such algorithms are not practical, they have served as a powerful theoretical tool and a basis for other important methods. For an extensive review of the applications of standard primal proximal point methods, we refer the reader to [137].

## Dual proximal point method

It is a well-known fact, observed for the first time in [146], that the application of the proximal point method to the dual problem, is equivalent to the application of the augmented Lagrangian method to the primal problem, which was proposed for the first time in $[92,145]$. In view of the previous fact, we will present the derivation of the augmented Lagrangian method for the primal problem (CP), while having in mind that an equivalent reformulation of the model results in the proximal point method for its dual (see [29, Chapter 3.4.4] or [72, 146]).

We start by defining the augmented Lagrangian function of (CP). Given an arbitrary starting guess $y_{0}$ for the dual variable, the augmented Lagrangian function is defined at the $k$-th iteration as:

$$
\begin{equation*}
\mathcal{L}_{\mu_{k}}^{\mathrm{ALM}}\left(x ; y_{k}\right):=f(x)-y_{k}^{\top}(A x-b)+\frac{1}{2 \mu_{k}}\|A x-b\|_{2}^{2} \tag{1.6}
\end{equation*}
$$

where $\left\{\mu_{k}\right\}$ is some non-increasing sequence of positive penalty parameters, and $y_{k}$ is the current estimate of an optimal Lagrange multiplier vector. The augmented Lagrangian method (ALM) is summarized below:

$$
\begin{aligned}
& x_{k+1}=\arg \min _{x}\left\{\mathcal{L}_{\mu_{k}}^{\mathrm{ALM}}\left(x ; y_{k}\right), \quad \text { s.t. } x \geq 0_{n}\right\}, \\
& y_{k+1}=y_{k}-\frac{1}{\mu_{k}}\left(A x_{k+1}-b\right)
\end{aligned}
$$

Notice that the update of the Lagrange multiplier estimates can be interpreted as the application of the dual ascent method. A different type of update would be possible, however, the presented update is cheap and effective, due to the strong concavity of the proximal ("regularized") dual problem (since $\mu_{k}>0$ ). Convergence and iteration complexity of such methods have been established multiple times (see for example $[104,146]$ ). There is a vast literature on the subject of augmented Lagrangian methods. For a plethora of technical results and references, the reader is referred to [27]. For convergence results of various extended versions of the method, the reader is referred to [64].

It is worth noting here that a common issue, arising when using augmented Lagrangian methods, is that of the choice of the penalty parameter. To the author's knowledge, an optimal tuning for this parameter is still unknown.

## Proximal method of multipliers

In [146], the author presented, for the first time, the proximal method of multipliers (PMM). The method consists of applying both primal and dual proximal point methods for solving (CP) and its dual. For an arbitrary starting point ( $x_{0}, y_{0}$ ), and using the already defined augmented Lagrangian function, given in
(1.6), PMM can be summarized by the following iteration:

$$
\begin{align*}
& x_{k+1}=\arg \min _{x}\left\{\mathcal{L}_{\mu_{k}}^{\mathrm{ALM}}\left(x ; y_{k}\right)+\frac{\mu_{k}}{2}\left\|x-x_{k}\right\|_{2}^{2}, \text { s.t. } x \geq 0_{n}\right\},  \tag{1.7}\\
& y_{k+1}=y_{k}-\frac{1}{\mu_{k}}\left(A x_{k+1}-b\right) .
\end{align*}
$$

One can observe that at every iteration of the method, the primal problem that we have to solve is strongly convex, while its dual, strongly concave. As shown in [146], the addition of the term $\frac{\mu_{k}}{2}\left\|x-x_{k}\right\|_{2}^{2}$ in the augmented Lagrangian method does not affect its convergence rate, while ensuring better numerical behaviour of its iterates. An extension of this algorithm, allowing for the use of more general penalties, can be found in [64].

We can write (1.7) equivalently by making use of the maximal monotone operator $T_{\mathcal{L}}: \mathbb{R}^{n+m} \rightrightarrows \mathbb{R}^{n+m}$ associated to (CP) and its dual (see $[146,147]$ ), that is defined as:

$$
\begin{equation*}
T_{\mathcal{L}}(x, y):=\left\{(u, v): v \in \nabla_{x} f(x)-A^{\top} y+\partial \delta_{+}(x), u=A x-b\right\}, \tag{1.8}
\end{equation*}
$$

where $\delta_{+}(\cdot)$ is an indicator function defined as:

$$
\delta_{+}(x):= \begin{cases}\infty & \text { if } \exists j: x^{j}<0  \tag{1.9}\\ 0 & \text { otherwise }\end{cases}
$$

and $\partial(\cdot)$ denotes the sub-differential of a function. In the case under consideration, we have that (see [148, Section 23]):

$$
z \in \partial \delta_{+}(x) \Leftrightarrow\left\{\begin{array}{ll}
z^{j}=0 & \text { if } x^{j}>0, \\
z^{j} \leq 0 & \text { if } x^{j}=0 .
\end{array}, \forall j=\{1, \ldots, n\}\right.
$$

By convention, if there exists a component $j$ such that $x_{j}<0$, we have that $\partial \delta_{+}(x)=\{\emptyset\}$. Given a bounded pair $\left(x^{*}, y^{*}\right)$ such that $\left(0_{m}, 0_{n}\right) \in T_{\mathcal{L}}\left(x^{*}, y^{*}\right)$, we can retrieve a vector $z^{*} \in \partial \delta_{+}\left(x^{*}\right)$, using which, the triple $\left(x^{*}, y^{*},-z^{*}\right)$ is an optimal solution for (CP) and its dual. By letting:

$$
\begin{equation*}
P_{k}:=\left(I_{(m+n)}+\frac{1}{\mu_{k}} T_{\mathcal{L}}\right)^{-1}, \tag{1.10}
\end{equation*}
$$

we can express (1.7) as:

$$
\begin{equation*}
\left(x_{k+1}, y_{k+1}\right)=P_{k}\left(x_{k}, y_{k}\right), \tag{1.11}
\end{equation*}
$$

and it can be shown that $P_{k}$ is single-valued and non-expansive (see [147]).

### 1.1.3 Regularization in interior point methods

In the context of interior point methods, it is often beneficial to include regularization in order to improve the spectral properties of the system matrix in (1.5). For example, notice that if the constraint matrix $A$ is rank-deficient, then the matrix in (1.5) might not be invertible. The latter can be immediately addressed by the introduction of a dual regularization, say $\delta>0$, ensuring that $\operatorname{rank}\left(\left[A \mid \delta I_{m}\right]\right)=m$. For a detailed analysis of the effect of dual regularization on such systems, the reader is referred to [4]. On the other hand, the most common and efficient approach in practice is to eliminate the variables $\Delta z$ from system (1.5), and solve the following symmetric reduced (augmented) system instead:

$$
\left[\begin{array}{cc}
-\left(\nabla_{x}^{2} f\left(x_{k}\right)+\Theta_{k}^{-1}\right) & A^{\top}  \tag{1.12}\\
A & 0_{m, m}
\end{array}\right]\left[\begin{array}{c}
\Delta x_{k} \\
\Delta y_{k}
\end{array}\right]=\left[\begin{array}{c}
\nabla_{x} f\left(x_{k}\right)-A^{\top} y_{k}-\sigma_{k} \mu_{k} X_{k}^{-1} e \\
b-A x_{k}
\end{array}\right],
$$

where $\Theta_{k}=X_{k} Z_{k}^{-1}$. Since $X_{k} z_{k} \rightarrow 0_{n}$ close to optimality, one can observe that the matrix $\Theta_{k}$ will contain some very large and some very small elements. Hence, the matrix in (1.12) will be increasingly ill-conditioned, as we approach optimality. The introduction of a primal regularization, say $\rho>0$, can ensure that the matrix $\nabla_{x}^{2} f\left(x_{k}\right)+\Theta_{k}^{-1}+\rho I_{n}$ will have eigenvalues that are bounded away from zero, and hence a significantly better worst-case conditioning than that of $\nabla_{x}^{2} f\left(x_{k}\right)+\Theta_{k}^{-1}$. In other words, regularization is commonly employed in IPMs, as a means of improving robustness and numerical stability (see [1]). As we will discuss later, by introducing regularization in IPMs, one can also gain efficiency. This is because regularization transforms the matrix in (1.12) into a quasi-definite one. It is known that such matrices can be factorized efficiently (see [171]).

To produce a diagonal term in the $(2,2)$ block, Vanderbei added artificial variables to all the constraints [171]. Saunders and Tomlin [154, 155] achieved a similar result for the $(1,1)$ and $(2,2)$ blocks, by adding Tikhonov-type regularization terms to the original problem. In later works, these Tikhonov-type regularization methods were replaced by algorithmic regularization schemes. In particular, and in view of the previous discussion, one can observe that a very natural way of introducing primal regularization to problem (CP), is through the application of the primal proximal point method. Similarly, dual regularization can be incorporated through the application of the dual proximal point method. This is a well-known fact. The authors in [1] presented a primal-dual regularized IPM for convex quadratic programming, and interpreted this regularization as the application of the proximal point method. Subsequently, the authors in [72] developed a primal-dual regularized IPM, which applies PMM to solve (CP) (assuming that $f(\cdot)$ is convex quadratic), and a single IPM iteration is employed for approximating the solution of each PMM sub-problem. There, global convergence of the method was proved, under some assumptions.

Similar ideas have been applied in the context of IPMs for general non-linear programming problems. For example, the authors in [5] presented an interior point method combined with the augmented Lagrangian method, and proved
that, under some general assumptions, the method converges to a local optimum. A similar approach, with infeasibility detection capabilities as well as a local convergence analysis, can be found in [8].

In Chapter 2 of this thesis, we present a variation of the method proposed in [72] (again assuming that $f(\cdot)$ is a convex quadratic function), in which general non-diagonal regularization matrices are employed, as a means of further improving factorization efficiency. There, we also analyze (by means of a perturbation analysis) the effect of regularization in the eigenvalues of the original Newton matrices. By exploiting this analysis, we propose a minimum regularization value (which depends on the problem under consideration) that does not perturb the system matrix significantly, while ensuring numerical stability. Then, in Chapter 3 we present an interior point-proximal method of multipliers (IP-PMM), which is interpreted as a primal-dual regularized IPM suitable for solving convex quadratic programming problems. Under standard assumptions, we show that the method converges to an $\epsilon$-optimal solution in a polynomial number of steps, while global convergence also holds. We also derive an infeasibility detection mechanism, as a byproduct of the theory. In Chapter 4, we extend IP-PMM for linear positive semi-definite programming problems, and show that polynomial convergence still holds, even if one solves the associated Newton systems inexactly. We should mention that these polynomial results are the first to appear in the literature for regularized IPM schemes. Another important issue that is stressed in Chapters 3, 4 , concerns the tuning of the regularization parameters. In particular, we argue that it must be heavily reliant on the value of the barrier parameter, and that in theory (and in practice) the regularization parameters must always be chosen to be of the same order of magnitude as the barrier parameter $\mu_{k}$ of the IPM.

### 1.2 Iterative solution of linear systems

The main computational bottleneck of any optimization algorithm employing a variant of Newton method, arises from the solution of the associated linear systems of equations, the coefficient matrices of which are usually sparse. In particular, this is true for IPM solvers (e.g. see [2,52, 77, 119, 129, 181]), augmented Lagrangian solvers (e.g. see [7, 9, 108, 174, 189]), sequential quadratic programming solvers (e.g. see $[10,31,131]$ ), or any other (semismooth) Newtonbased solver (e.g. see [101, 103, 106, 157]). Such linear systems were traditionally solved by means of factorization methods (such as the Cholesky, or the $L D L^{\top}$ decomposition-see $[2,62]$ for an extensive discussion on direct methods for sparse systems). While such factorization schemes have been implemented very efficiently, and are extremely general (i.e. they do not depend on the problem under consideration), they often require excessive memory, and as a result cannot be employed to solve large-scale instances.

In light of the previous, there have been many attempts (well covered in the literature) concerned with the use of iterative solution techniques for the aforementioned linear systems. Such iterative methods do not require the factorization of the system matrix, and attempt to find a sequence of approximate solutions by means of matrix-vector products (which can be performed without the explicit
storage of the entire system matrix, thus allowing one to work in a matrix-free framework-e.g. as in [17, 79]). The most widely-used iterative methods are the so-called Krylov subspace solvers, and in this thesis we will focus only on such schemes. The reader is referred to [85, 110] (and the references therein), for an extensive review of Krylov subspace methods. In what follows, we briefly mention some of the most popular variants. When the linear system under consideration has a symmetric positive definite coefficient matrix, one often employs the conjugate gradient (CG) method ([93]), which is a short-recurrence symmetric solver. For indefinite symmetric systems, the symmetric LQ or the minimal residual (MINRES) method ([135]) can be used instead, which are also shortrecurrence symmetric solvers. Alternatively, one might employ non-symmetric solvers, such as LSQR ([136]), LSMR ([68]) or the generalized minimal residual (GMRES) method ([153]). GMRES (which is an extension of MINRES for nonsymmetric systems) is a long-recurrence solver, and as a consequences requires additional memory for each iteration. While it can be more general, it is more difficult to analyze (see [86]). On the other hand, LSQR and LSMR are based on bi-orthogonalization, and hence do not require more memory than MINRES. However, they do not have any tractable optimality property, thus making their convergence analysis especially difficult. As a result, in this thesis we will focus on symmetric solvers, and specifically on CG and MINRES.

Krylov solvers can be further accelerated by means of preconditioning, and there exists a rapidly-developing literature focusing on the construction as well as the analysis of such preconditioners, either for specific applications, or for general problems. The idea of preconditioning is surprisingly old (in fact, it can be traced back to the work of Gauss and of Jacobi, e.g. see [73, 96]; the term originates from the work of Turing in [169]), and is not restricted to Krylov subspace solvers. Despite its long history, preconditioning constitutes a highly active research area, which saw increasing applicability due to the rise in large-scale problems, for which direct methods fail to deliver solutions (see the discussions in [21, 24, 25, $59,60,85,110]$, and the references therein).

The preconditioners proposed in the literature can be divided into symmetric (e.g. see $[21,30,74,126,132,133,161]$ ), and non-symmetric ones (e.g. see [59, 94, 100, 126, 153]). In this thesis, and in particular in Chapter 5, we focus on the former (and in particular on symmetric positive definite preconditioners), which can be used within short-recurrence Krylov-subspace methods (with the latter being usable only within long-recurrence solvers). Specifically, given the saddle point systems that arise from the application of a regularized IPM to convex programming (i.e. primal-dual regularized versions of system (1.12)), we derive general-purpose preconditioners, usable within MINRES or CG. These preconditioners are based on sparsifications of the associated linear systems. Such sparsifications are not problem-dependent. In particular, we exploit the properties of the logarithmic barrier to make knowledgeable decisions concerning which parts of the linear system should be dropped. Furthermore, we also consider dropping dense parts of the linear system that produce dense factors during factorization. These sparsified system matrices are then factorized using standard direct methods. In light of the induced sparsity of these preconditioners, we expect that their factorization requires reasonable memory and computations, and
we show their success in accelerating the associated iterative methods both theoretically and numerically. We also discuss possible low-rank corrections of the preconditioned matrices, in order to further accelerate the iterative solvers.

### 1.3 Structure of the thesis

In this thesis, we outline and showcase the benefits of regularization in the context of interior point methods for convex programming problems. While all the chapters follow the same notational rules, we assume that the notation used within each chapter is self-contained, unless stated otherwise (e.g. the same letters might represent different things in different chapters).

In Chapter 2 we provide a well-known interpretation of regularization within IPMs. In particular, we discuss the interpretation of regularization as the application of an appropriate proximal point method. In turn, this allows us to deduce that regularization can be exact, i.e. the solution of the regularized model (or more accurately, sequence of models) is also a solution to the original problem. The rest of this chapter focuses on two aspects of regularization. Firstly, we provide a perturbation analysis to explore the effect of regularization on the problem data (i.e. how much these are perturbed), as well as a comprehensive spectral analysis, that shows the positive effects of regularization in terms of stability introduced in the associated linear systems. Secondly, we propose a technique that allows us to employ regularization as a means of improving efficiency of IPM solvers, and interpret this as the application of a generalized proximal point method.

Subsequently, in Chapters 3, 4, we focus on a uniform regularization scheme (interpreted as the application of the standard proximal method of multipliers). We show that given an appropriate tuning of the regularization parameters, one can design regularized IPMs, for a very wide range of convex problems, that are guaranteed to converge in polynomial time under standard assumptions employed in the IPM literature. We show that this polynomial convergence holds even if the associated linear systems are solved inexactly. These results state that the polynomial complexity of standard IPMs is not lost when a regularized IPM scheme is designed appropriately. Furthermore, we stress the fact that regularization introduces stability in the linear systems solved within standard IPMs. Thus, the resulting solvers are significantly more robust, and in certain cases, much more efficient. This observation is verified numerically on standard test sets.

In Chapter 5, we focus on the efficient solution of regularized saddle point systems arising from the application of a regularized IPM to an arbitrary convex problem. In particular, we propose and analyze several general-purpose preconditioners that can be used to accelerate symmetric Krylov subspace solvers for such systems. We note that the solution of saddle point systems is the main bottleneck within IPM implementations, and we numerically demonstrate how the use of inexact Newton method within IPMs can significantly improve the efficiency of the associated solvers. All of the preconditioning approaches developed in this chapter take advantage of the regularized nature of the saddle point systems. Indeed, we showcase another very important benefit of regularization within IPMs;
the resulting linear systems are significantly easier to precondition, compared to their non-regularized counterparts.

In Chapter 6, by combining the developments of all previous chapters, we derive several application-specific regularized IPM implementations, for the solution of sparse approximation problems, arising in portfolio optimization, classification of data coming from functional magnetic resonance imaging, restoration of images corrupted by Poisson noise, as well as classification via logistic regression. In each of these applications, we compare the proposed specialized solver with state-of-the-art specialized first-order methods, that are traditionally used to solve such problems in the literature. We show that regularized IPMs are more robust, and in certain cases more efficient than the best first-order methods available, if their linear algebra phase is tuned appropriately.

Finally, in Chapter 7 we deliver our conclusions, as well as discuss several open problems and potential future research directions in the field of regularization for interior point methods.

### 1.4 Summary

The contents of this thesis are based on the following publications and preprints:

## Chapter 2

- S. Pougkakiotis and J. Gondzio. "Dynamic non-diagonal regularization in interior point methods for linear and convex quadratic programming", Journal of Optimization Theory and Applications, 181 (3), 905-945, 2019, 10.1007/s10957-019-01491-1, [141].


## Chapter 3

- S. Pougkakiotis and J. Gondzio. "An interior point-proximal method of multipliers for convex quadratic programming", Computational Optimization and Applications, 78 (2), 307-351, 2021, 10.1007/s10589-020-00240-9, [142].


## Chapter 4

- S. Pougkakiotis and J. Gondzio. "An interior point-proximal method of multipliers for linear positive semi-definite programming", Journal of Optimization Theory and Applications, 2021, 10.1007/s10957-021-01954-4, [143].


## Chapter 5

- L. Bergamaschi, J. Gondzio, Á. Martínez, J. W. Pearson and S. Pougkakiotis. "A new preconditioning approach for an interior point-proximal method of multipliers for linear and convex quadratic programming", Numerical Linear Algebra with Applications, 28 (4), e2361, 2021, 10.1002/nla.2361, [23].
- J. Gondzio, S. Pougkakiotis and J. W. Pearson. "General-purpose preconditioning for regularized interior point methods", arXiv:2107.06822, 2021, [82] (submitted to COAP, 14 July 2021).


## Chapter 6

- V. De Simone, D. di Serafino, J. Gondzio, S. Pougkakiotis and M. Viola. "Sparse approximations with interior point methods", arXiv:2102.13608, 2021, [52] (to appear in SIAM Review, accepted: 24 November 2021).

Permission from co-authors to include parts of these works in the thesis has been granted to the author.

During my PhD studies, I also co-authored the following paper, which was not used within this thesis:

- S. Pougkakiotis, J. W. Pearson, S. Leveque and J. Gondzio. "Fast solution methods for convex quadratic optimization of fractional differential equations", SIAM Journal on Matrix Analysis and Applications, 41 (3), 1443-1476, 2020, 10.1137/19M128288X, [144].

In [144], we present numerical methods suitable for solving convex quadratic fractional differential equation (FDE) constrained optimization problems, with box constraints on the state and/or control variables. We develop an alternating direction method of multipliers (ADMM) framework, which uses preconditioned Krylov subspace solvers for the resulting subproblems. The latter allows us to tackle a range of partial differential equation (PDE) optimization problems with box constraints, posed on space-time domains, that were previously out of the reach of state-of-the-art preconditioners. In particular, by making use of the powerful generalized locally Toeplitz (GLT) sequences theory, we show that any existing GLT structure present in the problem matrices is preserved by ADMM, and we propose some preconditioning methodologies that could be used within the solver, to demonstrate the generality of the approach. Focusing on convex quadratic programs with time-dependent 2-dimensional FDE constraints, we derive multilevel circulant preconditioners, which may be embedded within Krylov subspace methods, for solving the ADMM subproblems. Discretized versions of FDEs involve large dense linear systems. In order to overcome this difficulty, we design a recursive linear algebra, which is based on the fast Fourier transform (FFT). We manage to keep the storage requirements linear, with respect to the grid size $N$, while ensuring an $\boldsymbol{O}(N \log N)$ computational complexity per iteration of the Krylov solver. We implement the proposed method and demonstrate its scalability, generality, and efficiency through a series of experiments over different setups of the FDE optimization problem.

## Chapter 2

## Dynamic Regularization in IPMs for Convex QP

### 2.1 Introduction

In this chapter, we are concerned with finding the solution of linear and convex quadratic programming problems, using an infeasible primal-dual interior point method. As already mentioned in Chapter 1, interior point methods deal with the inequality constraints of the problem by introducing logarithmic barriers in the objective, which penalize when any of the inequality constraints is close to being violated. At each iteration, the optimality conditions of the barrier problems are formed and one (or a few) steps of Newton method are applied to them. For an extended literature review on interior point methods we refer the interested reader to [78].

Most implementations transform the Newton system into a symmetric indefinite system of linear equations, which when solved, determines the Newton direction. The latter constitutes the main computational effort and challenge for IPMs. At every iteration of the method, the system matrix as well as the right hand side change. There are three main reasons indicating why solving such a system can be challenging. The most obvious one, is that the dimension of such systems can be very large, which makes the task of solving them expensive in terms of processing time and memory requirements. A second important challenge, inherent in interior point methods, is that as the algorithm approaches optimality, the systems that we have to solve become increasingly ill-conditioned. Finally, a rank deficient constraint matrix can result in a singular Newton system matrix. It is well known that the latter two difficulties can be addressed by the use of some regularization technique, at the expense of solving a perturbed problem, e.g. [4].

Such regularization techniques, embedded in the interior-point framework for solving linear and convex quadratic programming problems, have been previously proposed in the literature. For example, in [1], a dynamic primal-dual regularization for interior point methods was derived. The authors solve a slightly altered symmetric indefinite system, to which a diagonal perturbation (regularization) has been introduced. This perturbation transforms the symmetric indefinite ma-
trix into a quasi-definite one. It is proved in [171] that such matrices are strongly factorizable. Hence, the regularized system can be factorized efficiently. The authors interpreted these regularization matrices as adding proximal terms to the primal and dual objective functions. The values of these perturbations are chosen dynamically during the factorization of the system matrix, where potentially unstable pivots are regularized stronger (using some pre-specified "large" regularization value), while safer ones are almost not regularized at all. In [72], based on this proximal point interpretation given in [1], the authors proposed a primal-dual pair of regularized models, where the duality correspondence arises by setting the regularization variables as proximal terms. They observed that for specific parameter values, this primal-dual regularized model is exact, that is it yields an optimal solution which is also an optimal solution of the respective non-regularized primal-dual pair. There, the authors introduced two uniform diagonal regularization matrices whose values were tuned experimentally over a variety of problems. A similar regularization was also used in [154]. It is worth mentioning that similar ideas have also been applied in the context of general non-linear optimization problems (e.g. see [5, 10], and the references therein).

In this chapter, we are taking a different approach. We observe that when an IPM progresses and approaches optimality, significant part of the primal-dual variables approaches zero fast and hence becomes negligible. Yet it is not straightforward how the algorithm might exploit this feature. The proposed method attempts to do so. The method dynamically chooses a suitable regularization for the symmetric indefinite system and effectively "annihilates" the effects of those parts of it, which do not contribute important information to the computation of the Newton direction. The proposed technique involves non-diagonal regularization matrices. However, their non-diagonal terms are only implicit; they do not need to be computed because they are immediately cancelled by other terms present in the linear system. Hence, the effect of adding such non-diagonal regularization is making the Newton system more sparse and therefore easier. In contrast to other previously developed approaches, this regularization is dynamically tuned based on the problem properties. We develop an approach which attempts to capture the needs of an arbitrary problem and regularize its system matrix accordingly. This alleviates the problem of finding specific regularization values that work well over a variety of problems. In general, the proposed approach is very conservative and regularizes the system as little as possible, while ensuring numerical stability.

The rest of this chapter is organized as follows. In Section 2.2, we present the adopted model, based on which, we define our regularization matrices, firstly for linear and then for convex quadratic programming problems. For both cases, we provide arguments indicating why the proposed dynamic tuning of the regularization matrices is expected to introduce a controlled perturbation to the problem. In Section 2.3 we provide a spectral analysis, which shows the effect of the proposed regularization and gives specific bounds for the eigenvalues of the regularized system matrix. In Section 2.4, we provide the algorithmic scheme along with some implementation details and numerical results, and finally in Section 2.5 we derive some conclusions.

### 2.2 Exact primal-dual regularization

### 2.2.1 Problem formulation

We consider the following primal-dual pair of convex quadratic programming problems in the standard form:

$$
\begin{gather*}
\min _{x}\left(c^{\top} x+\frac{1}{2} x^{\top} Q x\right), \text { s.t. } A x=b, \quad x \geq 0_{n},  \tag{CQP}\\
\max _{x, y, z}\left(b^{\top} y-\frac{1}{2} x^{\top} Q x\right), \quad \text { s.t. }-Q x+A^{\top} y+z=c, \quad z \geq 0_{n}, \tag{CQD}
\end{gather*}
$$

where $c, x, z \in \mathbb{R}^{n}, b, y \in \mathbb{R}^{m}, A \in \mathbb{R}^{m \times n}, Q \succeq 0_{n, n} \in \mathbb{R}^{n \times n}$. Without loss of generality, we assume that $m \leq n$. Note that if $Q=0_{n, n}$, (CQP)-(CQD) is a primal-dual pair of linear programming problems. If the problems under consideration are feasible, it can easily be verified that there exists an optimal primal-dual triple ( $x, y, z$ ) satisfying the Karush-Kuhn-Tucker (KKT) optimality conditions for this primal-dual pair (see for example [28, Prop. 2.3.4]).

Our model is based on the developments in [1, 72, 154]. More specifically, by applying a generalized primal-dual proximal point method on (CQP), similar to [10, 72], one can get the following pair of primal-dual regularized problems:

$$
\begin{align*}
& \min _{x, r}\left(c^{\top} x+\frac{1}{2} x^{\top} Q x+\frac{1}{2}\|r+\tilde{y}\|_{R_{d}}^{2}+\frac{1}{2}\|x-\tilde{x}\|_{R_{p}}^{2}\right)  \tag{r}\\
& \text { s.t. } A x+R_{d} r=b, x \geq 0_{n}, \\
& \max _{x, y, z, s}\left(b^{\top} y-\frac{1}{2} x^{\top} Q x-\frac{1}{2}\|y-\tilde{y}\|_{R_{d}}^{2}-\frac{1}{2}\|s+\tilde{x}\|_{R_{p}}^{2}\right)  \tag{r}\\
& \text { s.t. }-Q x-R_{p} s+A^{\top} y+z=c, z \geq 0_{n},
\end{align*}
$$

where $s \in \mathbb{R}^{n}, r \in \mathbb{R}^{m}$ are auxiliary variables introduced from the primal-dual application of the proximal point method and, $R_{p} \succeq 0_{n, n} \in \mathbb{R}^{n \times n}, R_{d} \succeq 0_{m, m} \in$ $\mathbb{R}^{m \times m}$ are the primal and dual regularization matrices respectively, that will be specified later. The duality correspondence follows after taking $r=y-\tilde{y}$ and $s=x-\tilde{x}$, where $\tilde{y}$ and $\tilde{x}$ are estimates of the dual and primal solutions $y^{*}, x^{*}$ respectively. Of course $R_{p}=0_{n, n}, R_{d}=0_{m, m}$ recovers the initial pair (CQP)(CQD). In [72], the authors observe that this pair of regularized problems is exact under some conditions on the estimates $\tilde{x}, \tilde{y}$. In such a case, an optimal solution of $\left(\mathrm{CQP}_{r}\right)-\left(\mathrm{CQD}_{r}\right)$ is also an optimal solution of (CQP)-(CQD). For more information about exactness of regularization, we refer the interested reader to [71].

In [5, 10, 72], models similar to $\left(\mathrm{CQP}_{r}\right)-\left(\mathrm{CQD}_{r}\right)$ are used, restricted however in the case where $R_{p}=\rho I_{n}$ and $R_{d}=\delta I_{m}$, for some positive values $\delta, \rho$. It is a well known fact, proved for the first time in [146], that these regularization schemes can be interpreted as the primal-dual application of the standard proximal point method. However, our model does not specify the structure of the regularization
matrices $R_{p}, R_{d}$. The only requirement is that these matrices are positive semidefinite. As we commented previously, this model can be interpreted as the application of a generalized primal-dual proximal point method. Such methods, instead of adding the typical 2-norm in the objective function, make use of the so called $D$-functions. In fact, one could easily verify that any elliptic norm (defined by an arbitrary positive definite matrix) satisfies the conditions, given in [39, 95], for being a D-function. In other words, our algorithm adds an elliptic norm in the objective, instead of the typical 2-norm. The focus of this chapter however, prevents us from going deeper into these matters. For more about proximal point methods, we refer the reader to $[39,75,95,137,146]$, and the references therein.

### 2.2.2 The Newton system

In order to solve the problems presented in the previous subsection, using interior point methods, we proceed by replacing the non-negativity constraints with logarithmic barriers in the objective. In view of the previous, we obtain the following primal-dual regularized barrier problems:

$$
\begin{align*}
& \min _{x, r}\left(c^{\top} x+\frac{1}{2} x^{\top} Q x+\frac{1}{2}\|r+\tilde{y}\|_{R_{d}}^{2}+\frac{1}{2}\|x-\tilde{x}\|_{R_{p}}^{2}-\mu \sum_{j=1}^{n} \ln \left(x_{j}\right)\right)  \tag{2.1}\\
& \text { s.t. } A x+R_{d} r=b, \\
& \max _{x, y, z, s}\left(b^{\top} y-\frac{1}{2} x^{\top} Q x-\frac{1}{2}\|y-\tilde{y}\|_{R_{d}}^{2}-\frac{1}{2}\|s+\tilde{x}\|_{R_{p}}^{2}+\mu \sum_{j=1}^{n} \ln \left(z_{j}\right)\right)  \tag{2.2}\\
& \text { s.t. }-Q x-R_{p} s+A^{\top} y+z=c .
\end{align*}
$$

Forming the Lagrangian of the primal barrier problem, we get:

$$
\begin{align*}
\mathcal{L}_{\tilde{x}, \tilde{y}, \mu}(x, y, r)= & c^{\top} x+\frac{1}{2} x^{\top} Q x+\frac{1}{2}\|r+\tilde{y}\|_{R_{d}}^{2}+ \\
& +\frac{1}{2}\|x-\tilde{x}\|_{R_{p}}^{2}-y^{\top}\left(A x+R_{d} r-b\right)-\mu \sum_{j=1}^{n} \ln \left(x_{j}\right) . \tag{2.3}
\end{align*}
$$

Following Section 1.1.1, we form the first order optimality conditions as:

$$
\begin{aligned}
\nabla_{x} \mathcal{L}_{\tilde{x}, \tilde{y}, \mu}(x, y, r) & =c+Q x+R_{p}(x-\tilde{x})-A^{\top} y-\mu X^{-1} e_{n} & & =0_{n} \\
\nabla_{y} \mathcal{L}_{\tilde{x}, \tilde{y}, \mu}(x, y, r) & =A x+R_{d} r-b & & =0_{m} \\
\nabla_{r} \mathcal{L}_{\tilde{x}, \tilde{y}, \mu}(x, y, r) & =R_{d}(r+\tilde{y})-R_{d} y & & =0_{m} .
\end{aligned}
$$

By the optimality conditions of the dual barrier problem, we further obtain:

$$
\begin{aligned}
R_{p} x-R_{p}(s+\tilde{x}) & =0_{n}, \\
X Z e_{n} & =\mu e_{n} .
\end{aligned}
$$

We write the optimality conditions in the form of a function, $F_{\tilde{x}_{k}, \tilde{y}_{k}, \mu}(w)$ :
$\mathbb{R}^{3 n+2 m} \rightarrow \mathbb{R}^{3 n+2 m}$ and we want to approximately solve:

$$
F_{\tilde{x}_{k}, \tilde{y}_{k}, \mu_{k}}(w):=\left[\begin{array}{c}
c+Q x+R_{p, k} s-A^{\top} y-z  \tag{2.4}\\
R_{d, k}\left(r+\tilde{y}_{k}\right)-R_{d, k} y \\
R_{p, k} x-R_{p, k}\left(s+\tilde{x}_{k}\right) \\
A x+R_{d, k} r-b \\
X Z e
\end{array}\right]=\left[\begin{array}{c}
0_{n} \\
0_{m} \\
0_{n} \\
0_{m} \\
\tau_{k} \mu_{k} e_{n}
\end{array}\right],
$$

at each IPM iteration $k$, where $w=(x, r, s, y, z), \mu_{k}>0$ is the barrier parameter and $\tau_{k} \in(0,1)$ is a centering parameter. We want to force $\mu_{k} \rightarrow 0$, since then, the solution of this system leads to the solution of $\left(\mathrm{CQP}_{r}\right)-\left(\mathrm{CQD}_{r}\right)$. Notice that $\left(\mathrm{CQP}_{r}\right)-\left(\mathrm{CQD}_{r}\right)$ is parametrized by the estimates $\tilde{x}_{k}$ and $\tilde{y}_{k}$. As observed in [72], if these estimates are close enough to some optimal solution of (CQP)(CQD), then an optimal solution of $\left(\mathrm{CQP}_{r}\right)-\left(\mathrm{CQD}_{r}\right)$ is also an optimal solution of (CQP)-(CQD). Following the developments in [1, 5, 10, 72], for proximal point methods, we update the estimates of $x^{*}, y^{*}$ as $\tilde{x}_{k}=x_{k}, \tilde{y}_{k}=y_{k}$ (i.e. we allow only a single iteration of IPM for each proximal point sub-problem; notice that the proposed regularization benefits from such an update. Nevertheless, an alternative approach will be presented in the following chapter, which in turn will allow us to derive a polynomially convergent regularized IPM). Next, Newton method is applied to the mildly non-linear system (2.4). After evaluating the Jacobian of $F_{\tilde{x}_{k}, \tilde{y}_{k}, \mu_{k}}\left(w_{k}\right)$, the Newton direction is determined at each IPM iteration by solving a system of the following form:

$$
\left[\begin{array}{ccccc}
Q & 0_{n, m} & R_{p, k} & -A^{\top} & -I_{n}  \tag{2.5}\\
0_{m, n} & R_{d, k} & 0_{m, n} & -R_{d, k} & 0_{m, n} \\
R_{p, k} & 0_{n, m} & -R_{p, k} & 0_{n, m} & 0_{n, n} \\
A & R_{d, k} & 0_{m, n} & 0_{m, m} & 0_{m, n} \\
Z_{k} & 0_{n, m} & 0_{n, n} & 0_{n, m} & X_{k}
\end{array}\right]\left[\begin{array}{c}
\Delta x \\
\Delta r \\
\Delta s \\
\Delta y \\
\Delta z
\end{array}\right]=\left[\begin{array}{c}
A^{\top} y_{k}+z_{k}-c-Q x_{k}-R_{p, k} s_{k} \\
R_{d, k} y_{k}-R_{d, k}\left(r_{k}+\tilde{y}_{k}\right) \\
R_{p, k}\left(s_{k}+\tilde{x}_{k}\right)-R_{p, k} x_{k} \\
b-A x_{k}-R_{d, k} r_{k} \\
\tau_{k} \mu_{k} e_{n}-X_{k} Z_{k} e_{n}
\end{array}\right],
$$

where $0_{m, n}$ is the zero matrix of dimensions $m \times n$, while $R_{p, k} \in \mathbb{R}^{n \times n}, R_{d, k} \in$ $\mathbb{R}^{m \times m}$ are the primal and dual regularization matrices, respectively. Notice that the matrices $X, Z, R_{p}$ and $R_{d}$ all depend on the iteration $k$ of the algorithm. Once the Newton direction $\Delta w=(\Delta x, \Delta r, \Delta s, \Delta y, \Delta z)$ is computed, the algorithm chooses a step-length $a_{k} \in(0,1)$ and sets the new iterate to $w_{k+1}=w_{k}+a_{k} \Delta w$. In order to compute the Newton direction efficiently, we want to eliminate some variables of (2.5). Since we set $\tilde{y}_{k}=y_{k}$, the second block equation of (2.5) gives:

$$
R_{d, k} \Delta r-R_{d, k} \Delta y=-R_{d, k}\left(r_{k}+y_{k}\right)+R_{d, k} y_{k}
$$

and if $R_{d, k} \succ 0_{m, m}$, we get the following relation:

$$
\begin{equation*}
\Delta y=r_{k}+\Delta r \tag{2.6}
\end{equation*}
$$

Similarly, the third block equation of (2.5) (substituting $\tilde{x}_{k}=x_{k}$ ), yields:

$$
R_{p, k} \Delta x-R_{p, k} \Delta s=R_{p, k}\left(s_{k}+x_{k}\right)-R_{p, k} x_{k}
$$

and if $R_{p, k} \succ 0_{n, n}$ we have that:

$$
\begin{equation*}
\Delta x=s_{k}+\Delta s \tag{2.7}
\end{equation*}
$$

Note that we always use either $R_{d, k} \succ 0_{m, m}$ or $R_{d, k}=0_{m, m}$ and similarly, either $R_{p, k} \succ 0_{n, n}$ or $R_{p, k}=0_{n, n}$. Hence, the previous two relations are either welldefined or absent. Using (2.6) and (2.7) to eliminate $\Delta r$ and $\Delta s$, respectively, we can reduce (2.5) to the following system:

$$
\left[\begin{array}{ccc}
-\left(Q+R_{p, k}\right) & A^{\top} & I_{n}  \tag{2.8}\\
A & R_{d, k} & 0_{m, n} \\
Z_{k} & 0_{n, m} & X_{k}
\end{array}\right]\left[\begin{array}{c}
\Delta x \\
\Delta y \\
\Delta z
\end{array}\right]=\left[\begin{array}{c}
c+Q x_{k}-A^{\top} y_{k}-z_{k} \\
b-A x_{k} \\
\tau_{k} \mu_{k} e_{n}-X_{k} Z_{k} e_{n}
\end{array}\right] .
$$

Next, we proceed by eliminating $\Delta z$. For that purpose, we have from the third row of (2.8) that:

$$
\begin{equation*}
\Delta z=-X_{k}^{-1} Z_{k} \Delta x-Z_{k} e_{n}+\tau_{k} \mu_{k} X^{-1} e_{n} . \tag{2.9}
\end{equation*}
$$

Substituting (2.9) into the first row of (2.8), we obtain the following reduced symmetric system (so called augmented system):

$$
\left[\begin{array}{cc}
-\left(Q+\Theta_{k}^{-1}+R_{p, k}\right) & A^{\top}  \tag{2.10}\\
A & R_{d, k}
\end{array}\right]\left[\begin{array}{c}
\Delta x \\
\Delta y
\end{array}\right]=\left[\begin{array}{c}
c+Q x_{k}-A^{\top} y_{k}-\tau_{k} \mu_{k} X_{k}^{-1} e_{n} \\
b-A x_{k}
\end{array}\right],
$$

where $\Theta_{k}=X_{k} Z_{k}^{-1}$. In the case of linear programming ( $Q=0_{n, n}$ ) or when solving quadratic separable problems (in which case $Q$ is diagonal), it may be beneficial to further eliminate $\Delta x$ from (2.10), which will end up at the so called normal equations. However, one should note that this is not a good idea when it comes to general convex quadratic programming problems, since pivoting on the $(1,1)$ block of $(2.10)$ could result in a dense system, even in cases where both $A$ and $Q$ are sparse. Having said that, we can eliminate $\Delta x$ by looking at the first block equation of (2.10), which gives:

$$
\begin{equation*}
\Delta x=\left(Q+\Theta_{k}^{-1}+R_{p, k}\right)^{-1}\left(A^{\top} \Delta y-c-Q x_{k}+A^{\top} y_{k}+\tau_{k} \mu_{k} X_{k}^{-1} e_{n}\right), \tag{2.11}
\end{equation*}
$$

and by substituting (2.11) into the second row of (2.10), we get the normal equations:

$$
\begin{equation*}
\left[A\left(Q+\Theta_{k}^{-1}+R_{p, k}\right)^{-1} A^{\top}+R_{d, k}\right] \Delta y=\xi \tag{2.12}
\end{equation*}
$$

where

$$
\xi=b-A x_{k}+A\left(Q+\Theta_{k}^{-1}+R_{p, k}\right)^{-1}\left(c+Q x_{k}-A^{\top} y_{k}-\tau_{k} \mu_{k} X_{k}^{-1} e_{n}\right),
$$

in which the system matrix is symmetric and positive definite.
The proposed model differs from the one derived in [72] in that it allows the use of general positive definite regularization matrices. For example, if $R_{p, k}, R_{d, k}$ are non-diagonal matrices, then this would amount to the primal and dual application of a generalized proximal point method that adds an elliptic norm in the objective, instead of the typical 2-norm that is employed in standard proximal
point methods. Notice that at every iteration of the algorithm, $R_{p}, R_{d}, \tilde{x}$ and $\tilde{y}$ are updated. In other words, $\left(\mathrm{CQP}_{r}\right)-\left(\mathrm{CQD}_{r}\right)$ represents a sequence of subproblems. At every such sub-problem, we apply a single iteration of the interior point method. How $R_{p}$ and $R_{d}$ are updated will be presented in the following subsection.

### 2.2.3 The regularization matrices

As IPM approaches optimality, the diagonal matrix $\Theta_{k}$ contains elements that converge to zero and others that diverge to infinity. This is because $\mu_{k} \rightarrow 0$ and we force the complementarity conditions to be approximately satisfied ( $X_{k} Z_{k} e \approx$ $\tau_{k} \mu_{k} e_{n}$ ). As a consequence, the matrices in (2.10) and (2.12) become extremely ill-conditioned. On top of that, it is often the case due to modelling choices, that the constraint matrix $A$ is not of full row rank, which makes the system matrices singular. It is well known, as shown by Armand and Benoist [4], that both these problems can be addressed with the use of regularization. The most common approach in the literature, is the addition of two diagonal regularization matrices, say $R_{p}, R_{d}$, whose values are tuned experimentally over a variety of problems ([1, 4, 72, 154]).

Roughly speaking, the goals of a regularization method for IPMs are ([1, 4, 5, 6, 10, 154]):

1. to improve the spectral properties of the matrices in (2.10) and (2.12),
2. without significantly perturbing the previous systems,
3. while preserving the sparsity of the problem and the computational efficiency of the method.

To the best of our knowledge, most of the regularization methods in literature manage to achieve the first and the third regularization goals, failing however to achieve the second goal with certainty. This is the case since these regularization methods are tuned experimentally. Hence, they do not rely on the properties of the problem itself, and as a consequence, such regularization values can only be good for some problems and poor for others. The proposed method takes a different approach, by introducing two non-diagonal regularization matrices $R_{p}$ and $R_{d}$, which are tuned based on the properties of the problem. Of course one could argue that this may disturb the sparsity and as a consequence the computational efficiency of the method, however, these non-diagonal matrices are created implicitly. As we will show later, not only the sparsity is preserved, but in fact it is improved.

As we already mentioned, as IPM approaches optimality, the matrix $\Theta_{k}$ contains some very large and some very small elements. The proposed regularization exploits this inherent feature of the method and splits the columns of the problem
matrix in two sets, say $\mathcal{N}$ and $\mathcal{B}$ such that:

$$
\begin{aligned}
& \forall j \in \mathcal{N}: x_{k}^{j} \rightarrow 0, z_{k}^{j} \rightarrow \hat{z}^{j}>0 \Rightarrow(\Theta)_{k}^{(j, j)}=\frac{x_{k}^{j}}{z_{k}^{j}} \approx \frac{x_{k}^{j} z_{k}^{j}}{\left(z_{k}^{j}\right)^{2}}=\mathbf{O}\left(\mu_{k}\right), \\
& \forall j \in \mathcal{B}: x_{k}^{j} \rightarrow \hat{x}^{j}>0, z_{k}^{j} \rightarrow 0 \Rightarrow(\Theta)_{k}^{(j, j)}=\frac{x_{k}^{j}}{z_{k}^{j}} \approx \frac{\left(x_{k}^{j}\right)^{2}}{x_{k}^{j} z_{k}^{j}}=\mathbf{O}\left(\mu_{k}^{-1}\right),
\end{aligned}
$$

where $|\mathcal{N}|=n_{1}$ and $|\mathcal{B}|=n_{2}$, with $n_{1}+n_{2}=n$. Notice that the previous splitting captures all the columns only if the method converges to a strictly complementary solution (that is the limit point satisfies: $\hat{x}^{\top} \hat{z}=0$ and $\hat{x}^{j}+\hat{z}^{j}>0, \forall j$ ). In the quadratic programming case, a strictly complementary solution may not exist. Hence, there might exist some indices $j \subseteq\{1, \cdots, n\}$ for which: $x_{k}^{j} \rightarrow 0$ and $z_{k}^{j} \rightarrow$ 0 . In such a case, it is unknown whether the value of $\Theta_{k}^{(j, j)}$ will be small or large. We can assume, without loss of generality, that any such indices will be classified as elements of $\mathcal{B}$ (although in practice this would depend on the value of $\Theta_{k}^{(j, j)}$, as we will show later). Of course for the case of linear programming ( $Q=0_{n, n}$ ), it is a well-known fact (see for example [179]) that a strictly complementary solution always exists, if the problem is feasible. Moreover, as shown in [89, 118], primaldual IPMs converge to such an optimal solution. If a strictly complementary solution exists for the quadratic programming case, it is shown in [90], that an infeasible primal-dual IPM which reduces the constraints violation at the same rate as $\mu_{k}$ is reduced, produces iterates that converge to a strictly complementary solution.

In what follows, we present the construction of the regularization for the case of linear programming and then we suggest an extension for convex quadratic programming.

## Linear programming

For the case of linear programming we employ a dual regularization, that is, in (2.5) we set $R_{p}=0_{n, n}$ and only use $R_{d} \succ 0_{m, m}$ to improve the spectral properties of the problem. For the rest of this section, we drop the iteration indicator $k$ from any matrix that depends on it, in order to simplify the notation. Given this set-up, and by permuting the columns so that the first $n_{1}$ of them correspond to indices in $\mathcal{N}$ while the remaining correspond to indices in $\mathcal{B}$, the augmented system in (2.10) takes the form:

$$
\left[\begin{array}{ccc}
-\left(\Theta^{\mathcal{N}}\right)^{-1} & 0_{n_{1}, n_{2}} & \left(A^{\mathcal{N}}\right)^{\top}  \tag{2.13}\\
0_{n_{2}, n_{1}} & -\left(\Theta^{\mathcal{B}}\right)^{-1} & \left(A^{\mathcal{B}}\right)^{\top} \\
A^{\mathcal{N}} & A^{\mathcal{B}} & R_{d}
\end{array}\right]\left[\begin{array}{c}
\Delta x^{\mathcal{N}} \\
\Delta x^{\mathcal{B}} \\
\Delta y
\end{array}\right]=\left[\begin{array}{c}
c^{\mathcal{N}}-\left(A^{\mathcal{N}}\right)^{\top} y_{k}-\tau_{k} \mu_{k}\left(X^{\mathcal{N}}\right)^{-1} e_{|\mathcal{N}|} \\
c^{\mathcal{B}}-\left(A^{\mathcal{B}}\right)^{\top} y_{k}-\tau_{k} \mu_{k}\left(X^{\mathcal{B}}\right)^{-1} e_{|\mathcal{B}|} \\
b-A x_{k}
\end{array}\right],
$$

where $A^{\mathcal{N}} \in \mathbb{R}^{m \times n_{1}}$ and $A^{\mathcal{B}} \in \mathbb{R}^{m \times n_{2}}$. Pivoting on the first $n_{1}$ columns of (2.13), gives the partially reduced augmented system:

$$
\begin{gather*}
{\left[\begin{array}{cc}
-\left(\Theta^{\mathcal{B}}\right)^{-1} & \left(A^{\mathcal{B}}\right)^{\top} \\
A^{\mathcal{B}} & A^{\mathcal{N}} \Theta^{\mathcal{N}}\left(A^{\mathcal{N}}\right)^{\top}+R_{d}
\end{array}\right]\left[\begin{array}{c}
\Delta x^{\mathcal{B}} \\
\Delta y
\end{array}\right]=}  \tag{2.14}\\
{\left[\begin{array}{c}
c^{\mathcal{B}}-\left(A^{\mathcal{B}}\right)^{\top} y_{k}-\tau_{k} \mu_{k}\left(X^{\mathcal{B}}\right)^{-1} e_{|\mathcal{B}|} \\
b-A x_{k}+A^{\mathcal{N}} \Theta^{\mathcal{N}}\left(c^{\mathcal{N}}-\left(A^{\mathcal{N}}\right)^{\top} y_{k}-\tau_{k} \mu_{k}\left(X^{\mathcal{N}}\right)^{-1} e_{|\mathcal{N}|}\right)
\end{array}\right] .}
\end{gather*}
$$

Since we know that $\Theta^{\mathcal{N}} \rightarrow 0_{n_{1}, n_{1}}$, we expect that the magnitude of $\left\|A^{\mathcal{N}} \Theta^{\mathcal{N}}\left(A^{\mathcal{N}}\right)^{\top}\right\|$ will be small when the method approaches optimality. Intuitively, our goal is to create a regularization matrix that will implicitly absorb the off-diagonal elements of $A^{\mathcal{N}} \Theta^{\mathcal{N}}\left(A^{\mathcal{N}}\right)^{\top}$ (promoting sparsity) and regularize the system with values having a slightly larger magnitude to that of the elements which were absorbed. For this class of problems, we will focus on solving the normal equations. Given (2.14), we can form the normal equations by eliminating $\Delta x^{\mathcal{B}}$, which gives the following system:

$$
\left[A^{\mathcal{B}} \Theta^{\mathcal{B}}\left(A^{\mathcal{B}}\right)^{\top}+A^{\mathcal{N}} \Theta^{\mathcal{N}}\left(A^{\mathcal{N}}\right)^{\top}+R_{d}\right] \Delta y=b-A x_{k}+A \Theta\left(c-A^{\top} y_{k}-\tau_{k} \mu_{k} X^{-1} e_{n}\right)
$$

We choose the following dual regularization matrix:

$$
\begin{equation*}
R_{d}=\left(\Delta_{d}-\operatorname{Off}\left(A^{\mathcal{N}} \Theta^{\mathcal{N}}\left(A^{\mathcal{N}}\right)^{\top}\right)\right) \tag{2.15}
\end{equation*}
$$

where $\Delta_{d}$ is a diagonal matrix chosen such that $R_{d} \succ 0_{m, m}$ and diagonally dominant, that is:

$$
\left(\Delta_{d}\right)^{(i, i)}>\sum_{j=1, j \neq i}^{m}\left|\left(A^{\mathcal{N}} \Theta^{\mathcal{N}}\left(A^{\mathcal{N}}\right)^{\top}\right)^{(i, j)}\right|, \quad \forall i=1, \ldots, m
$$

For computational efficiency and numerical stability, we choose $\Delta_{d}=\delta_{d, k} I_{m}$, with:

$$
\begin{equation*}
\delta_{d, k}=\left(\max _{j}\left(\Theta^{\mathcal{N}}\right)^{(j, j)}\right)\left\|A^{\mathcal{N}}\left(A^{\mathcal{N}}\right)^{\top}\right\|_{\infty} \tag{2.16}
\end{equation*}
$$

Observe that the regularization matrix given in (2.15), strongly depends on the properties of the problem as well as on the iteration $k$ of the IPM. In order to control which elements enter the set $\mathcal{N}$, at every iteration $k$, we enforce the following condition:

$$
\begin{equation*}
\max _{j}\left(\Theta^{\mathcal{N}}\right)^{(j, j)}\left\|A A^{\top}\right\|_{\infty} \leq \operatorname{reg}_{t h r, k}, \tag{2.17}
\end{equation*}
$$

where $\operatorname{reg}_{t h r, k}$ is set to 1 at the beginning of the optimization $(k=0)$, and is decreased at the same rate as $\mu_{k}$ (i.e. $\operatorname{reg}_{t h r, k}=\mathbf{O}\left(\mu_{k}\right)$ ). Once reg ${ }_{t h r, K}$ becomes smaller than a predefined value, say $\epsilon>0$, for some large $K \geq 1$, we fix it to this value $\left(\mathrm{reg}_{t h r, k}=\epsilon, \forall k \geq K\right)$. The choice of $\epsilon$ will be specified later. Note that (2.17) ensures that $\delta_{d, k}<\mathrm{reg}_{t h r, k}$, at every iteration. In order to show that sparsity is improved, we form again the normal equations' matrix using the
definition of $R_{d}$ to obtain:

$$
A \Theta A^{\top}+R_{d}=A^{\mathcal{B}} \Theta^{\mathcal{B}}\left(A^{\mathcal{B}}\right)^{\top}+\operatorname{Diag}\left(A^{\mathcal{N}} \Theta^{\mathcal{N}}\left(A^{\mathcal{N}}\right)^{\top}\right)+\Delta_{d} .
$$

From the previous one can easily observe that the sparsity of the normal equations is improved, since some off-diagonal elements of the matrix have been absorbed by the regularization.

Since reg ${ }_{\text {thr }, k}$ is not allowed to go to zero as $\mu_{k} \rightarrow 0$, we would like to know how much we perturb the Newton system, by having it fixed to some value $\epsilon>0$, when the method is close to optimality. In the rest of this subsection, we compute some perturbation bounds, which depend on the value of reg ${ }_{t h r}$.

Motivation Now that we have defined the regularization matrix for the case of linear programming problems, let us provide a motivation for this choice. Firstly, note that the proposed regularization has multiple objectives. On the one hand, we want to find a good criterion for tuning a uniform dual regularization matrix $\delta_{d, k} I$ based on the properties of the problem, such that the non-regularized problem matrix is not perturbed significantly while its spectral properties are improved. On the other hand, we use this uniform dual regularization value as a cut-off point, for dropping the smallest off-diagonal elements in the normal equations matrix, improving the computational efficiency of the method. In what follows we will provide an analysis indicating why the uniform dual regularization that we introduce is expected not to perturb the problem significantly. Then, we will show that further dropping the off-diagonal elements introduces a controlled perturbation.

Based on the previous, let us assume for now that $R_{d, k}=\delta_{d, k} I$, where $\delta_{d, k}$ is defined as in (2.16). For simplicity of notation, we omit the iteration subscript in $\delta_{d}$ and we let:

$$
M:=\left[\begin{array}{cc}
-\Theta^{-1} & A^{\top} \\
A & 0_{m, m}
\end{array}\right], \quad E:=\left[\begin{array}{cc}
0_{n, n} & 0_{n, m} \\
0_{m, n} & \delta_{d} I_{m}
\end{array}\right] .
$$

We want to analyze the difference in the eigenvalues of the matrices $M$ and $M+E$. For the rest of this subsection, let $\lambda_{i}$ denote the i-th smallest eigenvalue of $M$, $\tilde{\lambda}_{i}$ the i-th smallest eigenvalue of $M+E$, and $\lambda_{i}(t)$ the i-th smallest eigenvalue of $M+t E$, with $t \in[0,1]$. The smallest eigenvalues of $M$ (in the absolute value sense) will be increased after the addition of $E$ and this is of course desirable, since this was the main motivation for introducing the regularization. The following analysis provides perturbation bounds only for eigenvalues of $M$ that satisfy $\left|\lambda_{i}\right|>2\|E\|$. We will also assume that the eigenvalues that we analyze are simple (i.e. their algebraic multiplicity is 1 ). The analysis can be extended to multiple eigenvalues, however it gets unnecessarily complicated. Such an analysis is derived in the appendix of [127]. Let us now state a lemma derived in [164].

Lemma 2.2.1. Let $M$, $E$ be square Hermitian matrices. Denote by $\lambda_{i}(t)$ the $i$-th smallest eigenvalue of $M+t E$ and consider the eigenvector function $x(t)$ such that: $(M+t E) x(t)=\lambda_{i}(t) x(t)$, with $\|x(t)\|=1$, for some $t \in[0,1]$. If $\lambda_{i}(t)$ is
simple, then:

$$
\frac{\partial \lambda_{i}(t)}{\partial t}=x(t)^{H} E x(t)
$$

As observed in [127], if the eigenvector $x(t)$ has small components in the positions corresponding to the dominant elements of $E$, then $\frac{\partial \lambda_{i}(t)}{\partial t}$ is expected to be small. Let us now provide the following lemma, based on the developments in [44].
Lemma 2.2.2. Let $\lambda_{i} \neq 0$ be an eigenvalue of $M$ and $M x=\lambda_{i} x$, with $\|x\|=1$. Partitioning $x=\left[\begin{array}{ll}x_{1}^{H} & x_{2}^{H}\end{array}\right]^{H}$, we have:

$$
\left\|x_{2}\right\| \leq \frac{\|A\|}{\sqrt{\lambda_{i}^{2}+\|A\|^{2}}}
$$

Proof. The proof follows exactly the developments in [44], but we provide it here for completeness. From the second block equation of $M x=\lambda_{i} x$, we have:

$$
A x_{1}=\lambda_{i} x_{2} \Rightarrow x_{2}=\frac{1}{\lambda_{i}} A x_{1},
$$

where the latter is well defined since we have assumed that $\lambda_{i} \neq 0$. By taking norms on both sides in the previous equation, we get:

$$
\left\|x_{2}\right\| \leq \frac{1}{\left|\lambda_{i}\right|}\|A\|\left\|x_{1}\right\|
$$

But $\|x\|=1 \Rightarrow\left\|x_{1}\right\|=\sqrt{1-\left\|x_{2}\right\|^{2}}$. Hence, we have:

$$
\left\|x_{2}\right\| \leq \frac{\|A\| \sqrt{1-\left\|x_{2}\right\|^{2}}}{\left|\lambda_{i}\right|}
$$

By solving the previous inequality, we obtain:

$$
\left\|x_{2}\right\| \leq \frac{\|A\|}{\sqrt{\lambda_{i}^{2}+\|A\|^{2}}}
$$

which completes the proof.
The following lemma will be a useful tool for the analysis. We omit its trivial proof.

Lemma 2.2.3. Let $f(x)=\frac{x}{\sqrt{a+x^{2}}}$, where $a>0$. Then, $f(x)$ is a monotone increasing function for $x>0$.
Let us now bound the second block of the eigenvector function $x_{2}(t)$ based on the developments in [44].
Lemma 2.2.4. Assume that $\lambda_{i} \neq 0$ is the $i$-th smallest eigenvalue of $M$. Consider the eigenvector function $x(t)$ such that: $(M+t E) x(t)=\lambda_{i}(t) x(t)$, with $\|x(t)\|=1$, $\forall t \in[0,1]$. Partitioning $x(t)=\left[x_{1}(t)^{H} x_{2}(t)^{H}\right]^{H}$ and assuming that $\left|\lambda_{i}\right|>2\|E\|$, we have:

$$
\left\|x_{2}(t)\right\| \leq \frac{\|A\|}{\sqrt{\left(\left|\lambda_{i}\right|-2\|E\|\right)^{2}+\|A\|^{2}}}
$$

Proof. We omit the proof which follows from Lemma 2.2.3 combined with the previous developments. The interested reader can view [44, Lemma 2.8] for a detailed derivation which can be applied directly in our context.

Let us now derive the following theorem which bounds the difference between the $i$-th smallest eigenvalues of the matrices $M$ and $M+E$ respectively.

Theorem 2.2.1. Let $\lambda_{i}$ and $\lambda_{i}$ be the respective $i$-th smallest eigenvalues of $M$ and $M+E$ and define $\phi_{i}=\frac{\|A\|}{\sqrt{\left(\left|\lambda_{i}\right|-2\|E\|^{2}+\|A\|^{2}\right.}}$. For every $i$ such that $\left|\lambda_{i}\right|>2\|E\|$ we have that:

$$
\left|\lambda_{i}-\tilde{\lambda}_{i}\right| \leq\|E\| \phi_{i}^{2} .
$$

Proof. From Lemma 2.2.1 and Lemma 2.2.4 it follows that:

$$
\begin{aligned}
\left|\lambda_{i}-\tilde{\lambda}_{i}\right| & =\left|\lambda_{i}(0)-\lambda_{i}(1)\right| \\
& =\left|\int_{0}^{1} x(t)^{H} E x(t) d t\right| \\
& =\left|\int_{0}^{1} x_{2}(t)^{H} \delta_{d} I_{m} x_{2}(t) d t\right| \\
& =\delta_{d} \int_{0}^{1}\left\|x_{2}(t)\right\|^{2} d t \\
& \leq\|E\| \phi_{i}^{2}=\delta_{d} \phi_{i}^{2} .
\end{aligned}
$$

The proof is complete.

Note that, since $\phi_{i}<1$, the latter is a tighter bound than the general bound provided by Weyl's inequality, given that the eigenvalue under consideration is larger than $2\|E\|$. From the previous results we can draw several useful observations. As we already stated, the smaller the components of $x_{2}(t)$ are, the smaller $\frac{\partial \lambda_{i}(t)}{\partial t}$ is expected to be. But $x_{2}(t)$ is bounded by $\phi_{i}$. Hence, the smaller $\phi_{i}$ is, the more insensitive the eigenvalue $\lambda_{i}$ is to the perturbation $\|E\|=\delta_{d}$. In fact, in the previous theorem we proved that the error in the eigenvalue is bounded by $\|E\| \phi_{i}^{2}$.

Let us now examine the nature of $\phi_{i}$. Firstly, one can see that it depends on the norm of the constraint matrix $A$, and from Lemma 2.2.3 we can observe that it is monotone increasing with respect to the norm of $A$. What this tells us is that the smaller the norm of the constraint matrix $A$ is, the more insensitive the eigenvalues of matrix $M$ are to the perturbation $E$. Of course the latter holds only for eigenvalues that are sufficiently larger than $2\|E\|$. On the other hand, from the definition of $\phi_{i}$, we can see that it is beneficial to have a small $\|E\|$, since then, most of the eigenvalues of $M$ are expected to satisfy: $\left|\lambda_{i}\right|>2\|E\|$.

We now shift our attention to the proposed tuning of the regularization parameters. From (2.17), the set of indices $\mathcal{N}$ is such that:

$$
\max _{j}\left(\Theta^{\mathcal{N}}\right)^{(j, j)}\left\|A A^{\top}\right\|_{\infty} \leq \operatorname{reg}_{t h r} .
$$

Also, from (2.16), we have that $\delta_{d}=\max _{j}\left(\Theta^{\mathcal{N}}\right)^{(j, j)}\left\|A^{\mathcal{N}}\left(A^{\mathcal{N}}\right)^{\top}\right\|_{\infty}$. By combining the previous, we obtain:

$$
\|E\|=\delta_{d} \leq \frac{\operatorname{reg}_{t h r}\left\|A^{\mathcal{N}}\left(A^{\mathcal{N}}\right)^{\top}\right\|_{\infty}}{\left\|A A^{\top}\right\|_{\infty}}
$$

Observe that if $\left\|A A^{\top}\right\|_{\infty}$ is large, we allow few columns to enter the partition $\mathcal{N}$. In this case, $\phi_{i}$ is expected to be close to 1 for most of the eigenvalues $\lambda(M)$. On the other hand, $|\mathcal{N}|$ is increased if the infinity norm of $A A^{\top}$ is small, and in such a case, $\phi_{i}$ is expected to be small for many eigenvalues of the system matrix $M$. A more sophisticated choice for the regularization value based on the derived bounds is possible, however, the proposed regularization has two goals, that is not to perturb the system significantly while introducing sparsity to the problem, and hence the definition of $\delta_{d}$ is computationally advantageous for that. Note that taking advantage of the previously presented bounds indicates that the sufficiently large (in the absolute value sense) eigenvalues of the system matrix ( $\gg 2 \delta_{d}$ ) will be perturbed almost insignificantly. If some eigenvalues of the matrix are very small, the previous arguments break down. We will derive lower bounds for these eigenvalues in the next section.

Having introduced the diagonal uniform regularization $\delta_{d} I_{m}$, let us examine the effect of further dropping the off-diagonal elements $\operatorname{Off}\left(A^{\mathcal{N}} \Theta^{\mathcal{N}}\left(A^{\mathcal{N}}\right)^{\top}\right)$ from the normal equations (2.12). For that, we define $K=A \Theta A^{\top}+\delta_{d} I_{m}$ and $R=$ $\operatorname{Off}\left(A^{\mathcal{N}} \Theta^{\mathcal{N}}\left(A^{\mathcal{N}}\right)^{\top}\right)$ and consider the following generalized eigenvalue problem:

$$
\begin{equation*}
u^{\top} R u=\lambda u^{\top} K u \tag{2.18}
\end{equation*}
$$

The previous is well defined since $K \succ 0_{m, m}$. We will analyze the eigenvalues of $K^{-\frac{1}{2}} R K^{-\frac{1}{2}}$, which is similar to $K^{-1} R$. Now assume by contradiction that $\lambda_{\max }\left(K^{-\frac{1}{2}} R K^{-\frac{1}{2}}\right) \geq 1$. Then from (2.18) and for some eigenvector $u$ corresponding to the maximum eigenvalue, we would have:

$$
u^{\top} R u \geq u^{\top} K u .
$$

By adding $u^{\top} \operatorname{Diag}\left(A^{\mathcal{N}} \Theta^{\mathcal{N}}\left(A^{\mathcal{N}}\right)^{\top}\right) u$ to both sides of the previous inequality, we get:

$$
0 \geq u^{\top}\left(A^{\mathcal{B}} \Theta^{\mathcal{B}}\left(A^{\mathcal{B}}\right)^{\top}\right) u+u^{\top} \operatorname{Diag}\left(A^{\mathcal{N}} \Theta^{\mathcal{N}}\left(A^{\mathcal{N}}\right)^{\top}\right) u+u^{\top} \delta_{d} u
$$

which is a contradiction. Hence $\lambda_{\max }\left(K^{-\frac{1}{2}} R K^{-\frac{1}{2}}\right)<1$. On the other hand, if we assume by contradiction that $\lambda_{\min }\left(K^{-\frac{1}{2}} R K^{-\frac{1}{2}}\right) \leq-1$, from (2.18) and for an eigenvector $u$ corresponding to the minimum eigenvalue, we would obtain:

$$
u^{\top} R u \leq-u^{\top} K u=-u^{\top}\left(A \Theta A^{\top}+\delta_{d} I\right) u \leq-\delta_{d} u^{\top} u
$$

However, using (2.16), we get $-\delta_{d} I_{m}+R \succ 0_{m, m}$, hence $-\delta_{d} u^{\top} u<u^{\top} R u$, which contradicts the previous inequality. Hence, $\lambda_{\min }\left(K^{-\frac{1}{2}} R K^{-\frac{1}{2}}\right)>-1$. Now, one can easily observe that:

$$
K^{-1}(K-R)=I-K^{-1} R, \text { and } \quad \rho\left(K^{-1} R\right)<1
$$

where $\rho(\cdot)$ is the spectral radius, and hence the eigenvalues of $K^{-1}(K-R)$ are clustered around 1 . This supports the claim that further dropping the off-diagonal elements of the part of the normal equations corresponding to indices in $\mathcal{N}$, after adding a uniform dual regularization, introduces a controlled perturbation.

## Quadratic programming

Unlike the case of linear programming, for the case of quadratic programming we employ a primal-dual regularization, that is, we use both $R_{p} \succ 0_{n, n}$ and $R_{d} \succ 0_{m, m}$, as shown in (2.5), to improve the spectral properties of the problem. For this case, we modify the condition for allowing a column to enter the set $\mathcal{N}$, and at each iteration $k$, in place of (2.17), we require:

$$
\begin{equation*}
\max _{j}\left(\Theta^{\mathcal{N}}\right)^{(j, j)} \max \left\{\left\|A A^{\top}\right\|_{\infty},\left\|Q Q^{\top}\right\|_{\infty}\right\} \leq \operatorname{reg}_{t h r, k}, \tag{2.19}
\end{equation*}
$$

where reg $_{t h r, k}$ is updated as indicated in the linear programming case (Subsection 2.2.3). As before, by permuting the columns so that the first $n_{1}$ correspond to indices in $\mathcal{N}$ while the remaining ones correspond to indices in $\mathcal{B}$, the augmented system in (2.10) takes the form:

$$
M_{A S}\left[\begin{array}{c}
\Delta x^{\mathcal{N}}  \tag{2.20}\\
\Delta x^{\mathcal{B}} \\
\Delta y
\end{array}\right]=\left[\begin{array}{c}
\xi_{d \mathcal{N}} \\
\xi_{d \mathcal{B}} \\
\xi_{p}
\end{array}\right],
$$

where

$$
\begin{aligned}
M_{A S} & =\left[\begin{array}{ccc}
-\left(Q^{(\mathcal{N}, \mathcal{N})}+\left(\Theta^{\mathcal{N}}\right)^{-1}+R_{p \mathcal{N}}\right) & -\left(Q^{(\mathcal{B}, \mathcal{N})}\right)^{\top} & \left(A^{\mathcal{N}}\right)^{\top} \\
-Q^{(\mathcal{B}, \mathcal{N})} & -\left(Q^{(\mathcal{B}, \mathcal{B})}+\left(\Theta^{\mathcal{B}}\right)^{-1}+R_{p \mathcal{B}}\right) & A_{\mathcal{B}}^{\top} \\
A_{\mathcal{N}} & R_{d}
\end{array}\right], \\
\xi_{d \mathcal{N}} & =c^{\mathcal{N}}+\left[\begin{array}{ll}
Q^{(\mathcal{N}, \mathcal{N})} & \left(Q^{(\mathcal{B}, \mathcal{N})}\right)^{\top}
\end{array}\right]\left[\begin{array}{l}
x_{k}^{\mathcal{N}} \\
x_{k}^{\mathcal{B}}
\end{array}\right]-\left(A^{\mathcal{N}}\right)^{\top} y_{k}-\tau_{k} \mu_{k}\left(X^{\mathcal{N}}\right)^{-1} e_{|\mathcal{N}|}, \\
\xi_{d \mathcal{B}} & =c^{\mathcal{B}}+\left[\begin{array}{ll}
Q^{(\mathcal{B}, \mathcal{N})} & Q^{(\mathcal{B}, \mathcal{B})}
\end{array}\right]\left[\begin{array}{l}
x_{k}^{\mathcal{N}} \\
x_{k}^{\mathcal{B}}
\end{array}\right]-\left(A^{\mathcal{B}}\right)^{\top} y_{k}-\tau_{k} \mu_{k}\left(X^{\mathcal{B}}\right)^{-1} e_{|\mathcal{B}|}, \\
\xi_{p} & =b-A x_{k},
\end{aligned}
$$

and the permuted matrix $Q$ is:

$$
\mathscr{P} Q \mathscr{P}^{\top}=\left[\begin{array}{cc}
Q^{(\mathcal{N}, \mathcal{N})} & \left(Q^{(\mathcal{B}, \mathcal{N})}\right)^{\top} \\
Q^{(\mathcal{B}, \mathcal{N})} & Q^{(\mathcal{B}, \mathcal{B})}
\end{array}\right],
$$

with $Q^{(\mathcal{N}, \mathcal{N})} \in \mathbb{R}^{n_{1} \times n_{1}}, Q^{(\mathcal{B}, \mathcal{N})} \in \mathbb{R}^{n_{2} \times n_{1}}$ and $Q^{(\mathcal{B}, \mathcal{B})} \in \mathbb{R}^{n_{2} \times n_{2}}$ being the respective blocks of the matrix $Q$, while $R_{p \mathcal{N}} \in \mathbb{R}^{n_{1} \times n_{1}}$ and $R_{p \mathcal{B}} \in \mathbb{R}^{n_{2} \times n_{2}}$ are the only two non-zero blocks of the block-diagonal primal regularization matrix $R_{p}$. As we mentioned earlier, when we solve general convex quadratic programming problems, it is dangerous to eliminate the $(1,1)$ block of $(2.10)$ and solve the problem using (2.12), since the latter system may become dense. However in the linear programming case, our regularization matrix was tuned based on the properties
of the normal equations. In order to overcome this problem, we introduce a primal regularization that can absorb the non-diagonal elements of the $(1,1)$ block of the permuted augmented system (2.20). This allows us to safely (from the sparsity and computational point of view) pivot on this block and perform the analysis in a similar manner as in the linear programming case. Hence, we define:

$$
\begin{equation*}
R_{p \mathcal{N}}=\left(\Delta_{p \mathcal{N}}-\operatorname{Off}\left(Q^{(\mathcal{N}, \mathcal{N})}\right)\right) \tag{2.21}
\end{equation*}
$$

with

$$
\begin{equation*}
\Delta_{p \mathcal{N}}=\left\|Q^{(\mathcal{N}, \mathcal{N})}\right\|_{\infty} I_{n_{1}} \tag{2.22}
\end{equation*}
$$

where $\Delta_{p \mathcal{N}} \in \mathbb{R}^{n_{1} \times n_{1}}$ is a uniform diagonal matrix, which ensures that $R_{p \mathcal{N}} \succ$ $0_{n_{1}, n_{1}}$ and diagonally dominant. Although $\Delta_{p \mathcal{N}}$ can have sizeable values, (2.19) ensures that the respective elements in $\left(\Theta^{\mathcal{N}}\right)^{-1}$ have significantly larger values, making this perturbation acceptable. Using (2.21), the ( 1,1 ) block of (2.20) becomes:

$$
-\left(Q^{(\mathcal{N}, \mathcal{N})}+\left(\Theta^{\mathcal{N}}\right)^{-1}+R_{p \mathcal{N}}\right)=-\left(\left(\Theta^{\mathcal{N}}\right)^{-1}+D_{p \mathcal{N}}\right)
$$

where $D_{p \mathcal{N}}=\operatorname{Diag}\left(Q^{(\mathcal{N}, \mathcal{N})}\right)+\Delta_{p \mathcal{N}}$ is a diagonal matrix. For simplicity of notation, let

$$
\bar{Q}_{\mathcal{N}}=\left(\Theta^{\mathcal{N}}\right)^{-1}+D_{p \mathcal{N}}
$$

Pivoting on the $(1,1)$ block of $(2.20)$ results in the following partially reduced augmented system:

$$
\left[\begin{array}{cc}
M_{1} & \left(A^{\mathcal{B}}\right)^{\top}-Q^{(\mathcal{B}, \mathcal{N})} \bar{Q}_{\mathcal{N}}^{-1}\left(A^{\mathcal{N}}\right)^{\top}  \tag{2.23}\\
A^{\mathcal{B}}-A^{\mathcal{N}} \bar{Q}_{\mathcal{N}}^{-1}\left(Q^{(\mathcal{B}, \mathcal{N})}\right)^{\top} & R_{d}+A^{\mathcal{N}} \bar{Q}_{\mathcal{N}}^{-1}\left(A^{\mathcal{N}}\right)^{\top}
\end{array}\right]\left[\begin{array}{c}
\Delta x^{\mathcal{B}} \\
\Delta y
\end{array}\right]=\left[\begin{array}{l}
\xi_{1} \\
\xi_{2}
\end{array}\right],
$$

where:

$$
\begin{aligned}
M_{1} & =Q^{(\mathcal{B}, \mathcal{N})} \bar{Q}_{\mathcal{N}}^{-1}\left(Q^{(\mathcal{B}, \mathcal{N})}\right)^{\top}-\left(Q^{(\mathcal{B}, \mathcal{B})}+\left(\Theta^{\mathcal{B}}\right)^{-1}+R_{p \mathcal{B}}\right) \\
\xi_{1} & =\xi_{d \mathcal{B}}-Q_{\mathcal{B N}} \bar{Q}_{\mathcal{N}}^{-1} \xi_{d \mathcal{N}} \\
\xi_{2} & =\xi_{p}+A_{\mathcal{N}} \bar{Q}_{\mathcal{N}}^{-1} \xi_{d \mathcal{N}}
\end{aligned}
$$

Using a similar reasoning as before, we will tune the matrix $R_{p \mathcal{B}}$ so that sparsity is promoted. By looking at the $(1,1)$ block of $(2.23)$ (i.e. matrix $\left.M_{1}\right)$, one can see that an obvious choice for this matrix would be:

$$
\begin{equation*}
R_{p \mathcal{B}}=\Delta_{p \mathcal{B}}+\operatorname{Off}\left(Q^{(\mathcal{B}, \mathcal{N})} \bar{Q}_{\mathcal{N}}^{-1}\left(Q^{(\mathcal{B}, \mathcal{N})}\right)^{\top}\right) \tag{2.24}
\end{equation*}
$$

with

$$
\begin{equation*}
\Delta_{p \mathcal{B}}=\max _{j}\left(\bar{Q}_{\mathcal{N}}^{-1}\right)^{(j, j)}\left\|Q^{(\mathcal{B}, \mathcal{N})}\left(Q^{(\mathcal{B}, \mathcal{N})}\right)^{\top}\right\|_{\infty} I_{n_{2}}, \tag{2.25}
\end{equation*}
$$

where $\Delta_{p \mathcal{B}} \in \mathbb{R}^{n_{2} \times n_{2}}$ is a uniform diagonal matrix, which ensures that $R_{p \mathcal{B}} \succ 0_{n_{2}, n_{2}}$ and diagonally dominant. Finally, by looking at the $(2,2)$ block of $(2.23)$, we can define $R_{d}$ in a similar manner as in the linear programming case as:

$$
\begin{equation*}
R_{d}=\Delta_{d}-\operatorname{Off}\left(A^{\mathcal{N}} \bar{Q}_{\mathcal{N}}^{-1}\left(A^{\mathcal{N}}\right)^{\top}\right) \tag{2.26}
\end{equation*}
$$

with

$$
\begin{equation*}
\Delta_{d}=\max _{j}\left(\bar{Q}_{\mathcal{N}}^{-1}\right)^{(j, j)}\left\|A^{\mathcal{N}}\left(A^{\mathcal{N}}\right)^{\top}\right\|_{\infty} I_{m}, \tag{2.27}
\end{equation*}
$$

where again $\Delta_{d} \in \mathbb{R}^{m \times m}$ is a uniform diagonal matrix, which ensures that $R_{d} \succ$ $0_{m, m}$ and diagonally dominant. Note that condition (2.19), which defines columns qualified to enter $\mathcal{N}$, ensures that the positive elements of the diagonal matrices $\Delta_{p \mathcal{B}}, \Delta_{d}$ will be strictly less than reg ${ }_{\text {thr }, k}$, at every iteration $k$ of the algorithm.

Motivation As in the linear programming case, let us provide the motivation for the previously presented regularization scheme. We will derive some useful bounds that extend those provided in the motivation paragraph for the linear programming regularization. All the bounds stated here are direct applications of the results obtained in [44] and for simplicity are given without proofs. Let:

$$
M=\left[\begin{array}{cc}
-Q-\Theta^{-1} & A^{\top} \\
A & 0_{m, n}
\end{array}\right], \quad E=\left[\begin{array}{cc}
\Delta_{p} & 0_{n, m} \\
0_{m, n} & \delta_{d} I_{m}
\end{array}\right],
$$

and denote by $\lambda_{i}$ and $\tilde{\lambda}_{i}$ the $i$-th smallest eigenvalues of $M$ and $M+E$ respectively. Note that $\Delta_{p}$ is a permuted $n \times n$ diagonal matrix, comprised of the two uniform primal regularization matrices $\delta_{p \mathcal{N}} I_{n_{1}}, \delta_{p \mathcal{B}} I_{n_{2}}$, with $n_{1}+n_{2}=n$. Let $\zeta_{i}=\min _{\mu \in \lambda\left(-Q-\Theta^{-1}\right)}\left|\lambda_{i}-\mu\right|$, where $\lambda_{i} \in \lambda(M), \lambda_{i} \notin \lambda\left(-Q-\Theta^{-1}\right)$ and $\lambda_{i} \neq 0$. Let also $M x=\lambda_{i} x$, with $\|x\|=1$. Partitioning $x=\left[\begin{array}{ll}x_{1}^{H} & x_{2}^{H}\end{array}\right]^{H}$, it can be proven as before that:

$$
\left\|x_{1}\right\| \leq \frac{\|A\|}{\sqrt{\zeta_{i}^{2}+\|A\|^{2}}}, \quad\left\|x_{2}\right\| \leq \frac{\|A\|}{\sqrt{\lambda_{i}^{2}+\|A\|^{2}}}
$$

A counterpart of Lemma 2.2.4 for this case follows from [44] and states that if $\left|\lambda_{i}\right|>\delta_{d}+\|E\|$, then, $\forall t \in[0,1]:$

$$
\left\|x_{2}(t)\right\| \leq \frac{\|A\|}{\sqrt{\left(\left|\lambda_{i}\right|-\delta_{d}-\|E\|\right)^{2}+\|A\|^{2}}}=\phi_{i} .
$$

Similarly, if $\zeta_{i}>\left\|\Delta_{p}\right\|+\|E\|$, then $\forall t \in[0,1]$ we have:

$$
\left\|x_{1}(t)\right\| \leq \frac{\|A\|}{\sqrt{\left(\zeta_{i}-\left\|\Delta_{p}\right\|-\|E\|\right)^{2}+\|A\|^{2}}}=\varphi_{i}
$$

where $x(t)=\left[x_{1}(t)^{H} x_{2}(t)^{H}\right]^{H}$ solves the problem $(M+t E) x(t)=\lambda_{i}(t) x(t)$, for some $t \in[0,1]$. For a detailed derivation of the previous results, the interested reader can look at [44, Lemmas 2.8, 2.9].

Finally, the counterpart of Theorem 2.2.1 for this case states that for each $i$ such that: $\left|\lambda_{i}\right|>\delta_{d}+\|E\|$ and $\zeta_{i}>\left\|\Delta_{p}\right\|+\|E\|$, we have:

$$
\left|\lambda_{i}-\tilde{\lambda}_{i}\right| \leq\left\|\Delta_{p}\right\| \varphi_{i}^{2}+\delta_{d} \phi_{i}^{2} .
$$

These bounds are slightly less intuitive than the ones provided for the linear programming case, however similar arguments to those used in the linear programming case can be employed here, supporting the claim that the uni-
form regularization that we introduce does not perturb the sufficiently large (in the absolute value sense) eigenvalues of the non-regularized system significantly. The main reason why we provide these bounds is for completeness. We could proceed by showing, as in the linear programming case, that further dropping $\left.\operatorname{Off}\left(Q^{(\mathcal{N}, \mathcal{N})}\right), \operatorname{Off}\left(Q^{(\mathcal{B}, \mathcal{N})} \bar{Q}_{\mathcal{N}}^{-1} Q^{(\mathcal{B}, \mathcal{N})}\right)^{\top}\right)$ and $\operatorname{Off}\left(A^{\mathcal{N}} \bar{Q}_{\mathcal{N}}^{-1}\left(A^{\mathcal{N}}\right)^{\top}\right)$ (as the proposed non-diagonal regularization suggests) alters the eigenvalues of the diagonally regularized system in a controlled way, but for ease of presentation we omit this.

Rank deficient matrices and the value of $\epsilon$ Notice that both in the linear and the quadratic programming case, during some iterations of the IPM, no columns will satisfy the respective conditions for entering $\mathcal{N}$. In order to ensure that rank deficiency will not get in the way of the proposed method, at every such iteration $k$, we apply a uniform dual regularization $R_{d}=\operatorname{reg}_{t h r, k} I_{m}$, where reg $_{t h r, k}$ is updated as stated in Subsection 2.2.3. In the quadratic programming case, we also include a uniform primal regularization $R_{p}=\operatorname{reg}_{t h r, k} I_{n}$. We expect that sufficiently large (in the absolute value sense) eigenvalues ( $>2 \cdot \mathrm{reg}_{t h r, k}$ ) of the system are perturbed insignificantly by using such a uniform regularization. Once at least one column enters $\mathcal{N}$, we drop this uniform regularization, and start using the regularization matrices presented in this chapter.

Notice that reg ${ }_{t h r}$ is not allowed to decrease more than a pre-specified value $\epsilon>0$. We set this to: $\epsilon=\max \left\{\frac{0.1 \cdot \text { tol }}{\|A\|^{2}}, 10^{-13}\right\}$, where tol is the error tolerance for successful termination of the algorithm and is usually set to the values $10^{-6}$ or $10^{-8}$. This value is based on the bounds derived in the motivation paragraphs presented both for the linear and the quadratic programming case, so that $\epsilon \phi_{i}^{2}$ is small.

### 2.3 Spectral analysis

This section focuses on analysing the spectral properties of the regularized systems provided in the previous section. As before, the analysis is split into linear and quadratic programming respectively. For each of these cases, we will provide the spectral properties of the respective augmented and partially reduced augmented system, showing the effectiveness of the proposed regularization method.

### 2.3.1 Linear programming

For linear programming problems, we employ only dual regularization, that is we set $R_{p}=0_{n, n}$ and use only $R_{d} \succ 0_{m, m}$. In Subsection 2.2.3, it was noted that $\Delta_{d}$ is chosen such that $R_{d} \succ 0_{m, m}$ and diagonally dominant. This is very easy to see, by looking at the definition of (2.16) combined with (2.15). Since $R_{d}$ is diagonally dominant, we are able to invoke the Gershgorin circle theorem, which states that if:

$$
r_{i}=\sum_{j=1, j \neq i}^{m}\left|\left(A^{\mathcal{N}} \Theta^{\mathcal{N}}\left(A^{\mathcal{N}}\right)^{\top}\right)^{(i, j)}\right|,
$$

then any eigenvalue of $R_{d}$ is positive and lies in one of the following discs:

$$
\left\{\lambda:\left|\lambda-\delta_{d}\right| \leq r_{i}\right\},
$$

where $\delta_{d}$ is defined in (2.16), $i=1, \ldots, m$. This yields: $0<\lambda_{i} \leq \delta_{d}+r_{i}, \forall i=$ $1, \ldots, m$, where $\lambda_{i}$ represents the $i$-th eigenvalue of $R_{d}$. On the other hand, by construction, we know that

$$
\delta_{d} \geq r_{i}+\min _{j:\left(A^{\mathcal{N}} \Theta^{\mathcal{N}}\left(A^{\mathcal{N}}\right)^{\top}\right)^{(j, j)>0}}\left(\left(A^{\mathcal{N}} \Theta^{\mathcal{N}}\left(A^{\mathcal{N}}\right)^{\top}\right)^{(j, j)}\right), \quad \forall i=1, \ldots, m,
$$

and hence:
$\min _{j:\left(A^{\mathcal{N}} \Theta^{\mathcal{N}}\left(A^{\mathcal{N}}\right)^{\top}\right)^{(j, j)}>0}\left(\left(A^{\mathcal{N}} \Theta^{\mathcal{N}}\left(A^{\mathcal{N}}\right)^{\top}\right)^{(j, j)}\right) \leq \lambda_{i} \leq \delta_{d}+r_{i}<2 \delta_{d}, \quad \forall i=1, \ldots, m$.
Let us now analyze the spectral properties of the matrix in (2.13). For that we provide the following theorem, which gives bounds for the eigenvalues of the system. The proof is based on the developments in [151] and [161].

Theorem 2.3.1. For all $(x, z)>\left(0_{n}, 0_{n}\right)$ and $R_{d}$ as defined in (2.15), the coefficient matrix of (2.13) has exactly $n$ negative and $m$ positive eigenvalues. Order and denote them as:

$$
\nu_{-n} \leq \nu_{-n+1} \leq \ldots \nu_{-1}<0<\nu_{1} \leq \ldots \nu_{m} .
$$

These eigenvalues satisfy the following bounds:

$$
\begin{aligned}
\nu_{-1} & <-\min _{j}\left(\Theta^{-1}\right)^{(j, j)}, \\
\nu_{-n} & \geq \frac{1}{2}\left(\left(\lambda_{\min }\left(R_{d}\right)-\max _{j}\left(\Theta^{-1}\right)^{(j, j)}\right)-\left[\left(\max _{j}\left(\Theta^{-1}\right)^{(j, j)}+\lambda_{\min }\left(R_{d}\right)\right)^{2}+4\left(\sigma_{\max }(A)\right)^{2}\right]^{\frac{1}{2}}\right), \\
\nu_{m} & \leq \frac{1}{2}\left(2 \delta_{d}+\left(4 \delta_{d}^{2}+4\left(\sigma_{\max }(A)\right)^{2}\right)^{\frac{1}{2}}\right), \\
\nu_{1} & \geq \frac{1}{2}\left(\left(\lambda_{\min }\left(R_{d}\right)-\max _{j}\left(\Theta^{-1}\right)^{(j, j)}\right)+\left[\left(\max _{j}\left(\Theta^{-1}\right)^{(j, j)}+\lambda_{\min }\left(R_{d}\right)\right)^{2}+4\left(\sigma_{\min }(A)\right)^{2}\right]^{\frac{1}{2}}\right) .
\end{aligned}
$$

In case $\operatorname{rank}(A)<m$, the eigenspace of the eigenvalues originating only from $R_{d}$ is $\left\{0_{n}\right\} \times \operatorname{Null}\left(A^{\top}\right)$ and there are $m-\operatorname{rank}(A)$ such eigenvalues.

Proof. Firstly, from Sylvester's law of inertia we know that since $\Theta$ and $A \Theta A^{\top}+$ $R_{d}$ are positive definite, the regularized augmented system matrix of (2.13) possesses precisely $n$ negative and $m$ positive eigenvalues. If $\nu$ is an eigenvalue of the linear system matrix of (2.13), then there are vectors $u \in \mathbb{R}^{n}$ and $p \in \mathbb{R}^{m}$ that cannot both be zero, using which the eigenvalue problem can be written in the following form:

$$
\begin{array}{r}
-\Theta^{-1} u+A^{\top} p=\nu u  \tag{2.29}\\
A u+R_{d} p=\nu p .
\end{array}
$$

As observed in [72], if $\operatorname{rank}(A)<m$, there are some eigenvalues of the matrix in
(2.13), that satisfy: $R_{d} p=\nu p$. The associated eigenspace is $\left\{0_{n}\right\} \times \operatorname{Null}\left(A^{\top}\right)$.

If $\nu<0$ then $u \neq 0_{n}$ since otherwise $p=0_{m}$ because $R_{d} \succ 0_{m, m}$. On the other hand, if $\nu>0$ then $p \neq 0_{m}$ since otherwise $u=0_{n}$ because $\Theta^{-1} \succ 0_{n, n}$. Taking the inner product of the first equation of (2.29) with $u$, and the second equation with $p$ and subtracting the former from the latter gives:

$$
u^{\top} \Theta^{-1} u+p^{\top} R_{d} p=-\nu u^{\top} u+\nu p^{\top} p .
$$

Using the fact that $\Theta^{-1} \succ 0_{n, n}$, along with $R_{d} \succ 0_{m, m}$, and assuming that $\nu<0$ (i.e. $u \neq 0_{n}$ ):

$$
\left(\min _{j}\left(\Theta^{-1}\right)^{(j, j)}+\nu\right) u^{\top} u \leq \nu p^{\top} p
$$

where the inequality follows because the left hand side is as small as possible and we dropped the positive term $p^{\top} R_{d} p$. But since $\nu<0$ in this case, we know that $-\min _{j}\left(\Theta^{-1}\right)^{(j, j)}>\nu=\nu_{-1}$. Furthermore, if $\nu<0$ then we know that $R_{d}-\nu I_{m} \succ 0_{m, m}$. Hence it is invertible and we can solve the second equation of (2.29) with respect to $p$, substitute the result in the first equation and take the inner product with $u$ to get:

$$
\begin{gathered}
p=-\left(R_{d}-\nu I_{m}\right)^{-1} A u \\
-u^{\top} \Theta^{-1} u-u^{\top} A^{\top}\left(R_{d}-\nu I_{m}\right)^{-1} A u=\nu u^{\top} u
\end{gathered}
$$

Hence:

$$
-\max _{j}\left(\Theta^{-1}\right)^{(j, j)}-\left(\sigma_{\max }(A)\right)^{2}\left(\lambda_{\min }\left(R_{d}\right)-\nu\right)^{-1} \leq \nu
$$

where we observed that the left hand side has negative terms, took the most negative possible values for these terms and divided by $u^{\top} u$. Note that for the second term of the left hand side, we used the fact that for two positive definite matrices $A, B$, we have that $\lambda_{\min }(A+B) \geq \lambda_{\min }(A)+\lambda_{\min }(B)$. Solving the previous inequality with respect to $\nu$ (and using the roots of the second order equation), we get that:

$$
\begin{aligned}
\nu_{-n} \geq \frac{1}{2} & \left(\left(\lambda_{\min }\left(R_{d}\right)-\max _{j}\left(\Theta^{-1}\right)^{(j, j)}\right)\right. \\
& \left.-\left[\left(\max _{j}\left(\Theta^{-1}\right)^{(j, j)}+\lambda_{\min }\left(R_{d}\right)\right)^{2}+4\left(\sigma_{\max }(A)\right)^{2}\right]^{\frac{1}{2}}\right) .
\end{aligned}
$$

Now, for the case where $\nu>0$ (where we know that $p \neq 0_{m}$ ), we solve the first equation of (2.29) with respect to $u$, substitute the result in the second one and take the inner product with $p$, to get:

$$
\begin{gathered}
u=\frac{1}{\nu}\left(\frac{1}{\nu} \Theta^{-1}+I_{n}\right)^{-1} A^{\top} p \\
\frac{1}{\nu} p^{\top} A\left(\frac{1}{\nu} \Theta^{-1}+I_{n}\right)^{-1} A^{\top} p+p^{\top} R_{d} p=\nu p^{\top} p
\end{gathered}
$$

Observe that $\lambda_{\max }\left(\left(\frac{1}{\nu} \Theta^{-1}+I_{n}\right)^{-1}\right) \leq 1$. Given that all the terms on the left hand
side are positive, we can take upper bounds for every term, multiply everything by $\nu$ (since $\nu>0$ ) and divide both sides by $p^{\top} p$. This gives us the following second order inequality with respect to $\nu$ :

$$
\nu^{2}-\lambda_{\max }\left(R_{d}\right) \nu-\left(\sigma_{\max }(A)\right)^{2} \leq 0
$$

Solving the previous quadratic inequality, gives:

$$
\nu_{m} \leq \frac{1}{2}\left(2 \delta_{d}+\left(4 \delta_{d}^{2}+4\left(\sigma_{\max }(A)\right)^{2}\right)^{\frac{1}{2}}\right)
$$

where we used the right-most upper bound given in (2.28). Working similarly using the same equation but slightly altered, that is:

$$
\begin{gathered}
u=\left(\Theta^{-1}+\nu I_{n}\right)^{-1} A^{\top} p, \\
p^{\top} A\left(\Theta^{-1}+\nu I_{n}\right)^{-1} A^{\top} p+p^{\top} R_{d} p=\nu p^{\top} p,
\end{gathered}
$$

and by taking lower bounds on each term of the left hand side and re-arranging them, we get the following inequality:

$$
\nu^{2}+\left(\max _{j}\left(\Theta^{-1}\right)^{(j, j)}-\lambda_{\min }\left(R_{d}\right)\right) \nu-\left(\sigma_{\min }(A)^{2}+\max _{j}\left(\Theta^{-1}\right)^{(j, j)} \lambda_{\min }\left(R_{d}\right)\right) \geq 0
$$

Solving the previous yields the last bound:

$$
\begin{aligned}
\nu_{1} \geq & \frac{1}{2} \\
& \left(\left(\lambda_{\min }\left(R_{d}\right)-\max _{j}\left(\Theta^{-1}\right)^{(j, j)}\right)\right. \\
& \left.+\left[\left(\max _{j}\left(\Theta^{-1}\right)^{(j, j)}+\lambda_{\min }\left(R_{d}\right)\right)^{2}+4\left(\sigma_{\min }(A)\right)^{2}\right]^{\frac{1}{2}}\right),
\end{aligned}
$$

which completes the proof.
Below we provide an analogous theorem applied to the matrix of (2.14). Again, we use the definition of $R_{d}$ that is given in (2.15). With this in mind, we know that the $(2,2)$ block of $(2.14)$ is comprised of two diagonal matrices, i.e.:

$$
D^{*}=\operatorname{Diag}\left(A^{\mathcal{N}} \Theta^{\mathcal{N}}\left(A^{\mathcal{N}}\right)^{\top}\right)+\Delta_{d}
$$

where $\Delta_{d}$ is defined in (2.16). The proof is similar to that of the previous theorem, and hence it is not provided here.

Theorem 2.3.2. For all $(x, z)>\left(0_{n}, 0_{n}\right)$ and $R_{d}$ as defined in (2.15), the coefficient matrix of (2.14) has exactly $n_{2}$ negative and $m$ positive eigenvalues. Order and denote them as:

$$
\bar{\nu}_{-n_{2}} \leq \bar{\nu}_{-n_{2}+1} \leq \ldots \bar{\nu}_{-1}<0<\bar{\nu}_{1} \leq \ldots \bar{\nu}_{m} .
$$

These eigenvalues satisfy the following bounds:
$\bar{\nu}_{-1}<-\min _{j}\left(\left(\Theta^{\mathcal{B}}\right)^{-1}\right)^{(j, j)}$,

$$
\begin{aligned}
& \bar{\nu}_{-n_{2}} \geq \frac{1}{2}\left(\left(\min _{i}\left(D^{*}\right)^{(i, i)}-\max _{j}\left(\left(\Theta^{\mathcal{B}}\right)^{-1}\right)^{(j, j)}\right)\right. \\
&\left.-\left[\left(\max _{j}\left(\left(\Theta^{\mathcal{B}}\right)^{-1}\right)^{(j, j)}+\left(\min _{i}\left(D^{*}\right)^{(i, i)}\right)\right)^{2}+4\left(\sigma_{\max }\left(A^{\mathcal{B}}\right)\right)^{2}\right]^{\frac{1}{2}}\right), \\
& \bar{\nu}_{m} \leq \frac{1}{2}\left(\max _{i}\left(D^{*}\right)^{(i, i)}+\left(\left(\max _{i}\left(D^{*}\right)^{(i, i)}\right)^{2}+4\left(\sigma_{\max }\left(A^{\mathcal{B}}\right)\right)^{2}\right)^{\frac{1}{2}}\right), \\
& \bar{\nu}_{1} \geq \frac{1}{2}\left(\left(\min _{i}\left(D^{*}\right)^{i i}-\max _{j}\left(\left(\Theta^{\mathcal{B}}\right)^{-1}\right)^{(j, j)}\right)\right. \\
&+\left[\left(\max _{j}\left(\left(\Theta^{\mathcal{B}}\right)^{-1}\right)^{(j, j)}+\left(\min _{i}\left(D^{*}\right)^{(i, i)}\right)^{2}+4\left(\sigma_{\min }\left(A^{\mathcal{B}}\right)\right)^{2}\right]^{\frac{1}{2}}\right) .
\end{aligned}
$$

In case $\operatorname{rank}\left(A^{\mathcal{B}}\right)<m$, the eigenspace of the eigenvalues originating only from $D^{*}$ is $\left\{0_{n}\right\} \times \operatorname{Null}\left(\left(A^{\mathcal{B}}\right)^{\top}\right)$ and there are $m-\operatorname{rank}\left(A^{\mathcal{B}}\right)$ such eigenvalues.

Now we can compare the bounds given in Theorems 2.3.1 and 2.3.2 and observe clear advantages of using the partially reduced augmented system (2.14) over the full augmented system (2.13). Firstly, one can note that $-\min _{j}\left(\left(\Theta^{\mathcal{B}}\right)^{-1}\right)^{(j, j)}=$ $-\min _{j}\left(\Theta^{-1}\right)^{(j, j)}$, hence the bound for the largest negative eigenvalue is identical for both systems. However, there are two main differences:

1. We have that $\max _{j}\left(\left(\Theta^{\mathcal{B}}\right)^{-1}\right)^{(j, j)} \leq \max _{j}\left(\Theta^{-1}\right)^{(j, j)}$ (and usually we observe that $\left.\max _{j}\left(\left(\Theta^{\mathcal{B}}\right)^{-1}\right)^{(j, j)} \ll \max _{j}\left(\Theta^{-1}\right)^{(j, j)}\right)$. As a consequence the bound on the most negative eigenvalue of (2.13) will be larger (in the absolute value sense), than the bound on the respective eigenvalue of (2.14).
2. Our guaranteed lower bound for the minimum eigenvalue of $R_{d}$ is smaller than the respective lower bound for the minimum eigenvalue of $D^{*}$. In fact,

$$
\begin{gathered}
\min _{i}\left(D^{*}\right)^{(i, i)} \geq \delta_{d} \\
\lambda_{\min }\left(R_{d}\right) \geq \min _{j:\left(A^{\mathcal{N}} \Theta^{\mathcal{N}}\left(A^{\mathcal{N}}\right)^{\top}\right)^{(j, j)}>0}\left(\left(A^{\mathcal{N}} \Theta^{\mathcal{N}}\left(A^{\mathcal{N}}\right)^{\top}\right)^{(j, j)}\right),
\end{gathered}
$$

where $\delta_{d}$ is defined in (2.16) and the second lower bound is given in (2.28). By construction the first bound is better. As a consequence, the smallest positive eigenvalue of (2.14) is guaranteed to be at least as large as $\delta_{d}$.

### 2.3.2 Quadratic programming

For quadratic programming problems we employ a primal-dual regularization. In Subsection 2.2.3, it was noted that $\Delta_{d}$ is chosen such that $R_{d} \succ 0_{m, m}$ and diagonally dominant, while $\Delta_{p \mathcal{B}}$ is chosen such that $R_{p \mathcal{B}} \succ 0_{n_{2}, n_{2}}$ and diagonally dominant. This can be seen by looking at (2.27) combined with (2.26) and (2.25) combined with (2.24), respectively. Similarly, positive definiteness and diagonal dominance of $R_{p \mathcal{N}}$ follows immediately by construction, i.e., by looking at
equations (2.21) and (2.22). For notational convenience, we define:

$$
\bar{Q}_{\mathcal{N}}=\left(\Theta^{\mathcal{N}}\right)^{-1}+\operatorname{Diag}\left(Q^{(\mathcal{N}, \mathcal{N})}\right)+\Delta_{p \mathcal{N}} .
$$

- For $R_{d}$, we are able to invoke the Gershgorin circle theorem as in the linear programming case stating that if:

$$
r_{i}=\sum_{j=1, j \neq i}^{m}\left|\left(A^{\mathcal{N}} \bar{Q}_{\mathcal{N}}^{-1}\left(A^{\mathcal{N}}\right)^{\top}\right)^{(i, j)}\right|,
$$

then any eigenvalue of $R_{d}$ is positive and lies in one of the following discs:

$$
\left\{\lambda:\left|\lambda-\delta_{d}\right| \leq r_{i}\right\}
$$

where $\delta_{d}=\max _{j}\left(\bar{Q}_{\mathcal{N}}^{-1}\right)^{(j, j)}\left\|A^{\mathcal{N}}\left(A^{\mathcal{N}}\right)^{\top}\right\|_{\infty}, i=1, \ldots, m$. This yields: $0<$ $\lambda_{i} \leq \delta_{d}+r_{i}, \forall i=1, \ldots, m$, where $\lambda_{i}$ is the i-th eigenvalue of $R_{d}$. On the other hand, by construction we know that

$$
\delta_{d} \geq r_{i}+\min _{j:\left(A^{\mathcal{N}} \bar{Q}_{\mathcal{N}}^{-1}\left(A^{\mathcal{N}}\right)^{\top}\right)^{(j, j)}>0}\left(\left(A^{\mathcal{N}} \bar{Q}_{\mathcal{N}^{-1}}^{-1}\left(A^{\mathcal{N}}\right)^{\top}\right)^{(j, j)}\right),
$$

for all $i=1, \ldots, m$, and hence:

- For $R_{p \mathcal{B}}$, we apply the same theorem, however in this case we have:

$$
r_{i}=\sum_{j=1, j \neq i}^{n_{2}}\left|\left(Q^{(\mathcal{B}, \mathcal{N})} \bar{Q}_{\mathcal{N}}^{-1}\left(Q^{(\mathcal{B}, \mathcal{N}}\right)^{\top}\right)^{(i, j)}\right|,
$$

and any eigenvalue of $R_{p \mathcal{B}}$ is positive and lies in one of the following discs:

$$
\left\{\lambda:\left|\lambda-\delta_{p B}\right| \leq r_{i}\right\},
$$

where $\delta_{p \mathcal{B}}=\max _{j}\left(\bar{Q}_{\mathcal{N}}^{-1}\right)^{(j, j)}\left\|Q^{(\mathcal{B}, \mathcal{N})}\left(Q^{(\mathcal{B}, \mathcal{N})}\right)^{\top}\right\|_{\infty}, i=1, \ldots, n_{2}$. As before, we know that:

$$
\begin{equation*}
\min _{j:\left(Q^{(\mathcal{B}, \mathcal{N})} \bar{Q}_{\mathcal{N}}^{-1}\left(Q^{(\mathcal{N}, \mathcal{N})}\right)^{\top}\right)^{(j, j)}>0}\left(\left(Q^{(\mathcal{B}, \mathcal{N})} \bar{Q}_{\mathcal{N}}^{-1}\left(Q^{(\mathcal{B}, \mathcal{N})}\right)^{\top}\right)^{(j, j)}\right) \leq \lambda_{i} \leq \delta_{p \mathcal{B}}+r_{i}<2 \delta_{p \mathcal{B}}, \tag{2.31}
\end{equation*}
$$

for all $i=1, \ldots, n_{2}=|\mathcal{B}|$, where $\lambda_{i}$ is the i-th eigenvalue of $R_{p \mathcal{B}}$.

- Finally, we can work similarly to examine the spectral properties of $R_{p \mathcal{N}}$. Again by letting:

$$
r_{i}=\sum_{j=1, j \neq i}^{n_{1}}\left(Q^{(\mathcal{N}, \mathcal{N})}\right)^{(i, j)},
$$

any eigenvalue of $R_{p \mathcal{N}}$ is positive and lies in one of the following discs:

$$
\left\{\lambda:\left|\lambda-\delta_{p \mathcal{N}}\right| \leq r_{i}\right\}
$$

where $\delta_{p \mathcal{N}}=\left\|Q^{(\mathcal{N}, \mathcal{N})}\right\|_{\infty}, i=1, \ldots, n_{1}$. This yields: $0<\lambda_{i} \leq \delta_{p \mathcal{N}}+r_{i}$, $\forall i=1, \ldots, n_{1}$, where $\lambda_{i}$ is the $i$-th eigenvalue of $R_{p \mathcal{N}}$. But since $Q^{(\mathcal{N}, \mathcal{N})} \succeq$ $0_{n_{1}, n_{1}}$ as a principal minor of $Q \succeq 0_{n, n}$, we know that if a diagonal element of $Q^{(\mathcal{N}, \mathcal{N})}$ is zero, then its respective column and row are also zero. Hence this implies tighter final bounds, that is:

$$
\begin{equation*}
\min _{\left.j:\left(Q^{(\mathcal{N}, \mathcal{N})}\right)_{(i, j)}\right)>0}\left(\left(Q^{(\mathcal{N}, \mathcal{N})}\right)^{(j, j)}\right)<\lambda_{i} \leq \delta_{p \mathcal{N}}+r_{i}<2 \delta_{p \mathcal{N}}, \quad \forall i=1, \ldots, n_{1}=|\mathcal{N}| . \tag{2.32}
\end{equation*}
$$

Let us now analyze the spectral properties of (2.20). To that end, we provide the following theorem, which is the extension of Theorem 2.3.1 for the QP case. The proof is almost identical and hence it is not provided here. For notational convenience, let:

$$
H:=Q+\Theta^{-1}+R_{p}
$$

Theorem 2.3.3. For all $(x, z)>\left(0_{n}, 0_{n}\right)$ and $R_{d}, R_{p \mathcal{B}}, R_{p \mathcal{N}}$ as defined in (2.26), (2.24) and (2.21) respectively, the coefficient matrix of (2.20) has exactly $n$ negative and $m$ positive eigenvalues. Order and denote them as:

$$
\nu_{-n} \leq \nu_{-n+1} \leq \ldots \nu_{-1}<0<\nu_{1} \leq \ldots \nu_{m}
$$

These eigenvalues satisfy the following bounds:
$\nu_{-1}<-\lambda_{\min }(H)$,
$\nu_{-n} \geq \frac{1}{2}\left(\left(\lambda_{\min }\left(R_{d}\right)-\lambda_{\max }(H)\right)-\left[\left(\lambda_{\max }(H)+\lambda_{\min }\left(R_{d}\right)\right)^{2}+4\left(\sigma_{\max }(A)\right)^{2}\right]^{\frac{1}{2}}\right)$,
$\nu_{m} \leq \frac{1}{2}\left(2 \delta_{d}+\left(4 \delta_{d}^{2}+4\left(\sigma_{\max }(A)\right)^{2}\right)^{\frac{1}{2}}\right)$,
$\nu_{1} \geq \frac{1}{2}\left(\left(\lambda_{\min }\left(R_{d}\right)-\lambda_{\max }(H)\right)+\left[\left(\lambda_{\max }(H)+\lambda_{\min }\left(R_{d}\right)\right)^{2}+4\left(\sigma_{\min }(A)\right)^{2}\right]^{\frac{1}{2}}\right)$.
In case $\operatorname{rank}(A)<m$, the eigenspace of the eigenvalues originating only from $R_{d}$ is $\left\{0_{n}\right\} \times \operatorname{Null}\left(A^{\top}\right)$ and there are $m-\operatorname{rank}(A)$ such eigenvalues.

Below we provide a similar theorem, applied to (2.23). For that, we will use $R_{d}, R_{p \mathcal{B}}, R_{p \mathcal{N}}$ as defined in Subsection 2.2.3 as well as the respective eigenvalue bounds given in (2.30), (2.31), and (2.32). Using the definitions of the regularization matrices, we know that the matrix in the $(1,1)$ block of $(2.23)$ takes the form:

$$
\bar{H}:=Q^{(\mathcal{B}, \mathcal{B})}+\left(\Theta^{\mathcal{B}}\right)^{-1}+\Delta_{p \mathcal{B}}-\operatorname{Diag}\left(Q^{(\mathcal{B}, \mathcal{N})} \bar{Q}_{\mathcal{N}}^{-1}\left(Q^{(\mathcal{B}, \mathcal{N})}\right)^{\top}\right)
$$

while the $(2,2)$ block of ( 2.23 ) becomes:

$$
D^{*}:=\operatorname{Diag}\left(A^{\mathcal{N}} \bar{Q}_{\mathcal{N}}^{-1}\left(A^{\mathcal{N}}\right)^{\top}\right)+\Delta_{d} .
$$

Theorem 2.3.4. For all $(x, z)>\left(0_{n}, 0_{n}\right)$ and $R_{d}, R_{p \mathcal{B}}, R_{p \mathcal{N}}$ as defined in (2.26), (2.24) and (2.21) respectively, the coefficient matrix of (2.23) has exactly $n_{2}$ negative and $m$ positive eigenvalues. Order and denote them as:

$$
\bar{\nu}_{-n_{2}} \leq \bar{\nu}_{-n+1} \leq \ldots \bar{\nu}_{-1}<0<\bar{\nu}_{1} \leq \ldots \bar{\nu}_{m}
$$

These eigenvalues satisfy the following bounds:
$\bar{\nu}_{-1}<-\lambda_{\text {min }}(\bar{H})$,

$$
\begin{aligned}
\bar{\nu}_{-n_{2}} \geq \frac{1}{2} & \left(\left(\min _{j}\left(D^{*}\right)^{(j, j)}-\lambda_{\max }(\bar{H})\right)\right. \\
& \left.-\left[\left(\lambda_{\max }(\bar{H})+\min _{j}\left(D^{*}\right)^{(j, j)}\right)^{2}+4\left(\sigma_{\max }\left(A^{\mathcal{B}}-A^{\mathcal{N}} \bar{Q}_{\mathcal{N}}^{-1}\left(Q^{(\mathcal{B}, \mathcal{N})}\right)^{\top}\right)\right)^{2}\right]^{\frac{1}{2}}\right),
\end{aligned}
$$

$$
\bar{\nu}_{m} \leq \frac{1}{2}\left(\max _{j}\left(D^{*}\right)^{(j, j)}+\left[\max _{j}\left(\left(D^{*}\right)^{(j, j)}\right)^{2}+4\left(\sigma_{\max }\left(A^{\mathcal{B}}-A^{\mathcal{N}} \bar{Q}_{\mathcal{\mathcal { N }}}^{-1}\left(Q^{(\mathcal{B}, \mathcal{N})}\right)^{\top}\right)\right)^{2}\right]^{\frac{1}{2}}\right)
$$

$$
\bar{\nu}_{1} \geq \frac{1}{2}\left(\left(\min _{j}\left(D^{*}\right)^{(j, j)}-\lambda_{\max }(\bar{H})\right)\right.
$$

$$
\left.+\left[\left(\lambda_{\max }(\bar{H})+\min _{j}\left(D^{*}\right)^{(j, j)}\right)^{2}+4\left(\sigma_{\min }\left(A^{\mathcal{B}}-A^{\mathcal{N}} \bar{Q}_{\mathcal{N}}^{-1}\left(Q^{(\mathcal{B}, \mathcal{N})}\right)^{\top}\right)\right)^{2}\right]^{\frac{1}{2}}\right)
$$

In case $\operatorname{rank}\left(A^{\mathcal{B}}-A^{\mathcal{N}} \bar{Q}_{\mathcal{N}}^{-1}\left(Q^{(\mathcal{B}, \mathcal{N})}\right)^{\top}\right)<m$, the eigenspace of the eigenvalues originating only from $D^{*}$ is $\left\{0_{n}\right\} \times \operatorname{Null}\left(\left(A^{\mathcal{B}}\right)^{\top}-Q^{(\mathcal{B}, \mathcal{N})} \bar{Q}_{\mathcal{N}}^{-1}\left(A^{\mathcal{N}}\right)^{\top}\right)$ and there are $m-\operatorname{rank}\left(A^{\mathcal{B}}-A^{\mathcal{N}} \bar{Q}_{\mathcal{N}}^{-1}\left(Q^{(\mathcal{B}, \mathcal{N})}\right)^{\top}\right)$ such eigenvalues.

Let us compare the bounds given in Theorems 2.3.3 and 2.3.4 to observe once again the advantages of using the partially reduced augmented system (2.23) over the full augmented system (2.20). There are three significant differences in the eigenvalue bounds of these two systems:

1. For the bound on the largest negative eigenvalue of the two systems, we know that:

$$
\lambda_{\min }(H) \geq \min _{j}\left(\Theta^{-1}\right)^{(j, j)}+\lambda_{\min }\left(R_{p}\right),
$$

where

$$
\begin{aligned}
& \lambda_{\min }\left(R_{p}\right) \geq \min \left\{\min _{j:\left(Q^{(\mathcal{B}, \mathcal{N})} \bar{Q}_{\mathcal{N}}^{-1}\left(Q^{(\mathcal{B}, \mathcal{N})}\right)^{\top}\right)^{(j, j)}>0}\left(Q^{(\mathcal{B}, \mathcal{N})} \bar{Q}_{\mathcal{N}}^{-1}\left(Q^{(\mathcal{B}, \mathcal{N})}\right)^{\top}\right)^{(j, j)},\right. \\
& \left.\min _{j:\left(Q^{\left.(\mathcal{N}, \mathcal{N})()^{(j, j)}\right)>0}\right.}\left(\left(Q^{(\mathcal{N}, \mathcal{N})}\right)^{(j, j)}\right)\right\},
\end{aligned}
$$

from (2.31) and (2.32) respectively. However, since $\min _{j}\left(\left(\Theta^{\mathcal{B}}\right)^{-1}\right)^{(j, j)} \ll$ $\min _{j}\left(\left(\Theta^{\mathcal{N}}\right)^{-1}\right)^{(j, j)}$ we can conclude that:

$$
\lambda_{\min }(H) \geq \min _{j}\left(\left(\Theta^{\mathcal{B}}\right)^{-1}\right)^{(j, j)}+\lambda_{\min }\left(R_{p}\right)
$$

while

$$
\begin{aligned}
\lambda_{\min }(\bar{H}) \geq & \min _{j}\left(\left(\Theta^{\mathcal{B}}\right)^{-1}\right)^{(j, j)}+\max _{j}\left(\bar{Q}_{\mathcal{N}}^{-1}\right)^{(j, j)}\left\|Q^{(\mathcal{B}, \mathcal{N})}\left(Q^{(\mathcal{B}, \mathcal{N})}\right)^{\top}\right\|_{\infty}- \\
& \max _{j}\left(Q^{(\mathcal{B}, \mathcal{N})} \bar{Q}_{\mathcal{N}}^{-1}\left(Q^{(\mathcal{B}, \mathcal{N})}\right)^{\top}\right)^{(j, j)}
\end{aligned}
$$

where we used (2.25) as the definition of $\Delta_{p \mathcal{B}}$. We observe that the difference:

$$
\max _{j}\left(\bar{Q}_{\mathcal{N}}^{-1}\right)^{(j, j)}\left\|Q^{(\mathcal{B}, \mathcal{N})}\left(Q^{(\mathcal{B}, \mathcal{N})}\right)^{\top}\right\|_{\infty}-\max _{j}\left(Q^{(\mathcal{B}, \mathcal{N})} \bar{Q}_{\mathcal{N}}^{-1}\left(Q^{(\mathcal{B}, \mathcal{N})}\right)^{\top}\right)^{(j, j)}
$$

increases as more elements enter the set $\mathcal{N}$. On the other hand, $\lambda_{\text {min }}\left(R_{p}\right)$ is expected to decrease at every iteration of the interior-point method. Hence the bound on $\bar{\mu}_{-1}$ is expected to be better than that on $\mu_{-1}$, as more elements enter the partition $\mathcal{N}$.
2. For the bound on the most negative eigenvalue of the two systems, we know that:

$$
\lambda_{\max }(H) \leq \lambda_{\max }(Q)+\max _{j}\left(\Theta^{-1}\right)^{(j, j)}+\lambda_{\max }\left(R_{p}\right)
$$

where $\lambda_{\max }\left(R_{p}\right) \leq 2 \max \left\{\delta_{p \mathcal{N}}, \delta_{p \mathcal{B}}\right\}$. However, since $\max _{j}\left(\left(\Theta^{\mathcal{N}}\right)^{-1}\right)^{(j, j)} \geq$ $\max _{j}\left(\left(\Theta^{\mathcal{B}}\right)^{-1}\right)^{(j, j)}$, we observe that:

$$
\lambda_{\max }(H) \leq \lambda_{\max }(Q)+\max _{j}\left(\left(\Theta^{\mathcal{N}}\right)^{-1}\right)^{(j, j)}+\lambda_{\max }\left(R_{p}\right)
$$

where we used the definition of $\Delta_{p \mathcal{N}}$ given in (2.22). On the other hand,

$$
\lambda_{\max }(\bar{H}) \leq \lambda_{\max }\left(Q^{(\mathcal{B}, \mathcal{B}}\right)+\max _{j}\left(\left(\Theta^{\mathcal{B}}\right)^{-1}\right)^{(j, j)}+\left(\Delta_{p \mathcal{B}}\right)^{(i, i)}, \quad \forall i \in\{1, \ldots, n\}
$$

where, from (2.25), we know that

$$
\left(\Delta_{p \mathcal{B}}\right)^{(i, i)}=\max _{j}\left(\bar{Q}_{\mathcal{N}}^{-1}\right)^{(j, j)}\left\|Q^{(\mathcal{B}, \mathcal{N})}\left(Q^{(\mathcal{B}, \mathcal{N})}\right)^{\top}\right\|_{\infty}, \quad \forall i \in\{1, \ldots, n\} .
$$

Clearly the bound on $\lambda_{\max }(\bar{H})$ is significantly smaller than that on $\lambda_{\max }(H)$, since it is usually the case that $\max _{j}\left(\left(\Theta^{\mathcal{N}}\right)^{-1}\right)^{(j, j)} \gg \max _{j}\left(\left(\Theta^{\mathcal{B}}\right)^{-1}\right)^{(j, j)}$, while $\lambda_{\max }\left(R_{p}\right)>\max _{j}\left(\bar{Q}_{\mathcal{N}}^{-1}\right)^{(j, j)}\left\|Q^{(\mathcal{B}, \mathcal{N})}\left(Q^{(\mathcal{B}, \mathcal{N})}\right)^{\top}\right\|_{\infty}$. Hence, the most negative eigenvalue of (2.23) is expected to have a significantly smaller magnitude than that of (2.20).
3. As in the LP case, our guaranteed lower bound for the minimum eigenvalue of $R_{d}$ is smaller than the respective lower bound for the minimum eigenvalue
of $D^{*}$. In fact,

$$
\begin{gathered}
\min _{i} D_{i i}^{*} \geq \delta_{d} \\
\lambda_{\min }\left(R_{d}\right) \geq \min _{j:\left(A^{\mathcal{N}} \bar{Q}_{\mathcal{N}}^{-1}\left(A^{\mathcal{N}}\right)^{\top}\right)^{(j, j)}>0}\left(A^{\mathcal{N}} \bar{Q}_{\mathcal{N}^{-1}}^{-1}\left(A^{\mathcal{N}}\right)^{\top}\right)^{(j, j)},
\end{gathered}
$$

where we use $\delta_{d}$ as defined in (2.27), while the last inequality follows from (2.30). By construction, the first bound is better. As a consequence, the smallest positive eigenvalue of (2.23) is guaranteed to be at least as large as $\delta_{d}$.

### 2.4 Implementation and numerical results

### 2.4.1 The algorithmic framework

At this point, we are providing a generic algorithm (NDR-IPM), summarizing the infeasible primal-dual IPM with non-diagonal regularization. The algorithm solves the Newton system arising from the optimality conditions of (2.1)-(2.2), at each iteration, using a direct method. Note that this is just a general outline and does not contain the actual details of the implemented method. Implementation details will be presented in the next subsection. Within the algorithm, we make the distinction between linear and quadratic programming problems, by using the logical variables LP and QP, respectively.

### 2.4.2 Implementation details

We implemented the algorithm in MATLAB. The implementation solves linear and convex quadratic programming problems in the standard form. However, all the free variables are treated as variables bounded by some box constraints. We set some initial bounds,

$$
l_{f}=-10^{2} \leq x_{f} \leq 10^{2}=u_{f},
$$

for all the free variables. If the method pushes some of these variables to take values outside of this box, then the respective bounds are increased to give space for variables to increase their values. Note that this heuristic causes that extra iterations are needed to converge for a few problems, since every time the box constraints are changed, the method loses primal feasibility.

Regularization We set $\mathrm{reg}_{t h r, 0}=1$, and we decrease it at the same rate as $\mu_{k}$ decreases, until it becomes smaller than $\epsilon=\max \left\{\frac{\mathrm{tol} \cdot 10^{-1}}{\|A\|_{2}^{2}}, 10^{-13}\right\}$. Then, it takes this value and stays constant for the rest of the optimization process. As before, tol is the error tolerance specified by the user. At every iteration, we enable columns to enter the set $\mathcal{N}$ only if:

$$
\max _{j \in \mathcal{N}}(\Theta)^{(j, j)} \max \left\{\left\|A A^{\top}\right\|_{\infty},\left\|Q Q^{\top}\right\|_{\infty}\right\} \leq \operatorname{reg}_{t h r, k}
$$

```
Algorithm NDR-IPM Infeasible IPM with non-diagonal regularization
Input: \(A, Q, b, c\), tol, maxit
Parameters: \(0<\tau_{\min } \leq \tau_{\max }, \epsilon=\max \left\{\frac{\mathrm{tol} \cdot 10^{-1}}{\|A\|_{2}^{2}}, 10^{-13}\right\}, \nu \in(0,1)\).
Initial point: Choose a well-centred \(w_{0}=\left(x_{0}, y_{0}, r_{0}, s_{0}, z_{0}\right)\) with
\[
x_{0}, z_{0}>0_{n}, \mu_{0}=\frac{x_{0}^{\top} z_{0}}{n}, k=0 .
\]
    \(\operatorname{reg}_{t h r, 0}=1, \operatorname{res}_{p}^{0}=b-A x_{0}, \operatorname{res}_{d}^{0}=c-A^{\top} y_{0}-z_{0}+Q x_{0}\).
    while ( \(k<\) maxit) do
    if \(\left(\left(\left\|\operatorname{res}_{p}^{k}\right\|<\right.\right.\) tol \() \wedge\left(\left\|\operatorname{res}_{d}^{k}\right\|<\right.\) tol \() \wedge\left(\mu_{k}<\right.\) tol \(\left.)\right)\) then
        Declare convergence and return the optimal solution.
        return \(\left(x_{k}, y_{k}, z_{k}\right)\).
    else
        reg \(_{t h r, k}=\max \left\{O\left(\mu_{k}\right), \epsilon\right\}\).
        if \((\mathcal{N}=\emptyset)\) then
            \(R_{d}=\mathrm{reg}_{t h r, k} I_{m}\).
            if (QP) then
                \(R_{p}=\operatorname{reg}_{t h r, k} I_{n}\).
            end if
            else
                if (LP) then
                \(R_{d}\) from (2.15) and (2.16), \(R_{p}=0\).
                else if (QP) then
                    \(R_{d}\) from (2.26), (2.27) and \(R_{p}\) from (2.21), (2.22), (2.24), (2.25).
                end if
            end if
            Choose \(\tau_{k} \in\left[\tau_{\min }, \tau_{\text {max }}\right]\).
            if (LP) then
                Compute \(\Delta w_{k}=\left(\Delta x_{k}, \Delta y_{k}, \Delta r_{k}, \Delta z_{k}\right)\) using (2.12).
                \(\left(s_{k}=0_{n}, \Delta s=0_{n}\right)\).
            else if (QP) then
            Compute \(\Delta w_{k}=\left(\Delta x_{k}, \Delta y_{k}, \Delta r_{k}, \Delta s_{k}, \Delta z_{k}\right)\) using (2.23).
            end if
            \(a_{x}^{\max }=\min _{\Delta x^{i}<0}\left\{1,-\frac{x^{i}}{\Delta x^{i}}\right\}, \quad a_{z}^{\max }=\min _{\Delta z^{i}<0}\left\{1,-\frac{z^{i}}{\Delta z^{i}}\right\}\).
            \(x_{k}(a)=x_{k}+\nu a_{x}^{\max } \Delta x, r_{k}(a)=r_{k}+\nu a_{x}^{\max } \Delta r\).
            \(z_{k}(a)=z_{k}+\nu a_{z}^{\max } \Delta z, y_{k}(a)=y_{k}+\nu a_{z}^{\max } \Delta y, s_{k}(a)=s_{k}+\nu a_{z}^{\max } \Delta s\).
            \(\mu_{k}(a)=\frac{x_{k}(a)^{\top} z_{k}(a)}{n}\).
            \(k=k+1\).
    end if
    end while
```

This ensures that $\left(\Delta_{d}\right)^{(i, i)}$, as defined in (2.16) and (2.27) for linear and convex quadratic problems, respectively, is smaller than $\operatorname{reg}_{t h r, k}, \forall i \in\{1, \ldots, m\}, \forall k \geq$ 0 . The latter also holds for $\left(\Delta_{p \mathcal{B}}\right)^{(i, i)}$ as in (2.25), $\forall i \in\left\{1, \ldots, n_{2}\right\}$, which is only defined for quadratic programming problems. Of course for linear programming problems, we have $R_{p}=0_{n, n}$. Note that during the first iterations of the method, $\mathcal{N}$ is usually empty. In order to avoid instability, we include a uniform dual regularization $R_{d}=\operatorname{reg}_{t h r, k} I_{m}$. For the quadratic programming case, we also include a uniform primal regularization, that is: $R_{p}=\operatorname{reg}_{t h r, k} I_{n}$. This uniform regularization is dropped when $\mathcal{N}$ is non-empty. As an extra safeguard, when the factorization of the system fails, we increase reg ${ }_{t h r}$ by a factor of 10 and repeat the factorization. If this process is repeated for 6 consecutive times, we stop the method. All other implementation details concerning the regularization follow from Section 2.2.

Newton-step computation For general convex quadratic problems, the Newton direction is calculated from system (2.23), after computing its symmetric $L D L^{\top}$ decomposition, where $L$ is a lower triangular matrix and $D$ is diagonal. For that, we use the build-in MATLAB symmetric decomposition (i.e. ldl). We know that such a decomposition always exists, with $D$ diagonal, for the aforementioned system, since after introducing the regularization, the matrix of (2.23) is guaranteed to be quasi-definite; a class of matrices known to be strongly factorizable [171]. For that reason, we change the default pivot threshold of ldl to $10^{-13}$. We use such a small pivot threshold in order to avoid any $2 \times 2$ pivots during the factorization routine. For linear programming problems, we solve the system (2.12) (with $Q=0_{n, n}$ ), using the build-in Cholesky decomposition of MATLAB (i.e. chol). $\Delta x$ is then recovered from (2.11). In the quadratic programming case, $\Delta s$ is recovered from (2.7). In both cases $\Delta z$ is recovered from (2.9) and $\Delta r$ from (2.6).

Starting point We have already mentioned that the method is infeasible and hence the starting point does not need to be primal and dual feasible. The only requirement is that the initial values of the variables $x, z$ are strictly positive. We use a starting point that was proposed in [119]. Here we will only state it for completeness. To construct this point, we try to solve the pair of problems (CQP)-(CQD), but we ignore the non-negativity constraints. Such relaxed problems have closed form solutions:

$$
\tilde{x}=A^{\top}\left(A A^{\top}\right)^{-1} b, \quad \tilde{y}=\left(A A^{\top}\right)^{-1} A(c+Q \tilde{x}), \quad \tilde{z}=c-A^{\top} \tilde{y}+Q \tilde{x} .
$$

However, in order to ensure stability and efficiency, we regularize the matrix $A A^{\top}$ and employ the preconditioned conjugate gradient (PCG) method ([93]) to solve these systems (in order to avoid forming $A A^{\top}$ ). We use the classical Jacobi preconditioner to accelerate PCG, i.e. $P=\operatorname{Diag}\left(A A^{\top}\right)+\delta I_{m}$, where $\delta=8$, is set as the regularization parameter. Then, in order to guarantee positivity and sufficient magnitude of $x, z$, we compute the expressions $\delta_{x}=\max \{-1.5 \cdot$
$\left.\min _{i} \tilde{x}^{i}, 0\right\}$ and $\delta_{z}=\max \left\{-1.5 \cdot \min _{i} \tilde{z}^{i}, 0\right\}$ and we obtain:

$$
\tilde{\delta}_{x}=\delta_{x}+0.5 \frac{\left(\tilde{x}+\delta_{x} e\right)^{\top}\left(\tilde{z}+\delta_{z} e\right)}{\sum_{i=1}^{n}\left(\tilde{z}_{i}+\delta_{z}\right)}, \quad \tilde{\delta}_{z}=\delta_{z}+0.5 \frac{\left(\tilde{x}+\delta_{x} e_{n}\right)^{\top}\left(\tilde{z}+\delta_{z} e_{n}\right)}{\sum_{i=1}^{n}\left(\tilde{x}^{i}+\delta_{x}\right)}
$$

Finally, we define the starting point by setting:

$$
r^{0}=0, \quad s^{0}=0, \quad y^{0}=\tilde{y}, \quad z_{i}^{0}=\tilde{z}_{i}+\tilde{\delta}_{z}, \quad x_{i}^{0}=\tilde{x}_{i}+\tilde{\delta}_{x}, \quad i=1, \ldots, n .
$$

Centring parameter As minimum and maximum centring parameters, we fix $\tau_{\min }=0.05$ and $\tau_{\max }=0.95$. In the first iteration we use $\tau_{0}=0.5$. Then, at each iteration $k$, in order to determine the centring parameter $\tau_{k}$, we perform the following operations:

$$
\tau_{k}=\max \left\{\left(1-a_{x, k-1}\right)^{5},\left(1-a_{z, k-1}\right)^{5}\right\},
$$

where $a_{x, k-1}, a_{z, k-1}$ are the step-lengths in directions $\Delta x, \Delta z$ of the previous iteration, respectively. Then we assign:

$$
\tau_{k}=\min \left\{\tau_{k}, \tau_{\max }\right\}
$$

and finally

$$
\tau_{k}=\max \left\{\tau_{k}, \tau_{\min }\right\}
$$

The latter is a heuristic which performs well in infeasible IPM implementations.
Step-length computation In order to calculate the step-length, we apply the fraction to the boundary rule, that is we compute the largest step-lengths to the boundary of the non-negative orthant, i.e.:

$$
\begin{equation*}
\alpha_{x, \max }=\min _{\Delta x^{i}<0}\left\{1,-\frac{x^{i}}{\Delta x^{i}}\right\}, \quad \alpha_{z, \max }=\min _{\Delta z^{i}<0}\left\{1,-\frac{z^{i}}{\Delta z^{i}}\right\}, \tag{2.33}
\end{equation*}
$$

and we set:

$$
\begin{equation*}
\alpha_{x}=\nu \alpha_{x, \max }, \quad \alpha_{z}=\nu \alpha_{z, \max } \tag{2.34}
\end{equation*}
$$

where $\nu \in(0,1)$ is set to $\nu=0.995$. The constant $\nu$ acts as a safeguard against bad directions. Taking a full step towards a direction can potentially push the iterates of the algorithm close to the boundary. This, in turn, can significantly slow down the convergence of the method. The primal variables $x, r$ are updated using the step-length $\alpha_{x}$ while the dual variables $y, s, z$ are updated using the step-length $\alpha_{z}$.

Termination Criteria Finally, the algorithm is terminated either if the number of maximum iterations specified by the user is reached, or when all the following three conditions are satisfied:

$$
\frac{\left\|c-A^{\top} y+Q x-z\right\|}{\|c\|+1} \leq \mathrm{tol}, \quad \frac{\|b-A x\|}{\|b\|+1} \leq \mathrm{tol}, \quad \mu \leq \mathrm{tol}
$$

where tol is the tolerance specified by the user.

### 2.4.3 Numerical results

We have made a particular effort to keep the implementation as simple as possible, so that the regularization effects can easily be seen and analyzed. For that reason, we applied scaling only to problems which required it to converge and this was needed only for 5 out of the 218 problems solved. On the other hand, no predictor-corrector technique was included. We tested our method on problems coming from the Netlib collection [130] as well as on a set of convex quadratic programming problems given in [116]. We present the numerical results, firstly for linear programming problems and then for quadratic programming ones. In order to demonstrate the effects of the proposed regularization method, we will compare it with an interior point method that uses a uniform regularization. This uniform regularization scheme can be interpreted as the application of a single iteration of a standard proximal point method, in contrast to the proposed method, which can be interpreted as the application of a single iteration of a generalized proximal point method. The experiments in this section were conducted on a PC with a 2.2 GHz Intel Core i5 processor (dual-core) and 4GB RAM, run under Linux operating system. The MATLAB version used was R2018a.

Linear programming problems As we have already stated, for linear programming problems we use only dual regularization, that is we set $R_{p}=0_{n, n}$ and $s=0_{n}$ in $\left(\mathrm{CQP}_{r}\right)-\left(\mathrm{CQD}_{r}\right)$. For that reason, we will compare our method with an algorithm that uses a uniform dual regularization, $R_{d}=\operatorname{reg}_{t h r, k} I_{m}, \forall k \geq 0$, where $\mathrm{reg}_{t h r, k}$ is updated as indicated in the previously presented Regularization paragraph. If $\mathcal{N}=\emptyset$, the two methods are exactly the same. Hence, the difference between the methods arises when some columns of the constraint matrix have entered the set $\mathcal{N}$. The tolerance used in the experiments for the linear programming problems was tol $=10^{-6}$. We will not use a smaller tolerance because our method does not have primal regularization. As a consequence, if some elements of $\Theta^{\mathcal{B}}$ become very large, this can create numerical instability if there is no primal regularization to keep such entries manageable in terms of machine precision. As an extra safeguard, when the factorization fails, we increase the uniform regularization value by a factor of 10 until the factorization is completed successfully. Finally, we set the maximum iterations of the method to be maxit $=200$. If this number is reached, the algorithm stops indicating that the optimal solution was not found. To conclude we use:

$$
\text { tol }=10^{-6}, \text { maxit }=200 .
$$

In order to present the importance of regularization, as well as the overall comparison of the two different regularization schemes, we include Figure 2.1, which contains the performance profiles, over the whole Netlib set, of three different methods. The green triangles correspond to the IPM with non-diagonal regularization. The red stars correspond to the IPM with uniform regularization, and finally the blue crosses correspond to an IPM without regularization. In

Figure 2.1a, we present the performance profiles with respect to the total time to convergence, while in Figure 2.1b the performance profiles with respect to the total number of iterations. The horizontal axis (in logarithmic scale), represents the performance ratio with respect to the best performance achieved by one of the three methods for each problem. For example, 2 in the horizontal axis is interpreted as: "what percentage of problems was solved by each method, in at most 2 times the best achieved time for each problem". The vertical axis shows the percentage of problems solved by each method for different values of the performance ratio. Efficiency is measured by the rate at which each of the lines increases, as the ratio increases. Robustness is measured by the maximum percentage achieved by each of the methods. For more information about performance profiles, we refer the reader to [58], where this benchmarking scheme was proposed.


Figure 2.1: Performance profiles over the Netlib test set.

By looking at Figure 2.1, one can observe the importance of regularization in terms of robustness of the method. Both IPM with non-diagonal regularization and IPM with uniform regularization solved all 96 problems of the Netlib collection. The former did so in 146.63 seconds and a total of 3,322 IPM iterations. The latter needed 165.72 seconds and a total of 3,442 iterations. In other words, the IPM using the proposed regularization, solved the whole set in $11.5 \%$ less time, requiring $3 \%$ less iterations. The computational benefits of the non-diagonal regularization become obvious in the larger instances of the Netlib collection. The IPM scheme that does not employ any regularization fails to solve $18.75 \%$ of the problems in the Netlib collection (we should note that the success rate of the non-regularized version can be improved either by means of pre-processing, or by significantly slowing down the method to ensure better stability). On the other hand, the IPM with non-diagonal regularization is more efficient in terms of time to convergence, when compared to the other two methods. Notice that this is not the case for the IPM using uniform regularization, which is less efficient than the other two methods for $70 \%$ of the problems. As expected, the IPM that does not use regularization converges in fewer iterations for most of the problems that it successfully solves. This is expected, since in the regularized schemes, we are perturbing the Newton system. Obviously, this perturbation is benign, in the sense that it allows us to significantly improve the robustness of the method.

We also include Table 2.1, in which the factorization times are compared when using non-diagonal and uniform regularization respectively, over the last four iterations of problems DFL001 and GREENBEA. The size of the respective constraint matrices also includes columns which were added to transform the problems to the standard form. Extra information, concerning the cardinality of the partition $\mathcal{N}$, the iteration count as well as the time needed to compute the Cholesky factorization of the system matrix at the respective iteration, has been collected in Table 2.1.

Table 2.1: Sparsity introduced from the non-diagonal regularization (linear programming)

| Name | $m$ | $n$ | Non-diagonal Reg. |  |  | Uniform Reg. |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  |  | Iter. | $\|\mathcal{N}\|$ | $t_{\text {fact }}$ (sec.) | Iter. | $t_{\text {fact }}$ (sec.) |
| DFL001 | 9,785 | 15,477 | 81 | 4,089 | 0.05 | 79 | 0.09 |
|  |  |  | 82 | 5,709 | 0.02 | 80 | 0.09 |
|  |  |  | 83 | 6,247 | 0.02 | 81 | 0.09 |
|  |  |  | 84 | 7,280 | 0.01 | 82 | 0.09 |
| GREENBEA | 3,770 | 5,973 | 66 | 2,512 | $3 \cdot 10^{-3}$ | 66 | 0.01 |
|  |  |  | 67 | 2,536 | $2 \cdot 10^{-3}$ | 67 | 0.01 |
|  |  |  | 68 | 1,210 | $8 \cdot 10^{-3}$ | 68 | 0.01 |
|  |  |  | 69 | 2,647 | $2 \cdot 10^{-3}$ | 69 | 0.01 |

Convex quadratic programming problems For this class of problems, we employ a primal-dual dynamic regularization. Hence, we will compare our method with an algorithm that uses a uniform primal-dual regularization. Such a method adds two uniform diagonal matrices $R_{p}=\operatorname{reg}_{t h r, k} I_{n}$ and $R_{d}=\operatorname{reg}_{t h r, k} I_{m}$ to the $(1,1)$ and $(2,2)$ blocks of the augmented system, respectively. This scheme can be interpreted as a single iteration of the proximal method of multipliers, in contrast to the proposed regularization scheme, which is a single iteration of a generalized proximal method of multipliers. As an extra safeguard, when the factorization fails, we increase the uniform regularization value by a factor of 10 until the factorization is completed successfully. The tolerance used in the experiments for this class of problems was tol $=10^{-8}$. As in the linear programming case, we set the maximum iterations of the method to be maxit $=200$. To conclude we use:

$$
\text { tol }=10^{-8}, \text { maxit }=200
$$

Following the linear programming case, we include Figure 2.2, which contains the performance profiles, over the whole Maros-Mészáros repository of convex quadratic programming problems ([116]), of three methods; the proposed IPM with non-diagonal regularization, the IPM with uniform primal-dual regularization and the same IPM but without regularization. In Figure 2.2a, a comparison of the total time to convergence is presented, while Figure 2.2 b contains the comparison of the total number of iterations.


Figure 2.2: Performance profiles over the Maros-Mészáros test set.

By looking at Figure 2.2, we can observe that as in the linear programming case, regularization seems crucial for the robustness of the method. In other words, one can observe that the IPM without regularization fails to solve $8.4 \%$ of the problems of this test set. However, in contrast to the linear programming case, the results do not demonstrate any significant advantage in terms of sparsity of linear systems achievable by the new regularization technique. This is a consequence of the fact that the problems under consideration are of small to medium size, while the overhead of setting up the partially reduced augmented system (2.23) is time consuming in MATLAB, where manipulating a permuted matrix is costly, due to MATLAB's default mechanism of storing matrices by columns. Nevertheless, both IPM with non-diagonal regularization and IPM with uniform regularization, solved all 122 problems. The former required 386.12 seconds and a total of 4,162 IPM iterations. The latter required 400.24 seconds and a total of 4,170 iterations. In other words, the non-diagonal scheme required $3 \%$ less time and a similar number of iterations, as compared to the uniform scheme, for this test set. In this case, the non-regularized IPM is more efficient than the other two methods for most of the problems that it solves. This indicates that the problems in this test set are very sensitive to perturbations. We should mention here, that the proposed tuning of the non-diagonal regularization is quite conservative. Hence, we would expect that one could improve the efficiency of such a method at the expense of its robustness.

As before, in order to illustrate the effect of the non-diagonal regularization in terms of factorization performance, we provide Table 2.2, in which the factorization times obtained when using non-diagonal and uniform regularization respectively are compared, over the last four iterations of problems LISWET1, FORPLAN and SHELL. The size of the constraint matrix in each case also includes columns which were added to transform the problem to the standard form. Information concerning the cardinality of the partition $\mathcal{N}$, the iteration count as well as the time needed to compute the $L D L^{\top}$ factorization of the system matrix at the respective iteration, is gathered in Table 2.2.

The examples presented in Table 2.2, confirm the previous observations drawn from the linear programming examples. In particular, we can observe the benefits of the proposed non-diagonal regularization, in terms of factorization performance. On the other hand, the convergence of the method does not seem to be

Table 2.2: Sparsity introduced from the non-diagonal regularization (quadratic programming)

| Name | $m$ | $n$ | Non-diagonal Reg. |  |  | Uniform Reg. |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  |  | Iter. | $\|\mathcal{N}\|$ | $t_{\text {fact }}$ (sec.) | Iter. | $t_{\text {fact }}(\mathrm{sec}$. |
| LISWET1 | 20,002 | 30,004 | 20 | 9,670 | 0.05 | 19 | 0.07 |
|  |  |  | 21 | 9,815 | 0.06 | 21 | 0.07 |
|  |  |  | 22 | 9935 | 0.06 | 22 | 0.08 |
|  |  |  | 23 | 9984 | 0.06 | 23 | 0.07 |
| FORPLAN | 186 | 517 | 62 | 199 | $10^{-3}$ | 62 | $4 \cdot 10^{-3}$ |
|  |  |  | 63 | 199 | $2 \cdot 10^{-3}$ | 63 | $4 \cdot 10^{-3}$ |
|  |  |  | 64 | 199 | $10^{-3}$ | 64 | $4 \cdot 10^{-3}$ |
|  |  |  | 65 | 199 | $2 \cdot 10^{-3}$ | 65 | $3 \cdot 10^{-3}$ |
| SHELL | 903 | 2,144 | 52 | 563 | $3 \cdot 10^{-3}$ | 52 | 0.01 |
|  |  |  | 53 | 565 | $3 \cdot 10^{-3}$ | 53 | 0.01 |
|  |  |  | 54 | 565 | $3 \cdot 10^{-3}$ | 54 | 0.01 |
|  |  |  | 55 | 721 | $3 \cdot 10^{-3}$ | 55 | 0.01 |

affected when big part of the columns of the constraint matrix lie in partition $\mathcal{N}$.

### 2.5 Conclusions

In this chapter, we derived a dynamic non-diagonal regularization scheme suitable for interior point methods. The proposed scheme is automatically tuned based on the properties of the problem, such that sufficiently large eigenvalues of the Newton system are perturbed insignificantly. The presence of non-diagonal terms in the regularization matrices allows us to introduce more sparsity in the linear system, solved to determine the Newton direction at each iteration of the interior point method. The regularization matrices can be computed expeditiously, enabling more efficient factorizations of the system matrix. Computational results demonstrate the efficiency of the approach, as well as the importance of regularization for numerical stability and thus robustness of the solver. The results also support the claim that the proposed rule, for tuning the regularization matrices based on the properties of the problem, produces a regularization which perturbs the system almost insignificantly while maintaining numerical stability.

## Chapter 3

## An IP-PMM for Convex QP

### 3.1 Introduction

In this chapter, we consider the primal-dual pair of linearly constrained convex quadratic programming problems, given in (CQP)-(CQD). We develop a pathfollowing primal-dual regularized IPM for solving convex quadratic programming problems. The algorithm is obtained by applying one or a few iterations of an infeasible primal-dual interior point method in order to solve sub-problems arising from the proximal method of multipliers (and hence is termed as IP-PMM). Under standard assumptions, we prove polynomial complexity of the algorithm and provide global convergence guarantees. To our knowledge, this is the first polynomial complexity result for a general primal-dual regularized IPM scheme. Notice that a complexity result is given for a primal regularized IPM for linear complementarity problems in [191]. However, the authors significantly alter the Newton directions, making their method very difficult to generalize and hard to achieve efficiency in practice.

An important feature of the presented method is that it makes use of only one penalty parameter, that is the logarithmic penalty one. The aforementioned penalty has been extensively studied and understood, and as a result, IPMs achieve fast and reliable convergence in practice. This is not the case for the penalty parameters of proximal methods. In other words, IP-PMM inherits the fast and reliable convergence properties of IPMs, as well as the strong convexity of the PMM sub-problems, hence improving the conditioning of the Newton system solved at each IPM iteration, while providing a reliable tuning for the penalty parameter, independently of the problem at hand. The proposed approach is implemented and its reliability is demonstrated through extensive experimentation. The implementation slightly deviates from the theory, however, most of the theoretical results are verified in practice. The main purpose of this chapter is to provide a reliable method that can be used for solving general convex quadratic problems, without the need of pre-processing, or of previous knowledge about the problems. The implemented method is supported by a novel theoretical result, indicating that regularization alleviates various issues arising in IPMs, without affecting their important worst-case polynomial complexity. As a by-product of the theory, an implementable infeasibility detection mechanism is also derived
and tested in practice.
Before completing this section, let us introduce some notation that is used in the rest of this chapter. An optimal solution to the pair (CQP)-(CQD) will be denoted as $\left(x^{*}, y^{*}, z^{*}\right)$. Optimal solutions of different primal-dual pairs will be denoted using an appropriate subscript, in order to distinguish them. For example, we use the notation $\left(x_{r}^{*}, y_{r}^{*}, z_{r}^{*}\right)$, for representing an optimal solution for a PMM sub-problem. The subscript is employed for distinguishing the "regularized" solution, from the solution of the initial problem, that is $\left(x^{*}, y^{*}, z^{*}\right)$.

The rest of this chapter is organized as follows. In Section 3.2, we provide the algorithmic framework of the method. Consequently, in Section 3.3, we prove polynomial complexity of the algorithm, and global convergence is established. In Section 3.4, a necessary condition for infeasibility is derived, which is later used to construct an infeasibility detection mechanism. Numerical results of the implemented method are presented and discussed in Section 3.5. Finally, we derive some conclusions in Section 3.6.

### 3.2 Algorithmic framework

In this section, we will merge the proximal method of multipliers with an infeasible interior point method. For that purpose, assume that, at some iteration $k$ of the method, we have available an estimate $\eta_{k}$ for a Lagrange multiplier vector. Similarly, we denote by $\zeta_{k}$ the estimate of a primal solution. We define the proximal penalty function that has to be minimized at the $k$-th iteration of PMM, for solving (CQP), given the estimates $\eta_{k}, \zeta_{k}$, as:

$$
\begin{equation*}
\mathcal{L}_{\mu_{k}}^{P M M}\left(x ; \zeta_{k}, \eta_{k}\right):=c^{\top} x+\frac{1}{2} x^{\top} Q x-\eta_{k}^{\top}(A x-b)+\frac{1}{2 \mu_{k}}\|A x-b\|_{2}^{2}+\frac{\mu_{k}}{2}\left\|x-\zeta_{k}\right\|_{2}^{2}, \tag{3.1}
\end{equation*}
$$

with $\left\{\mu_{k}\right\}$ some non-increasing sequence of positive penalty parameters. In order to solve the PMM sub-problem (1.7) (where $f(x)=c^{\top} x+(1 / 2) x^{\top} Q x$ ), we will apply one (or a few) iterations of an infeasible IPM. To that end, we alter (3.1), by including logarithmic barriers, that is:

$$
\begin{equation*}
\mathcal{L}_{\mu_{k}}^{I P-P M M}\left(x ; \zeta_{k}, \eta_{k}\right):=\mathcal{L}_{\mu_{k}}^{P M M}\left(x ; \zeta_{k}, \eta_{k}\right)-\mu_{k} \sum_{j=1}^{n} \ln x^{j}, \tag{3.2}
\end{equation*}
$$

and we treat $\mu_{k}$ as the barrier parameter. In order to form the optimality conditions of this sub-problem, we equate the gradient of $\mathcal{L}_{\mu_{k}}^{I P-P M M}\left(\cdot ; \eta_{k}, \zeta_{k}\right)$ to the zero vector, i.e.:

$$
c+Q x-A^{\top} \eta_{k}+\frac{1}{\mu_{k}} A^{\top}(A x-b)+\mu_{k}\left(x-\zeta_{k}\right)-\mu_{k} X^{-1} e_{n}=0_{n} .
$$

Following the developments in [5], we can define the variables $y=\eta_{k}-\frac{1}{\mu_{k}}(A x-$ $b$ ) and $z=\mu_{k} X^{-1} e_{n}$, to get the following (equivalent) system of equations (first-
order optimality conditions):

$$
\left[\begin{array}{c}
c+Q x-A^{\top} y-z+\mu_{k}\left(x-\zeta_{k}\right)  \tag{3.3}\\
A x+\mu_{k}\left(y-\eta_{k}\right)-b \\
X z-\mu_{k} e_{n}
\end{array}\right]=\left[\begin{array}{c}
0_{n} \\
0_{m} \\
0_{n}
\end{array}\right]
$$

Let us introduce some notation that will be used later. Given two arbitrary vectors $b \in \mathbb{R}^{m}, c \in \mathbb{R}^{n}$, we define the following semi-norm:

$$
\begin{equation*}
\|(b, c)\|_{\mathcal{A}}:=\min _{x, y, z}\left\{\|(x, z)\|_{2}: A x=b,-Q x+A^{\top} y+z=c\right\} . \tag{3.4}
\end{equation*}
$$

This semi-norm has been used before in [121], as a way to measure infeasibility for the case of linear programming problems $\left(Q=0_{n, n}\right)$. For a discussion on the properties of the aforementioned semi-norm, as well as how one can evaluate it (using the QR factorization of $A$ ), the reader is referred to [121, Section 4].

Starting point Let us define the starting point used within IP-PMM. For that, we set $\left(x_{0}, z_{0}\right)=\rho\left(e_{n}, e_{n}\right)$, for some $\rho>0$. We also set $y_{0}$ to some arbitrary vector (e.g. $y_{0}=e_{m}$ ), such that $\left\|y_{0}\right\|_{\infty}=\boldsymbol{O}(1)$, and $\mu_{0}=\frac{x_{0}^{\top} z_{0}}{n}$. Then we have:

$$
\begin{equation*}
A x_{0}=b+\bar{b},-Q x_{0}+A^{\top} y_{0}+z_{0}=c+\bar{c}, \zeta_{0}=x_{0}, \eta_{0}=y_{0} . \tag{3.5}
\end{equation*}
$$

for some vectors $\bar{b}, \bar{c}$.
Neighbourhood We develop a path-following method. Hence, we have to describe a neighbourhood in which the iterations of the method should lie. However, unlike typical path-following methods, we define a family of neighbourhoods that depend on the PMM sub-problem parameters.

Given the starting point in (3.5), penalty $\mu_{k}$, and estimates $\eta_{k}, \zeta_{k}$, we define the following regularized set of centers:

$$
\mathcal{C}_{\mu_{k}}^{+}\left(\zeta_{k}, \eta_{k}\right):=\left\{(x, y, z) \in \mathcal{C}_{\mu_{k}}\left(\zeta_{k}, \eta_{k}\right):(x, z)>\left(0_{n}, 0_{n}\right), X z=\mu_{k} e_{n}\right\}
$$

where

$$
\mathcal{C}_{\mu_{k}}\left(\zeta_{k}, \eta_{k}\right):=\left\{\begin{array}{c}
A x+\mu_{k}\left(y-\eta_{k}\right)=b+\frac{\mu_{k}}{\mu_{0}} \bar{b} \\
(x, y, z): \\
-Q x+A^{\top} y+z-\mu_{k}\left(x-\zeta_{k}\right)=c+\frac{\mu_{k}}{\mu_{0}} \bar{c}
\end{array}\right\}
$$

and $\bar{b}, \bar{c}$ are as in (3.5). The term set of centers originates from [121], where a similar set is studied.

In order to enlarge the previous set, we define the following set:

$$
\tilde{\mathcal{C}}_{\mu_{k}}\left(\zeta_{k}, \eta_{k}\right):=\left\{\begin{array}{c}
A x+\mu_{k}\left(y-\eta_{k}\right)=b+\frac{\mu_{k}}{\mu_{0}}\left(\bar{b}+\tilde{b}_{k}\right) \\
(x, y, z):-Q x+A^{\top} y+z-\mu_{k}\left(x-\zeta_{k}\right) \stackrel{\mu_{k}}{=} c+\frac{\mu_{k}}{\mu_{0}}\left(\bar{c}+\tilde{c}_{k}\right) \\
\left\|\left(\tilde{b}_{k}, \tilde{c}_{k}\right)\right\|_{2} \leq K_{N},\left\|\left(\tilde{b}_{k}, \tilde{c}_{k}\right)\right\|_{\mathcal{A}} \leq \gamma_{\mathcal{A}} \rho
\end{array}\right\},
$$

where $K_{N}>0$ is a constant, $\gamma_{\mathcal{A}} \in(0,1)$, and $\rho>0$ is as defined in the starting point. The vectors $\tilde{b}_{k}$ and $\tilde{c}_{k}$ represent the current scaled (by $\mu_{0} / \mu_{k}$ ) infeasibility
and vary depending on the iteration $k$. In particular, these vectors can be formally defined recursively, depending on the iterations of IP-PMM. However, such a definition is not necessary for the developments to follow. In essence, the only requirement is that these scaled infeasibility vectors are bounded above by some constants, with respect to the 2-norm as well as the semi-norm defined in (3.4). Using the latter set, we are now ready to define a family of neighbourhoods:

$$
\begin{align*}
\mathcal{N}_{\mu_{k}}\left(\zeta_{k}, \eta_{k}\right):= & \left\{(x, y, z) \in \tilde{\mathcal{C}}_{\mu_{k}}\left(\zeta_{k}, \eta_{k}\right):\right.  \tag{3.6}\\
& \left.(x, z)>\left(0_{n}, 0_{n}\right), x^{i} z^{i} \geq \gamma_{\mu} \mu_{k}, i \in\{1, \ldots, n\}\right\},
\end{align*}
$$

where $\gamma_{\mu} \in(0,1)$ is a constant preventing component-wise complementarity products from approaching zero faster than $\mu_{k}=\left(x_{k}^{\top} z_{k}\right) / n$. Obviously, the starting point defined in (3.5) belongs to the neighbourhood $\mathcal{N}_{\mu_{0}}\left(\zeta_{0}, \eta_{0}\right)$, with $\left(\tilde{b}_{0}, \tilde{c}_{0}\right)=\left(0_{m}, 0_{n}\right)$. Notice from the definition of the neighbourhood, that it depends on the choice of the constants $K_{N}, \gamma_{\mathcal{A}}, \gamma_{\mu}$. However, as the neighbourhood also depends on the parameters $\mu_{k}, \eta_{k}, \zeta_{k}$, we omit the dependence on the constants, for simplicity of notation.

Newton system At every IP-PMM iteration, we approximately solve a perturbed form of the conditions in (3.3), by applying a variation of Newton method. In particular, we form the Jacobian of the left-hand side of (3.3) and we perturb the right-hand side of the Newton equation as follows:

$$
\begin{align*}
& {\left[\begin{array}{ccc}
-\left(Q+\mu_{k} I_{n}\right) & A^{\top} & I \\
A & \mu_{k} I_{m} & 0_{m, n} \\
Z_{k} & 0_{n, m} & X_{k}
\end{array}\right]\left[\begin{array}{l}
\Delta x_{k} \\
\Delta y_{k} \\
\Delta z_{k}
\end{array}\right]=} \\
& \quad-\left[\begin{array}{c}
-\left(c+\frac{\tau_{k} \mu_{k}}{\mu_{0}} \bar{c}\right)-Q x_{k}+A^{\top} y_{k}+z_{k}-\tau_{k} \mu_{k}\left(x_{k}-\zeta_{k}\right) \\
A x_{k}+\tau_{k} \mu_{k}\left(y_{k}-\eta_{k}\right)-\left(b+\frac{\tau_{k} \mu_{k}}{\mu_{0}} \bar{b}\right) \\
X_{k} Z_{k} e_{n}-\tau_{k} \mu_{k} e_{n}
\end{array}\right], \tag{3.7}
\end{align*}
$$

where $\bar{b}, \bar{c}$ are as in (3.5). Notice that we perturb the right-hand side of the Newton system in order to ensure that the iterates remain in the neighbourhood (3.6), while trying to reduce the value of the penalty (barrier) parameter $\mu_{k}$.

We are now able to derive Algorithm IP-PMM-QP, summarizing the proposed interior point-proximal method of multipliers. We will prove polynomial complexity of this scheme in the next section, under standard assumptions.

Notice, in Algorithm IP-PMM-QP, that we force $\tau$ to be less than 0.5 . This value is set, without loss of generality, for simplicity of exposition. Similarly, in the choice of the step-length, we require that $\mu_{k}(\alpha) \leq(1-0.01 \alpha) \mu_{k}$. The constant 0.01 is chosen for ease of presentation. It depends on the choice of the maximum value of $\tau$. The constants $K_{N}, \gamma_{\mathcal{A}}, \gamma_{\mu}$, are used in the definition of the neighbourhood in (3.6). Their values can be considered to be arbitrary. The input tol, represents the error tolerance (chosen by the user). The terminating conditions require the Euclidean norm of primal and dual infeasibility, as well as complementarity, to be less than this tolerance. In such a case, we accept the iterate as a solution triple. The estimates $\eta, \zeta$ are not updated if primal or dual infeasibility are not both sufficiently decreased. In this case, we keep

```
Algorithm IP-PMM-QP Interior Point-Proximal Method of Multipliers
Input: \(A, Q, b, c\), tol.
Parameters: \(0<\tau_{\min } \leq \tau_{\max } \leq 0.5, K_{N}, \bar{K}, k^{\dagger}>0,0<\gamma_{\mathcal{A}}, \gamma_{\mu}<1, \bar{k}=0\).
Starting point: Set as in (3.5).
    for \((k=0,1,2, \cdots)\) do
        if \(\left(\left(\left\|A x_{k}-b\right\|_{2}<\right.\right.\) tol \() \wedge\left(\left\|c+Q x_{k}-A^{\top} y_{k}-z_{k}\right\|_{2}<\right.\) tol \() \wedge\left(\mu_{k}<\right.\) tol \(\left.)\right)\) then
            return \(\left(x_{k}, y_{k}, z_{k}\right)\).
        else
            Choose \(\tau_{k} \in\left[\tau_{\text {min }}, \tau_{\text {max }}\right]\) and solve (3.7).
            Choose step-length \(\alpha_{k}\), as the largest \(\alpha \in(0,1]\) such that:
                \(\mu_{k}(\alpha) \leq(1-0.01 \alpha) \mu_{k}\),
                    and, \(\left(x_{k}+\alpha_{k} \Delta x_{k}, y_{k}+\alpha_{k} \Delta y_{k}, z_{k}+\alpha_{k} \Delta z_{k}\right) \in \mathcal{N}_{\mu_{k}(\alpha)}\left(\zeta_{k}, \eta_{k}\right)\),
                where \(\mu_{k}(\alpha)=\frac{\left(x_{k}+\alpha_{k} \Delta x_{k}\right)^{\top}\left(z_{k}+\alpha_{k} \Delta z_{k}\right)}{n}\).
            Set \(\left(x_{k+1}, y_{k+1}, z_{k+1}\right)=\left(x_{k}+\alpha_{k} \Delta x_{k}, y_{k}+\alpha_{k} \Delta y_{k}, z_{k}+\alpha_{k} \Delta z_{k}\right)\).
            Update \(\mu_{k+1}=\frac{x_{k+1}^{\top} z_{k+1}}{n}\).
            Let \(r_{p}=A x_{k+1}-\left(b+\frac{\mu_{k+1}}{\mu_{0}} \bar{b}\right), r_{d}=\left(c+\frac{\mu_{k+1}}{\mu_{0}} \bar{c}\right)+Q x_{k+1}-A^{\top} y_{k+1}-z_{k+1}\).
            Set \(\left(\zeta_{k+1}, \eta_{k+1}\right)=\left(\zeta_{k}, \eta_{k}\right)\).
            if \(\left(\left(\left\|\left(r_{p}, r_{d}\right)\right\|_{2} \leq K_{N} \frac{\mu_{k+1}}{\mu_{0}}\right) \wedge\left(\left\|\left(r_{p}, r_{d}\right)\right\|_{\mathcal{A}} \leq \gamma_{\mathcal{A}} \rho \frac{\mu_{k+1}}{\mu_{0}}\right)\right)\) then
            if \(\left(\left\|\left(x_{k+1}, y_{k+1}\right)\right\|_{\infty} \leq \bar{K}\right)\) then
                \(\left(\zeta_{k+1}, \eta_{k+1}\right)=\left(x_{k+1}, y_{k+1}\right)\).
            else if \(\left(\bar{k} \leq k^{\dagger}\right)\) then
                \(\left(\zeta_{k+1}, \eta_{k+1}\right)=\left(x_{k+1}, y_{k+1}\right), \bar{k}=\bar{k}+1\).
            end if
        end if
        end if
    end for
```

the estimates constant while continuing decreasing the penalty parameter $\mu_{k}$. Following the usual practice with proximal and augmented Lagrangian methods, we accept a new estimate when the respective residual is sufficiently decreased. However, the algorithm requires the evaluation of the semi-norm defined in (3.4), at every iteration. While this is not practical, it can be achieved in polynomial time, with respect to the size of the problem. For a detailed discussion on this, the reader is referred to [121, Section 4].

Let us notice that Algorithm IP-PMM-QP deviates from standard IPM schemes due to the solution of a different Newton system, as well as due to the possible updates of the proximal estimates, i.e. $\zeta_{k}$ and $\eta_{k}$. Notice that when these estimates are updated, the neighbourhood in (3.6) changes as well, since it is parametrized by them. Intuitively, when this happens, the algorithm accepts the current iterate as a sufficiently accurate solution to the associated PMM sub-problem. However,
as we will see in Section 3.3, it is not necessary for these estimates to converge to a primal-dual solution, for Algorithm IP-PMM-QP to converge. Instead, it suffices to ensure that these estimates will remain bounded. To that end, we allow only $k^{\dagger}$ arbitrary updates (for some $k^{\dagger}=\boldsymbol{O}(1)$ ) as well as any update that belongs to a large ball with radius $\bar{K}=\boldsymbol{O}(1)$ (where this value is chosen arbitrarily). In light of this, Algorithm IP-PMM-QP is not studied as an inner-outer scheme, but rather as a standard IPM scheme. We will return to this point at the end of Section 3.3.

### 3.3 Convergence analysis of IP-PMM

In this section we prove polynomial complexity and global convergence of Algorithm IP-PMM-QP. The proof follows by induction on the iterations of IPPMM. That is, given an iterate $\left(x_{k}, y_{k}, z_{k}\right)$ at an arbitrary iteration $k$, we prove that the next iterate belongs to the appropriate neighbourhood required by the algorithm. In turn, this allows us to prove global and polynomial convergence of IP-PMM-QP. An outline of the proof can be briefly explained as follows:

- Initially, we present some technical results in Lemmas 3.3.1-3.3.3 which are required for the analysis throughout this section.
- In turn, we prove boundedness of the iterates $\left(x_{k}, y_{k}, z_{k}\right)$ of IP-PMM in Lemma 3.3.4. In particular we show that $\left\|\left(x_{k}, y_{k}, z_{k}\right)\right\|_{2}=\boldsymbol{O}(n)$ and $\left\|\left(x_{k}, z_{k}\right)\right\|_{1}=\boldsymbol{O}(n)$ for every $k \geq 0$.
- Then, we prove boundedness of the Newton direction computed at every IP-PMM iteration in Lemma 3.3.5. More specifically, we prove that $\left\|\left(\Delta x_{k}, \Delta y_{k}, \Delta z_{k}\right)\right\|_{2}=\boldsymbol{O}\left(n^{3}\right)$ for every $k \geq 0$.
- In Lemma 3.3.6 we prove the existence of a positive step-length $\bar{\alpha}$ so that the new iterate of IP-PMM, $\left(x_{k+1}, y_{k+1}, z_{k+1}\right)$, belongs to the updated neighbourhood $\mathcal{N}_{\mu_{k+1}}\left(\zeta_{k+1}, \eta_{k+1}\right)$, for every $k \geq 0$. In particular, we show that $\bar{\alpha} \geq \frac{\bar{\kappa}}{n^{4}}$, where $\bar{\kappa}$ is a constant independent of $n$ and $m$.
- Q-linear convergence of the barrier parameter $\mu_{k}$ (to zero) is established in Theorem 3.3.1.
- The polynomial complexity of IP-PMM is then proved in Theorem 3.3.2, showing that it converges to an $\epsilon$-accurate solution in at most $\boldsymbol{O}\left(n^{4}\left|\log \left(\frac{1}{\epsilon}\right)\right|\right)$ steps.
- Finally, global converge to an optimal solution of (CQP)-(CQD) is established in Theorem 3.3.3.

For the rest of this section, we will make use of the following two assumptions, which are commonly employed when analyzing the complexity of an IPM.

Assumption 1. There exists an optimal solution $\left(x^{*}, y^{*}, z^{*}\right)$ for the primal-dual pair (CQP)-(CQD), such that $\left\|x^{*}\right\|_{\infty} \leq K^{*},\left\|y^{*}\right\|_{\infty} \leq K^{*}$ and $\left\|z^{*}\right\|_{\infty} \leq K^{*}$, for some constant $K^{*} \geq 0$, independent of $n$ and $m$.

Assumption 2. The constraint matrix of (CQP) has full row rank, that is $\operatorname{rank}(A)=m$. Furthermore, we assume that there exist constants $K_{A, 1}>0$, $K_{A, 2}>0, K_{Q}>0$ and $K_{r}>0$, independent of $n$ and $m$, such that:

$$
\sigma_{\min }(A) \geq K_{A, 1}, \quad \sigma_{\max }(A) \leq K_{A, 2}, \quad \lambda_{\max }(Q) \leq K_{Q}, \quad\|(c, b)\|_{\infty} \leq K_{r}
$$

Note that the independence of the previous constants from the problem dimensions is assumed for simplicity of exposition; this is a common practice when analyzing the complexity of interior point methods. If these constants depend polynomially on $n$ (or $m$ ), the analysis still holds by suitably altering the worstcase polynomial bound for the number of iterations of the algorithm.

Let us now use the properties of the proximal operator defined in (1.10).
Lemma 3.3.1. Given Assumption 1, and for all $\eta \in \mathbb{R}^{m}, \zeta \in \mathbb{R}^{n}$ and $0 \leq \mu<\infty$, there exists a unique pair $\left(x_{r}^{*}, y_{r}^{*}\right)$, such that $\left(x_{r}^{*}, y_{r}^{*}\right)=P(\zeta, \eta)$, and

$$
\begin{equation*}
\left\|\left(x_{r}^{*}, y_{r}^{*}\right)-\left(x^{*}, y^{*}\right)\right\|_{2} \leq\left\|(\zeta, \eta)-\left(x^{*}, y^{*}\right)\right\|_{2} \tag{3.8}
\end{equation*}
$$

where $P(\cdot)$ is defined as in (1.10) and $\left(x^{*}, y^{*}\right)$ is the same as in Assumption 1.
Proof. We know that $P(\cdot, \cdot)$ is single-valued and non-expansive (see [147]), and hence there exists a unique pair $\left(x_{r}^{*}, y_{r}^{*}\right)$, such that $\left(x_{r}^{*}, y_{r}^{*}\right)=P(\zeta, \eta)$, for all $\eta, \zeta$ and $0 \leq \mu<\infty$. Given the optimal triple of Assumption 1, we can use the non-expansiveness of $P(\cdot)$ in (1.10), to show that:

$$
\left\|P(\zeta, \eta)-P\left(x^{*}, y^{*}\right)\right\|_{2}=\left\|\left(x_{r}^{*}, y_{r}^{*}\right)-\left(x^{*}, y^{*}\right)\right\|_{2} \leq\left\|(\zeta, \eta)-\left(x^{*}, y^{*}\right)\right\|_{2},
$$

where we used the fact that $P\left(x^{*}, y^{*}\right)=\left(x^{*}, y^{*}\right)$, which follows directly from (1.8), as we can see that $\left(0_{m}, 0_{n}\right) \in T_{\mathcal{L}}\left(x^{*}, y^{*}\right)$. This completes the proof.

In the next lemma we bound the solution of every PMM sub-problem encountered by Algorithm IP-PMM-QP. We also establish uniform bounds for the sequence $\left\{\left\|\left(\zeta_{k}, \eta_{k}\right)\right\|\right\}$.

Lemma 3.3.2. Given Assumptions 1, 2, there exists a triple $\left(x_{r_{k}}^{*}, y_{r_{k}}^{*}, z_{r_{k}}^{*}\right)$, satisfying:

$$
\begin{align*}
A x_{r_{k}}^{*}+\mu\left(y_{r_{k}}^{*}-\eta_{k}\right)-b & =0_{m}, \\
-c-Q x_{r_{k}}^{*}+A^{\top} y_{r_{k}}^{*}+z_{r_{k}}^{*}-\mu\left(x_{r_{k}}^{*}-\zeta_{k}\right) & =0_{n},  \tag{3.9}\\
\left(x_{r_{k}}^{*}\right)^{\top}\left(z_{r_{k}}^{*}\right) & =0,
\end{align*}
$$

with $\left\|\left(x_{r_{k}}^{*}, y_{r_{k}}^{*}, z_{r_{k}}^{*}\right)\right\|_{2}=\boldsymbol{O}(\sqrt{n})$, for all $\eta_{k} \in \mathbb{R}^{m}, \zeta_{k} \in \mathbb{R}^{n}$ produced by Algorithm IP-PMM-QP, and any $\mu \in[0, \infty)$. Moreover, we have that $\left\|\left(\zeta_{k}, \eta_{k}\right)\right\|_{2}=\boldsymbol{O}(\sqrt{n})$, for all $k \geq 0$.

Proof. We prove the claim by induction on the iterates, $k \geq 0$, of Algorithm IP-PMM-QP. At iteration $k=0$, we have that $\eta_{0}=y_{0}$ and $\zeta_{0}=x_{0}$. But from the construction of the starting point in (3.5), we know that $\left\|\left(x_{0}, y_{0}\right)\right\|_{2}=\boldsymbol{O}(\sqrt{n})$. Hence, $\left\|\left(\zeta_{0}, \eta_{0}\right)\right\|_{2}=\boldsymbol{O}(\sqrt{n})$ (assuming that $n>m$ ). From Lemma 3.3.1, we
know that there exists a unique pair $\left(x_{r_{0}}^{*}, y_{r_{0}}^{*}\right)$ such that:

$$
\left(x_{r_{0}}^{*}, y_{r_{0}}^{*}\right)=P_{0}\left(\zeta_{0}, \eta_{0}\right), \quad \text { and } \quad\left\|\left(x_{r_{0}}^{*}, y_{r_{0}}^{*}\right)-\left(x^{*}, y^{*}\right)\right\|_{2} \leq\left\|\left(\zeta_{0}, \eta_{0}\right)-\left(x^{*}, y^{*}\right)\right\|_{2} .
$$

Using the triangular inequality, and combining the latter inequality with our previous observations, as well as Assumption 1, yields that

$$
\left\|\left(x_{r_{0}}^{*}, y_{r_{0}}^{*}\right)\right\|_{2} \leq 2\left\|\left(x^{*}, y^{*}\right)\right\|_{2}+\left\|\left(\zeta_{0}, \eta_{0}\right)\right\|_{2}=\boldsymbol{O}(\sqrt{n}),
$$

and the bound is uniform, since $\left\|\left(x^{*}, y^{*}\right)\right\|_{\infty} \leq K^{*}$ (from Assumption 1) and $\left\|\left(\zeta_{0}, \eta_{0}\right)\right\|_{\infty}=\boldsymbol{O}(1)$ (from (3.5)). From the definition of the operator in (1.11), we know that:

$$
\begin{aligned}
-c-Q x_{r_{0}}^{*}+A^{\top} y_{r_{0}}^{*}-\mu\left(x_{r_{0}}^{*}-\zeta_{k}\right) & \in \partial \delta_{+}\left(x_{r_{0}}^{*}\right), \\
A x_{r_{0}}^{*}+\mu\left(y_{r_{0}}^{*}-\eta_{k}\right)-b & =0_{m},
\end{aligned}
$$

where $\partial\left(\delta_{+}(\cdot)\right)$ is the sub-differential of the indicator function defined in (1.9). Hence, we know that there must exist $-z_{r_{0}}^{*} \in \partial \delta_{+}\left(x_{r_{0}}^{*}\right)$ (and hence, $z_{r_{0}}^{*} \geq 0_{n}$, $\left.\left(x_{r_{0}}^{*}\right)^{\top}\left(z_{r_{0}}^{*}\right)=0\right)$, such that:

$$
z_{r_{0}}^{*}=c+Q x_{r_{0}}^{*}-A^{\top} y_{r_{0}}^{*}+\mu\left(x_{r_{0}}^{*}-\zeta_{k}\right),\left(x_{r_{0}}^{*}\right)^{\top}\left(z_{r_{0}}^{*}\right)=0,\left\|z_{r_{0}}^{*}\right\|_{2}=\boldsymbol{O}(\sqrt{n}),
$$

where $\left\|z_{r_{0}}^{*}\right\|_{2}=\boldsymbol{O}(\sqrt{n})$ follows from Assumption 2, combined with $\left\|\left(x_{0}, y_{0}\right)\right\|_{2}=$ $\boldsymbol{O}(\sqrt{n})$.

Let us now consider an arbitrary iteration $k$ of Algorithm IP-PMM-QP. There are two cases for the subsequent iteration:

1. The proximal estimates are updated, that is $\left(\zeta_{k+1}, \eta_{k+1}\right)=\left(x_{k+1}, y_{k+1}\right)$, or
2. the proximal estimates stay the same, i.e. $\left(\zeta_{k+1}, \eta_{k+1}\right)=\left(\zeta_{k}, \eta_{k}\right)$.

Case 1. We know by construction that this case can only occur if the following condition is satisfied:

$$
\left\|\left(r_{p}, r_{d}\right)\right\|_{2} \leq K_{N} \frac{\mu_{k+1}}{\mu_{0}}
$$

where $r_{p}, r_{d}$ are defined in Algorithm IP-PMM-QP. However, from the neighbourhood conditions in (3.6), we know that:

$$
\left\|\left(r_{p}+\mu_{k+1}\left(y_{k+1}-\eta_{k}\right), r_{d}+\mu_{k+1}\left(x_{k+1}-\zeta_{k}\right)\right)\right\|_{2} \leq K_{N} \frac{\mu_{k+1}}{\mu_{0}} .
$$

Combining the last two inequalities by applying the triangular inequality, and using the properties of Algorithm IP-PMM-QP recursively, we obtain

$$
\left\|\left(x_{k+1}, y_{k+1}\right)\right\|_{2} \leq \frac{2 K_{n}}{\mu_{0}}+\left\|\left(\zeta_{k}, \eta_{k}\right)\right\|_{2} \leq \max \left\{k^{\dagger} \frac{2 K_{n}}{\mu_{0}}+\left\|\left(\zeta_{0}, \eta_{0}\right)\right\|, \bar{K} \sqrt{n}\right\}=\boldsymbol{O}(\sqrt{n}) .
$$

Hence, $\left\|\left(\zeta_{k+1}, \eta_{k+1}\right)\right\|_{2}=\boldsymbol{O}(\sqrt{n})$. Then, we can invoke Lemma 3.3.1, with $\eta=$ $\eta_{k+1}, \zeta=\zeta_{k+1}$ and $\mu=\mu_{k+1}$, which implies that:

$$
\left\|\left(x_{r_{k+1}}^{*}, y_{r_{k+1}}^{*}\right)-\left(x^{*}, y^{*}\right)\right\|_{2} \leq\left\|\left(\zeta_{k+1}, \eta_{k+1}\right)-\left(x^{*}, y^{*}\right)\right\|_{2} .
$$

A simple manipulation yields that $\left\|\left(x_{r_{k+1}}^{*}, y_{r_{k+1}}^{*}\right)\right\|_{2}=\boldsymbol{O}(\sqrt{n})$. As before, we use (1.11) alongside Assumption 2 to show the existence of $-z_{r_{k+1}}^{*} \in \partial \delta_{+}\left(x_{r_{k+1}}^{*}\right)$, such that the triple $\left(x_{r_{k+1}}^{*}, y_{r_{k+1}}^{*}, z_{r_{k+1}}^{*}\right)$ satisfies (3.9) with $\left\|z_{r_{k+1}}^{*}\right\|_{2}=\boldsymbol{O}(\sqrt{n})$.

Case 2. In this case, we have $\left(\zeta_{k+1}, \eta_{k+1}\right)=\left(\zeta_{k}, \eta_{k}\right)$, and using the properties of Algorithm IP-PMM-QP we can show that $\left\|\left(\zeta_{k}, \eta_{k}\right)\right\|_{2}=\boldsymbol{O}(\sqrt{n})$, for any $k$. The same reasoning as before implies the existence of a triple $\left(x_{r_{k+1}}^{*}, y_{r_{k+1}}^{*}, z_{r_{k+1}}^{*}\right)$ satisfying (3.9), with $\left\|\left(x_{r_{k+1}}^{*}, y_{r_{k+1}}^{*}, z_{r_{k+1}}^{*}\right)\right\|_{2}=\boldsymbol{O}(\sqrt{n})$.

In the following lemma we define an auxiliary point solving a particular parametrized nonlinear system of equations. This point is subsequently utilized in Lemma 3.3.4, and in fact it allows us to show boundedness of the iterates $\left(x_{k}, y_{k}, z_{k}\right)$ of Algorithm IP-PMM-QP.

Lemma 3.3.3. Given Assumptions 1, 2, $\eta_{k}$ and $\zeta_{k}$ produced at an arbitrary iteration $k \geq 0$ of Algorithm IP-PMM-QP and any $\mu \in[0, \infty)$, there exists a triple $(\tilde{x}, \tilde{y}, \tilde{z})$ which satisfies the following system of equations:

$$
\begin{align*}
A \tilde{x}+\mu \tilde{y} & =b+\bar{b}+\mu \eta_{k}+\tilde{b}_{k}, \\
-Q \tilde{x}+A^{\top} \tilde{y}+\tilde{z}-\mu \tilde{x} & =c+\bar{c}-\mu \zeta_{k}+\tilde{c}_{k},  \tag{3.10}\\
\tilde{X} \tilde{z} & =\theta e_{n},
\end{align*}
$$

for some arbitrary $\theta>0(\theta=\boldsymbol{\Theta}(1))$, with $(\tilde{x}, \tilde{z}) \geq \xi\left(e_{n}, e_{n}\right)(\xi=\boldsymbol{\Theta}(1))$ and $\|(\tilde{x}, \tilde{y}, \tilde{z})\|_{2}=\boldsymbol{O}(\sqrt{n})$, where $\tilde{b}_{k}, \tilde{c}_{k}$ are defined in (3.6), while $\bar{b}, \bar{c}$ are defined with the starting point in (3.5).

Proof. Let $k \geq 0$ denote an arbitrary iteration of Algorithm IP-PMM-QP. Let also $\bar{b}, \bar{c}$ as defined in (3.5), and $\tilde{b}_{k}, \tilde{c}_{k}$, as defined in the neighbourhood conditions in (3.6). Given an arbitrary positive constant $\theta>0$, we consider the following barrier primal-dual pair:

$$
\begin{gather*}
\min _{x}\left(\left(c+\bar{c}+\tilde{c}_{k}\right)^{\top} x+\frac{1}{2} x^{\top} Q x-\theta \sum_{j=1}^{n} \ln x^{j}\right) \text {, s.t. } A x=b+\bar{b}+\tilde{b}_{k},  \tag{3.11}\\
\max _{x, y, z}\left(\left(b+\bar{b}+\tilde{b}_{k}\right)^{\top} y-\frac{1}{2} x^{\top} Q x+\theta \sum_{j=1}^{n} \ln z^{j}\right) \text {, s.t. }-Q x+A^{\top} y+z=c+\bar{c}+\tilde{c}_{k} . \tag{3.12}
\end{gather*}
$$

Let us now define the following triple:

$$
(\hat{x}, \hat{y}, \hat{z}):=\arg \min _{(x, y, z)}\left\{\|(x, z)\|_{2}: A x=\tilde{b}_{k},-Q x+A^{\top} y+z=\tilde{c}_{k}\right\} .
$$

From the neighbourhood conditions (3.6), we know that $\left\|\left(\tilde{b}_{k}, \tilde{c}_{k}\right)\right\|_{\mathcal{A}} \leq \gamma_{\mathcal{A}} \rho$, and from the definition of the semi-norm in (3.4), we have that: $\|(\hat{x}, \hat{z})\|_{2} \leq \gamma_{\mathcal{A}} \rho$. Using (3.4) alongside Assumption 2, we can also show that $\|\hat{y}\|_{2}=\boldsymbol{\Theta}\left(\|(\hat{x}, \hat{z})\|_{2}\right)$. On the other hand, from the definition of the starting point, we have that: $\left(x_{0}, z_{0}\right)=\rho\left(e_{n}, e_{n}\right)$. By defining the following auxiliary point:

$$
(\bar{x}, \bar{y}, \bar{z})=\left(x_{0}, y_{0}, z_{0}\right)+(\hat{x}, \hat{y}, \hat{z}),
$$

we have that $(\bar{x}, \bar{z}) \geq\left(1-\gamma_{\mathcal{A}}\right) \rho\left(e_{n}, e_{n}\right)$. By construction, the triple $(\bar{x}, \bar{y}, \bar{z})$ is a feasible solution for the primal-dual pair in (3.11)-(3.12), giving bounded primal and dual objective values, respectively.

Using our previous observations, alongside the fact that $\operatorname{rank}(A)=m$ (Assumption 2), we can confirm that there must exist a large constant $M>0$, and a triple $\left(x_{s}^{*}, y_{s}^{*}, z_{s}^{*}\right)$ solving (3.11)-(3.12), such that $\left\|\left(x_{s}^{*}, z_{s}^{*}\right)\right\|_{\infty} \leq M \Rightarrow$ $\left\|\left(x_{s}^{*}, y_{s}^{*}, z_{s}^{*}\right)\right\|_{2}=\boldsymbol{O}(\sqrt{n})$.

Let us now apply the proximal method of multipliers to (3.11)-(3.12), given the estimates $\zeta_{k}, \eta_{k}$. We should note at this point, that the proximal operator used here is different from that in (1.10), since it is based on a different maximal monotone operator from that in (1.8). In particular, we associate the following maximal monotone operator to (3.11)-(3.12):

$$
\begin{aligned}
\tilde{T}_{\mathcal{L}}(x, y):=\{(u, v): v & =Q x+\left(c+\bar{c}+\tilde{c}_{k}\right)-A^{\top} y-\theta X^{-1} e_{n}, \\
u & \left.=A x-\left(b+\bar{b}+\tilde{b}_{k}\right)\right\} .
\end{aligned}
$$

As before, the proximal operator is defined as $\tilde{P}:=\left(I_{m+n}+\tilde{T}_{\mathcal{L}}\right)^{-1}$, and is singlevalued and non-expansive. We let any $\mu \in(0, \infty)$ (noting that the case where $\mu=$ 0 follows directly by Assumption 1), and define the following penalty function:

$$
\begin{aligned}
\tilde{\mathcal{L}}_{\mu, \theta}\left(x ; \zeta_{k}, \eta_{k}\right):= & \left(c+\bar{c}+\tilde{c}_{k}\right)^{\top} x+\frac{1}{2} x^{\top} Q x+\frac{1}{2} \mu\left\|x-\zeta_{k}\right\|_{2}^{2}+ \\
& \frac{1}{2 \mu}\left\|A x-\left(b+\bar{b}+\tilde{b}_{k}\right)\right\|_{2}^{2}-\left(\eta_{k}\right)^{\top}\left(A x-\left(b+\bar{b}+\tilde{b}_{k}\right)\right)-\theta \sum_{j=1}^{n} \ln x^{j} .
\end{aligned}
$$

By defining the variables $y=\eta_{k}-\frac{1}{\mu}\left(A x-\left(b+\bar{b}+\tilde{b}_{k}\right)\right)$ and $z=\theta X^{-1} e_{n}$, we can see that the optimality conditions of this PMM sub-problem are exactly those stated in (3.10). Equivalently, we can find a pair $(\tilde{x}, \tilde{y})$ such that $(\tilde{x}, \tilde{y})=\tilde{P}\left(\zeta_{k}, \eta_{k}\right)$ and set $\tilde{z}=\theta \tilde{X}^{-1} e_{n}$. Then, we can use the non-expansiveness of $\tilde{P}$, as in Lemma 3.3.1, to obtain:

$$
\left\|(\tilde{x}, \tilde{y})-\left(x_{s}^{*}, y_{s}^{*}\right)\right\|_{2} \leq\left\|\left(\zeta_{k}, \eta_{k}\right)-\left(x_{s}^{*}, y_{s}^{*}\right)\right\|_{2} .
$$

But we know, from Lemma 3.3.2, that $\left\|\left(\zeta_{k}, \eta_{k}\right)\right\|_{2}=\boldsymbol{O}(\sqrt{n})$, for all $k \geq 0$. Combining this with our previous observations yields that $\|(\tilde{x}, \tilde{y})\|_{2}=\boldsymbol{O}(\sqrt{n})$. Setting $\tilde{z}=\theta \tilde{X}^{-1} e_{n}$, gives a triple ( $\left.\tilde{x}, \tilde{y}, \tilde{z}\right)$ that satisfies (3.10), while $\|(\tilde{x}, \tilde{y}, \tilde{z})\|_{2}=$ $\boldsymbol{O}(\sqrt{n})$. To conclude the proof, let us notice that the value of $\tilde{\mathcal{L}}_{\mu, \theta}\left(x ; \zeta_{k}, \eta_{k}\right)$ will grow unbounded as $x_{j} \rightarrow 0$ or $x_{j} \rightarrow \infty$, for any $j \in\{1, \ldots, n\}$. Hence, there must exist a constant $\tilde{M}>0$, such that the minimizer of this function must satisfy $\frac{1}{\tilde{M}} \leq \tilde{x}^{j} \leq \tilde{M}$, for all $j \in\{1, \ldots, n\}$. The relation $\tilde{X} \tilde{z}=\theta e_{n}$ then implies that $\frac{\theta}{\tilde{M}} \leq \tilde{z}^{j} \leq \theta \tilde{M}$. Hence $(\tilde{x}, \tilde{z}) \geq \xi\left(e_{n}, e_{n}\right)$, where $\xi>0$ and $\xi=\boldsymbol{\Theta}(1)$.

Lemma 3.3.4. Given Assumptions 1 and 2, the iterates ( $x_{k}, y_{k}, z_{k}$ ) produced by Algorithm IP-PMM-QP, for all $k \geq 0$, are such that:

$$
\left\|\left(x_{k}, z_{k}\right)\right\|_{1}=\boldsymbol{O}(n),\left\|\left(x_{k}, y_{k}, z_{k}\right)\right\|_{2}=\boldsymbol{O}(n) .
$$

Proof. Let an iterate $\left(x_{k}, y_{k}, z_{k}\right) \in \mathcal{N}_{\mu_{k}}\left(\zeta_{k}, \eta_{k}\right)$, produced by Algorithm IP-PMMQP during an arbitrary iteration $k \geq 0$, be given. Firstly, we invoke Lemma 3.3.3, from which we have a triple ( $\tilde{x}, \tilde{y}, \tilde{z}$ ) satisfying (3.10), for $\mu=\mu_{k}$. Similarly, by invoking Lemma 3.3.2, we know that there exists a triple $\left(x_{r_{k}}^{*}, y_{r_{k}}^{*}, z_{r_{k}}^{*}\right)$ satisfying (3.9), with $\mu=\mu_{k}$. Consider the following auxiliary point:

$$
\begin{equation*}
\left(\left(1-\frac{\mu_{k}}{\mu_{0}}\right) x_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{x}-x_{k},\left(1-\frac{\mu_{k}}{\mu_{0}}\right) y_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{y}-y_{k},\left(1-\frac{\mu_{k}}{\mu_{0}}\right) z_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{z}-z_{k}\right) \tag{3.13}
\end{equation*}
$$

Using (3.13) and (3.9)-(3.10) (for $\mu=\mu_{k}$ ), one can observe that:

$$
\begin{aligned}
A\left(\left(1-\frac{\mu_{k}}{\mu_{0}}\right) x_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{x}-x_{k}\right)+\mu_{k}\left(\left(1-\frac{\mu_{k}}{\mu_{0}}\right) y_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{y}-y_{k}\right) & = \\
\left(1-\frac{\mu_{k}}{\mu_{0}}\right)\left(A x_{r_{k}}^{*}+\mu_{k} y_{r_{k}}^{*}\right)+\frac{\mu_{k}}{\mu_{0}}\left(A \tilde{x}+\mu_{k} \tilde{y}\right)-A x_{k}-\mu_{k} y_{k} & = \\
\left(1-\frac{\mu_{k}}{\mu_{0}}\right)\left(b+\mu_{k} \eta_{k}\right)+\frac{\mu_{k}}{\mu_{0}}\left(b+\mu_{k} \eta_{k}+\tilde{b}+\bar{b}\right)-A x_{k}-\mu_{k} y_{k} & = \\
b+\mu_{k} \eta_{k}+\frac{\mu_{k}}{\mu_{0}}(\tilde{b}+\bar{b})-A x_{k}-\mu_{k} y_{k} & =0_{m},
\end{aligned}
$$

where the last equality follows from the definition of the neighbourhood $\mathcal{N}_{\mu_{k}}\left(\zeta_{k}, \eta_{k}\right)$. Similarly:

$$
\begin{aligned}
& -\left(Q+\mu_{k} I_{n}\right)\left(\left(1-\frac{\mu_{k}}{\mu_{0}}\right) x_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{x}-x_{k}\right) \\
& \quad+A^{\top}\left(\left(1-\frac{\mu_{k}}{\mu_{0}}\right) y_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{y}-y_{k}\right)+\left(\left(1-\frac{\mu_{k}}{\mu_{0}}\right) z_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{z}-z_{k}\right)=0_{n}
\end{aligned}
$$

By combining the previous two relations, we have:

$$
\begin{array}{r}
\left(\left(1-\frac{\mu_{k}}{\mu_{0}}\right) x_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{x}-x_{k}\right)^{\top}\left(\left(1-\frac{\mu_{k}}{\mu_{0}}\right) z_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{z}-z_{k}\right)= \\
\left(\left(1-\frac{\mu_{k}}{\mu_{0}}\right) x_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{x}-x_{k}\right)^{\top}\left(Q+\mu_{k} I\right)\left(\left(1-\frac{\mu_{k}}{\mu_{0}}\right) x_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{x}-x_{k}\right)  \tag{3.14}\\
+\mu_{k}\left(\left(1-\frac{\mu_{k}}{\mu_{0}}\right) y_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{y}-y_{k}\right)^{\top}\left(\left(1-\frac{\mu_{k}}{\mu_{0}}\right) y_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{y}-y_{k}\right) \geq 0 .
\end{array}
$$

From (3.14), it can be seen that:

$$
\begin{aligned}
((1- & \left.\left.-\frac{\mu_{k}}{\mu_{0}}\right) x_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{x}\right)^{\top} z_{k}+\left(\left(1-\frac{\mu_{k}}{\mu_{0}}\right) z_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{z}\right)^{\top} x_{k} \\
& \leq\left(\left(1-\frac{\mu_{k}}{\mu_{0}}\right) x_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{x}\right)^{\top}\left(\left(1-\frac{\mu_{k}}{\mu_{0}}\right) z_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{z}\right)+x_{k}^{\top} z_{k} .
\end{aligned}
$$

However, from Lemmas 3.3.2 and 3.3.3, we have that: $(\tilde{x}, \tilde{z}) \geq \xi\left(e_{n}, e_{n}\right)$, for some positive constant $\xi=\boldsymbol{\Theta}(1)$, while $\left\|\left(x_{r_{k}}^{*}, z_{r_{k}}^{*}\right)\right\|_{2}=\boldsymbol{O}(\sqrt{n})$, and $\|(\tilde{x}, \tilde{z})\|_{2}=\boldsymbol{O}(\sqrt{n})$. Furthermore, by definition we have that $n \mu_{k}=x_{k}^{\top} z_{k}$. By combining all the
previous, we obtain:

$$
\begin{align*}
& \frac{\mu_{k}}{\mu_{0}} \xi\left(e^{\top} x_{k}+e^{\top} z_{k}\right) \leq \\
&\left(\left(1-\frac{\mu_{k}}{\mu_{0}}\right) x_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{x}\right)^{\top} z_{k}+\left(\left(1-\frac{\mu_{k}}{\mu_{0}}\right) z_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{z}\right)^{\top} x_{k} \leq  \tag{3.15}\\
&\left(\left(1-\frac{\mu_{k}}{\mu_{0}}\right) x_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{x}\right)^{\top}\left(\left(1-\frac{\mu_{k}}{\mu_{0}}\right) z_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{z}\right)+x_{k}^{\top} z_{k}= \\
& \frac{\mu_{k}}{\mu_{0}}\left(1-\frac{\mu_{k}}{\mu_{0}}\right)\left(x_{r_{k}}^{*}\right)^{\top} \tilde{z}+\frac{\mu_{k}}{\mu_{0}}\left(1-\frac{\mu_{k}}{\mu_{0}}\right) \tilde{x}^{\top} z_{r_{k}}^{*}+\left(\frac{\mu_{k}}{\mu_{0}}\right)^{2} \tilde{x}^{\top} \tilde{z}+x_{k}^{\top} z_{k}=\boldsymbol{O}\left(\mu_{k} n\right),
\end{align*}
$$

where we used (3.9) $\left(\left(x_{r_{k}}^{*}\right)^{\top}\left(z_{r_{k}}^{*}\right)=0\right)$. Hence, (3.15) implies that:

$$
\left\|\left(x_{k}, z_{k}\right)\right\|_{1}=\boldsymbol{O}(n) .
$$

From equivalence of norms, we have that $\left\|\left(x_{k}, z_{k}\right)\right\|_{2} \leq\left\|\left(x_{k}, z_{k}\right)\right\|_{1}$. Finally, from the neighbourhood conditions we know that:

$$
c+Q x_{k}-A^{\top} y_{k}-z_{k}+\mu_{k}\left(x_{k}-\zeta_{k}\right)+\frac{\mu_{k}}{\mu_{0}}\left(\tilde{c}_{k}+\bar{c}\right)=0 .
$$

All terms above (except for $y_{k}$ ) have a 2-norm that is $\boldsymbol{O}(n)$ (note that $\|(\bar{c}, \bar{b})\|_{2}=$ $\boldsymbol{O}(\sqrt{n})$ using Assumption 2 and the definition in (3.5)). Hence, using again Assumption 2 yields that $\left\|y_{k}\right\|_{2}=\boldsymbol{O}(n)$, and completes the proof.

As in a typical IPM convergence analysis, we proceed by bounding some components of the scaled Newton direction. The proof of that uses similar arguments to those presented in [181, Lemma 6.5]. Combining this result with Assumption 2, allows us to bound also the non-scaled Newton direction.

Lemma 3.3.5. Given Assumptions 1 and 2, and the Newton direction (denoted as ( $\left.\Delta x_{k}, \Delta y_{k}, \Delta z_{k}\right)$ ) obtained by solving system (3.7) during an arbitrary iteration $k \geq 0$ of Algorithm IP-PMM-QP, we have that:

$$
\left\|D_{k}^{-1} \Delta x_{k}\right\|_{2}=\boldsymbol{O}\left(n^{2} \mu^{\frac{1}{2}}\right),\left\|D_{k} \Delta z_{k}\right\|_{2}=\boldsymbol{O}\left(n^{2} \mu^{\frac{1}{2}}\right),\left\|\left(\Delta x_{k}, \Delta y_{k}, \Delta z_{k}\right)\right\|_{2}=\boldsymbol{O}\left(n^{3}\right)
$$

with $D_{k}^{2}:=X_{k} Z_{k}^{-1}$.
Proof. Consider an arbitrary iteration $k$ of Algorithm IP-PMM-QP. We invoke Lemmas 3.3.2, 3.3.3, for $\mu=\tau_{k} \mu_{k}$. That is, there exists a triple ( $x_{r_{k}}^{*}, y_{r_{k}}^{*}, z_{r_{k}}^{*}$ ) satisfying (3.9), and a triple ( $\tilde{x}, \tilde{y}, \tilde{z}$ ) satisfying (3.10), for $\mu=\tau_{k} \mu_{k}$. Using the centering parameter $\tau_{k}$, we define the following vectors:

$$
\begin{align*}
& \hat{c}:=-\left(\tau_{k} \frac{\bar{c}}{\mu_{0}}-\left(1-\tau_{k}\right)\left(x_{k}-\zeta_{k}+\frac{\mu_{k}}{\mu_{0}}\left(\tilde{x}-x_{r_{k}}^{*}\right)\right)\right), \\
& \hat{b}:=-\left(\tau_{k} \frac{\bar{b}}{\mu_{0}}+\left(1-\tau_{k}\right)\left(y_{k}-\eta_{k}+\frac{\mu_{k}}{\mu_{0}}\left(\tilde{y}-y_{r_{k}}^{*}\right)\right)\right), \tag{3.16}
\end{align*}
$$

where $\bar{b}, \bar{c}$, $\mu_{0}$ are defined in (3.5). Using Lemmas 3.3.2, 3.3.3, 3.3.4, and Assumption 2, we know that $\|(\hat{c}, \hat{b})\|_{2}=\boldsymbol{O}(n)$. Then, by applying again Assumption

2, we know that there must exist a vector $\hat{x}$ such that: $A \hat{x}=\hat{b},\|\hat{x}\|_{2}=\boldsymbol{O}(n)$, and by setting $\hat{z}=\hat{c}+Q \hat{x}+\mu \hat{x}$, we have that $\|\hat{z}\|_{2}=\boldsymbol{O}(n)$ and:

$$
\begin{equation*}
A \hat{x}=\hat{b},-Q \hat{x}+\hat{z}-\mu_{k} \hat{x}=\hat{c} \tag{3.17}
\end{equation*}
$$

Using $\left(x_{r_{k}}^{*}, y_{r_{k}}^{*}, z_{r_{k}}^{*}\right),(\tilde{x}, \tilde{y}, \tilde{z})$, as well as the triple $\left(\hat{x}, 0_{m}, \hat{z}\right)$, where $(\hat{x}, \hat{z})$ is defined in (3.17), we can define the following auxiliary triple:

$$
\begin{equation*}
(\bar{x}, \bar{y}, \bar{z})=\left(\Delta x_{k}, \Delta y_{k}, \Delta z_{k}\right)+\frac{\mu_{k}}{\mu_{0}}(\tilde{x}, \tilde{y}, \tilde{z})-\frac{\mu_{k}}{\mu_{0}}\left(x_{r_{k}}^{*}, y_{r_{k}}^{*}, z_{r_{k}}^{*}\right)+\mu_{k}\left(\hat{x}, 0_{m}, \hat{z}\right) \tag{3.18}
\end{equation*}
$$

Using (3.18), (3.16), (3.9)-(3.10) (with $\mu=\tau_{k} \mu_{k}$ ), and the second block equation of (3.7), we have:

$$
\begin{aligned}
A \bar{x}+\mu_{k} \bar{y}= & \left(A \Delta x_{k}+\mu_{k} \Delta y_{k}\right)+\frac{\mu_{k}}{\mu_{0}}\left(\left(A \tilde{x}+\mu_{k} \tilde{y}\right)-\left(A x_{r_{k}}^{*}+\mu_{k} y_{r_{k}}^{*}\right)\right)+\mu_{k} A \hat{x} \\
= & \left(b+\tau_{k} \frac{\mu_{k}}{\mu_{0}} \bar{b}-A x_{k}-\tau_{k} \mu_{k}\left(y_{k}-\eta_{k}\right)\right) \\
& +\frac{\mu_{k}}{\mu_{0}}\left(\left(A \tilde{x}+\mu_{k} \tilde{y}\right)-\left(A x_{r_{k}}^{*}+\mu_{k} y_{r_{k}}^{*}\right)\right) \\
& -\mu_{k}\left(\tau_{k} \frac{\bar{b}}{\mu_{0}}+\left(1-\tau_{k}\right)\left(y_{k}-\eta_{k}\right)\right)-\frac{\mu_{k}}{\mu_{0}}\left(1-\tau_{k}\right) \mu_{k}\left(\tilde{y}-y_{r_{k}}^{*}\right) \\
= & \left(b+\tau_{k} \frac{\mu_{k}}{\mu_{0}} \bar{b}-A x_{k}-\tau_{k} \mu_{k}\left(y_{k}-\eta_{k}\right)\right)+\frac{\mu_{k}}{\mu_{0}}\left(b+\tau_{k} \mu_{k} \eta_{k}+\bar{b}+\tilde{b}_{k}\right) \\
& -\frac{\mu_{k}}{\mu_{0}}\left(\tau_{k} \mu_{k} \eta_{k}+b\right)-\mu_{k}\left(\tau_{k} \frac{\bar{b}}{\mu_{0}}+\left(1-\tau_{k}\right)\left(y_{k}-\eta_{k}\right)\right) \\
= & b+\frac{\mu_{k}}{\mu_{0}}\left(\bar{b}+\tilde{b}_{k}\right)-A x_{k}-\mu_{k}\left(y_{k}-\eta_{k}\right) \\
= & 0_{m},
\end{aligned}
$$

where the last equation follows from the neighbourhood conditions (i.e. $\left(x_{k}, y_{k}, z_{k}\right) \in$ $\left.\mathcal{N}_{\mu_{k}}\left(\zeta_{k}, \eta_{k}\right)\right)$. Similarly, we can show that:

$$
-Q \bar{x}+A^{\top} \bar{y}+\bar{z}-\mu_{k} \bar{x}=0_{n}
$$

The previous two equalities imply that:

$$
\begin{equation*}
\bar{x}^{\top} \bar{z}=\bar{x}^{\top}\left(Q \bar{x}-A^{\top} \bar{y}+\mu_{k} \bar{x}\right)=\bar{x}^{\top}\left(Q+\mu_{k} I\right) \bar{x}+\mu_{k} \bar{y}^{\top} \bar{y} \geq 0 \tag{3.19}
\end{equation*}
$$

On the other hand, using the last block equation of the Newton system (3.7), we have:

$$
\begin{aligned}
Z_{k} \bar{x}+X_{k} \bar{z}= & -X_{k} Z_{k} e_{n}+\tau_{k} \mu_{k} e_{n} \\
& +\frac{\mu_{k}}{\mu_{0}} Z_{k}\left(\tilde{x}-x_{r_{k}}^{*}\right)+\frac{\mu_{k}}{\mu_{0}} X_{k}\left(\tilde{z}-z_{r_{k}}^{*}\right)+\mu_{k} Z_{k} \hat{x}+\mu_{k} X_{k} \hat{z} .
\end{aligned}
$$

Let $W_{k}=\left(X_{k} Z_{k}\right)^{\frac{1}{2}}$. By multiplying both sides of the previous equation by $W_{k}^{-1}$,
we obtain:

$$
\begin{align*}
D_{k}^{-1} \bar{x}+D_{k} \bar{z}= & -W_{k}^{-1}\left(X_{k} Z_{k} e_{n}-\tau_{k} \mu_{k} e_{n}\right) \\
& +\frac{\mu_{k}}{\mu_{0}}\left(D_{k}^{-1}\left(\tilde{x}-x_{r_{k}}^{*}\right)+D_{k}\left(\tilde{z}-z_{r_{k}}^{*}\right)\right)+\mu_{k}\left(D_{k}^{-1} \hat{x}+D_{k} \hat{z}\right) . \tag{3.20}
\end{align*}
$$

But, from (3.19), we know that $\bar{x}^{\top} \bar{z} \geq 0$, and hence:

$$
\left\|D_{k}^{-1} \bar{x}+D_{k} \bar{z}\right\|_{2}^{2} \geq\left\|D_{k}^{-1} \bar{x}\right\|_{2}^{2}+\left\|D_{k} \bar{z}\right\|_{2}^{2}
$$

Combining (3.20) with the previous inequality, gives:

$$
\begin{aligned}
\left\|D_{k}^{-1} \bar{x}\right\|_{2}^{2}+\left\|D_{k} \bar{z}\right\|_{2}^{2} \leq & \left\{\left\|W_{k}^{-1}\right\|_{2}\left\|X_{k} Z_{k} e_{n}-\tau_{k} \mu_{k} e_{n}\right\|_{2}+\right. \\
& \frac{\mu_{k}}{\mu_{0}}\left(\left\|D_{k}^{-1}\left(\tilde{x}-x_{r_{k}}^{*}\right)\right\|_{2}+\left\|D_{k}\left(\tilde{z}-z_{r_{k}}^{*}\right)\right\|_{2}\right)+ \\
& \left.\mu_{k}\left(\left\|D_{k}^{-1} \hat{x}\right\|_{2}+\left\|D_{k} \hat{z}\right\|_{2}\right)\right\}^{2} .
\end{aligned}
$$

We isolate one of the two terms of the left hand side of the previous inequality, take square roots on both sides, use (3.18) and apply the triangle inequality to it, to obtain:

$$
\begin{align*}
\left\|D_{k}^{-1} \Delta x_{k}\right\|_{2} \leq & \left\|W_{k}^{-1}\right\|_{2}\left\|X_{k} Z_{k} e_{n}-\tau_{k} \mu_{k} e_{n}\right\|_{2}+ \\
& \frac{\mu_{k}}{\mu_{0}}\left(2\left\|D_{k}^{-1}\left(\tilde{x}-x_{r_{k}}^{*}\right)\right\|_{2}+\left\|D_{k}\left(\tilde{z}-z_{r_{k}}^{*}\right)\right\|_{2}\right)+  \tag{3.21}\\
& \mu_{k}\left(2\left\|D_{k}^{-1} \hat{x}\right\|_{2}+\left\|D_{k} \hat{z}\right\|_{2}\right) .
\end{align*}
$$

We now proceed to bounding the terms in the right hand side of (3.21). Firstly, notice from the neighbourhood conditions (3.6) that $\gamma_{\mu} \mu_{k} \leq x_{k}^{i} z_{k}^{i}$. This in turn implies that:

$$
\left\|W_{k}^{-1}\right\|_{2}=\max _{i} \frac{1}{\left(x_{k}^{i} z_{k}^{i}\right)^{\frac{1}{2}}} \leq \frac{1}{\left(\gamma_{\mu} \mu_{k}\right)^{\frac{1}{2}}} .
$$

On the other hand, we have that:

$$
\begin{aligned}
\left\|X_{k} Z_{k} e_{n}-\tau_{k} \mu_{k} e_{n}\right\|_{2}^{2} & =\left\|X_{k} Z_{k} e_{n}\right\|^{2}-2 \tau_{k} \mu_{k} x_{k}^{\top} z_{k}+\tau_{k}^{2} \mu_{k}^{2} n \\
& \leq\left\|X_{k} Z_{k} e_{n}\right\|_{1}^{2}-2 \tau_{k} \mu_{k} x_{k}^{\top} z_{k}+\tau_{k}^{2} \mu_{k}^{2} n \\
& =\left(\mu_{k} n\right)^{2}-2 \tau_{k} \mu_{k}^{2} n+\tau_{k}^{2} \mu_{k}^{2} n \\
& \leq \mu_{k}^{2} n^{2} .
\end{aligned}
$$

Hence, combining the previous two relations yields:

$$
\left\|W_{k}^{-1}\right\|_{2}\left\|X_{k} Z_{k} e_{n}-\tau_{k} \mu_{k} e_{n}\right\|_{2} \leq \frac{n}{\gamma_{\mu}^{\frac{1}{2}}} \mu_{k}^{\frac{1}{2}}=\boldsymbol{O}\left(n \mu^{\frac{1}{2}}\right)
$$

We proceed by bounding $\left\|D_{k}^{-1}\right\|_{2}$. To that end, using Lemma 3.3.4, we have:

$$
\left\|D_{k}^{-1}\right\|_{2}=\max _{i}\left|\left(D_{k}^{i i}\right)^{-1}\right|=\left\|D_{k}^{-1} e_{n}\right\|_{\infty}=\left\|W_{k}^{-1} Z_{k} e_{n}\right\|_{\infty} \leq\left\|W_{k}^{-1}\right\|_{2}\left\|z_{k}\right\|_{1}=\boldsymbol{O}\left(\frac{n}{\mu_{k}^{\frac{1}{2}}}\right) .
$$

Similarly, we have that:

$$
\left\|D_{k}\right\|_{2}=\boldsymbol{O}\left(\frac{n}{\mu_{k}^{\frac{1}{2}}}\right)
$$

Hence, using the previous bounds, as well as Lemmas 3.3.2, 3.3.3, we obtain:

$$
\begin{aligned}
& 2 \frac{\mu_{k}}{\mu_{0}}\left\|D_{k}^{-1}\left(\tilde{x}-x_{r_{k}}^{*}\right)\right\|_{2}+\frac{\mu_{k}}{\mu_{0}}\left\|D_{k}\left(\tilde{z}-z_{r_{k}}^{*}\right)\right\|_{2} \\
& \quad \leq 2 \frac{\mu_{k}}{\mu_{0}}\left(\left\|D_{k}^{-1}\right\|_{2}+\left\|D_{k}\right\|_{2}\right) \max \left\{\left\|\tilde{x}-x_{r_{k}}^{*}\right\|_{2},\left\|\tilde{z}-z_{r_{k}}^{*}\right\|_{2}\right\}=\boldsymbol{O}\left(n^{\frac{3}{2}} \mu_{k}^{\frac{1}{2}}\right)
\end{aligned}
$$

and

$$
\mu_{k}\left(2\left\|D_{k}^{-1} \hat{x}\right\|_{2}+\left\|D_{k} \hat{z}\right\|_{2}\right) \leq 2 \mu_{k}\left(\left\|D_{k}^{-1}\right\|_{2}+\left\|D_{k}\right\|_{2}\right) \max \left\{\|\hat{x}\|_{2},\|\hat{z}\|_{2}\right\}=\boldsymbol{O}\left(n^{2} \mu_{k}^{\frac{1}{2}}\right)
$$

Combining all the previous bounds yields the claimed bound for $\left\|D_{k}^{-1} \Delta x_{k}\right\|_{2}$. One can bound $\left\|D_{k} \Delta z_{k}\right\|_{2}$ in the same way. The latter is omitted for ease of presentation.

Finally, we have that:

$$
\left\|\Delta x_{k}\right\|_{2}=\left\|D_{k} D_{k}^{-1} \Delta x_{k}\right\|_{2} \leq\left\|D_{k}\right\|_{2}\left\|D_{k}^{-1} \Delta x_{k}\right\|_{2}=\boldsymbol{O}\left(n^{3}\right) .
$$

Similarly, we can show that $\left\|\Delta z_{k}\right\|_{2}=\boldsymbol{O}\left(n^{3}\right)$. From the first block equation of the Newton system in (3.7), alongside Assumption 2, we can show that $\left\|\Delta y_{k}\right\|_{2}=$ $\boldsymbol{O}\left(n^{3}\right)$, which completes the proof.
We are now able to prove that at every iteration of Algorithm IP-PMM-QP, there exists a step-length $\alpha_{k}>0$, using which, the new iterate satisfies the conditions required by the algorithm. The lower bound on any such step-length will later determine the polynomial complexity of the method.
Lemma 3.3.6. Given Assumptions 1 and 2 , there exists a step-length $\bar{\alpha} \in(0,1)$, such that for all $\alpha \in[0, \bar{\alpha}]$, for all $i \in\{1, \ldots, n\}$, and for all iterations $k \geq 0$ of Algorithm IP-PMM-QP, the following relations hold:

$$
\begin{array}{r}
\left(x_{k}+\alpha \Delta x_{k}\right)^{\top}\left(z_{k}+\alpha \Delta z_{k}\right) \geq\left(1-\alpha\left(1-\beta_{1}\right)\right) x_{k}^{\top} z_{k}, \\
\left(x_{k}^{i}+\alpha \Delta x_{k}^{i}\right)\left(z_{k}^{i}+\alpha \Delta z_{k}^{i}\right) \geq \frac{\gamma_{\mu}}{n}\left(x_{k}+\alpha \Delta x_{k}\right)^{\top}\left(z_{k}+\alpha \Delta z_{k}\right), \\
\left(x_{k}+\alpha \Delta x_{k}\right)^{\top}\left(z_{k}+\alpha \Delta z_{k}\right) \leq\left(1-\alpha\left(1-\beta_{2}\right)\right) x_{k}^{\top} z_{k}, \tag{3.24}
\end{array}
$$

where, without loss of generality, $\beta_{1}=\frac{\tau_{\min }}{2}$ and $\beta_{2}=0.99$. Moreover, $\bar{\alpha} \geq \frac{\bar{\varepsilon}}{n^{4}}$ for all $k \geq 0$, where $\bar{\kappa}>0$ is independent of $n$ and $m$. If $\left(x_{k}, y_{k}, z_{k}\right) \in \mathcal{N}_{\mu_{k}}\left(\zeta_{k}, \eta_{k}\right)$, then letting:

$$
\left(x_{k+1}, y_{k+1}, z_{k+1}\right)=\left(x_{k}+\alpha \Delta x_{k}, y_{k}+\alpha \Delta y_{k}, z_{k}+\alpha \Delta z_{k}\right), \quad \text { for all } \alpha \in(0, \bar{\alpha}],
$$

and $\mu_{k+1}=\left(x_{k+1}^{\top} z_{k+1}\right) / n$, gives $\left(x_{k+1}, y_{k+1}, z_{k+1}\right) \in \mathcal{N}_{\mu_{k+1}}\left(\zeta_{k+1}, \eta_{k+1}\right)$, where $\eta_{k}, \zeta_{k}$ are updated as in Algorithm IP-PMM-QP.

Proof. In order to prove the first three inequalities, we follow the developments in [181, Lemma 6.7]. From Lemma 3.3.5, we have that there exists a constant $K_{\Delta}>0$, such that:

$$
\left(\Delta x_{k}\right)^{\top} \Delta z_{k}=\left(D_{k}^{-1} \Delta x_{k}\right)^{\top}\left(D_{k} \Delta z_{k}\right) \leq\left\|D_{k}^{-1} \Delta x_{k}\right\|_{2}\left\|D_{k} \Delta z_{k}\right\|_{2} \leq K_{\Delta}^{2} n^{4} \mu_{k} .
$$

Similarly, it is easy to see that:

$$
\left|\Delta x_{k}^{i} \Delta z_{k}^{i}\right| \leq K_{\Delta}^{2} n^{4} \mu_{k} .
$$

On the other hand, by summing over all $n$ components of the last block equation of the Newton system (3.7), we have:

$$
\begin{equation*}
z_{k}^{\top} \Delta x_{k}+x_{k}^{\top} \Delta z_{k}=e_{n}^{\top}\left(Z_{k} \Delta x_{k}+X_{k} \Delta z_{k}\right)=e_{n}^{\top}\left(-X_{k} Z_{k} e_{n}+\tau_{k} \mu_{k} e_{n}\right)=\left(\tau_{k}-1\right) x_{k}^{\top} z_{k}, \tag{3.25}
\end{equation*}
$$

while the components of the last block equation of the Newton system (3.7) can be written as:

$$
\begin{equation*}
z_{k}^{i} \Delta x_{k}^{i}+x_{k}^{i} \Delta z_{k}^{i}=-x_{k}^{i} z_{k}^{i}+\tau_{k} \mu_{k} . \tag{3.26}
\end{equation*}
$$

We proceed by proving (3.22). Using (3.25), we have:

$$
\begin{aligned}
\left(x_{k}+\right. & \left.\alpha \Delta x_{k}\right)^{\top}\left(z_{k}+\alpha \Delta z_{k}\right)-\left(1-\alpha\left(1-\beta_{1}\right)\right) x_{k}^{\top} z_{k} \\
& =x_{k}^{\top} z_{k}+\alpha\left(\tau_{k}-1\right) x_{k}^{\top} z_{k}+\alpha^{2} \Delta x_{k}^{\top} \Delta z_{k}-(1-\alpha) x_{k}^{\top} z_{k}-\alpha \beta_{1} x_{k}^{\top} z_{k} \\
& \geq \alpha\left(\tau_{k}-\beta_{1}\right) x_{k}^{\top} z_{k}-\alpha^{2} K_{\Delta}^{2} n^{4} \mu_{k} \geq \alpha\left(\frac{\tau_{\min }}{2}\right) n \mu_{k}-\alpha^{2} K_{\Delta}^{2} n^{4} \mu_{k}
\end{aligned}
$$

where we set (without loss of generality) $\beta_{1}=\frac{\tau_{\min }}{2}$. The rightmost term of the previous inequality will be non-negative for every $\alpha$ satisfying:

$$
\alpha \leq \frac{\tau_{\min }}{2 K_{\Delta}^{2} n^{3}} .
$$

In order to prove (3.23), we will use (3.26) and the fact that from the neighbourhood conditions we have that $x_{k}^{i} z_{k}^{i} \geq \gamma_{\mu} \mu_{k}$. In particular, we obtain:

$$
\begin{aligned}
\left(x_{k}^{i}+\alpha \Delta x_{k}^{i}\right)\left(z_{k}^{i}+\alpha \Delta z_{k}^{i}\right) & \geq(1-\alpha) x_{k}^{i} z_{k}^{i}+\alpha \tau_{k} \mu_{k}-\alpha^{2} K_{\Delta}^{2} n^{4} \mu_{k} \\
& \geq \gamma_{\mu}(1-\alpha) \mu_{k}+\alpha \tau_{k} \mu_{k}-\alpha^{2} K_{\Delta}^{2} n^{4} \mu_{k} .
\end{aligned}
$$

By combining all the previous, we get:

$$
\begin{aligned}
\left(x_{k}^{i}+\right. & \left.\alpha \Delta x_{k}^{i}\right)\left(z_{k}^{i}+\alpha \Delta z_{k}^{i}\right)-\frac{\gamma_{\mu}}{n}\left(x_{k}+\alpha \Delta x_{k}\right)^{\top}\left(z_{k}+\alpha \Delta z_{k}\right) \\
& \geq \alpha \tau_{k}\left(1-\gamma_{\mu}\right) \mu_{k}-\left(1+\frac{\gamma_{\mu}}{n}\right) \alpha^{2} K_{\Delta}^{2} n^{4} \mu_{k} \\
& \geq \alpha \tau_{\min }\left(1-\gamma_{\mu}\right) \mu_{k}-2 \alpha^{2} K_{\Delta}^{2} n^{4} \mu_{k} .
\end{aligned}
$$

In turn, the last term of the previous relation will be non-negative for every $\alpha$
satisfying:

$$
\alpha \leq \frac{\tau_{\min }\left(1-\gamma_{\mu}\right)}{2 K_{\Delta}^{2} n^{4}}
$$

Finally, to prove (3.24), we set (without loss of generality) $\beta_{2}=0.99$. We know, from Algorithm IP-PMM-QP, that $\tau_{\max } \leq 0.5$. With the previous two remarks in mind, we have:

$$
\begin{aligned}
\frac{1}{n}\left(x_{k}\right. & \left.+\alpha \Delta x_{k}\right)^{\top}\left(z_{k}+\alpha \Delta z_{k}\right)-(1-0.01 \alpha) \mu_{k} \\
& \leq(1-\alpha) \mu_{k}+\alpha \tau_{k} \mu_{k}+\alpha^{2} \frac{K_{\Delta}^{2} n^{4}}{n} \mu_{k}-(1-0.01 \alpha) \mu_{k} \\
& \leq-0.99 \alpha \mu_{k}+0.5 \alpha \mu_{k}+\alpha^{2} \frac{K_{\Delta}^{2} n^{4}}{n} \mu_{k} \\
& =-0.49 \alpha \mu_{k}+\alpha^{2} \frac{K_{\Delta}^{2} n^{4}}{n} \mu_{k}
\end{aligned}
$$

The last term will be non-positive for every $\alpha$ satisfying:

$$
\alpha \leq \frac{0.49}{K_{\Delta}^{2} n^{3}} .
$$

By combining all the previous bounds on the step-length, we have that (3.22)(3.24) will hold for every $\alpha \in\left(0, \alpha^{*}\right]$, where:

$$
\begin{equation*}
\alpha^{*}:=\min \left\{\frac{\tau_{\min }}{2 K_{\Delta}^{2} n^{3}}, \frac{\tau_{\min }\left(1-\gamma_{\mu}\right)}{2 K_{\Delta}^{2} n^{4}}, \frac{0.49}{K_{\Delta}^{2} n^{3}}, 1\right\} . \tag{3.27}
\end{equation*}
$$

Next, we would like to find the maximum $\bar{\alpha} \in\left(0, \alpha^{*}\right]$, such that:

$$
\left(x_{k}(\alpha), y_{k}(\alpha), z_{k}(\alpha)\right) \in \mathcal{N}_{\mu_{k}(\alpha)}\left(\zeta_{k}, \eta_{k}\right), \text { for all } \alpha \in(0, \bar{\alpha}]
$$

where $\mu_{k}(\alpha)=\frac{x_{k}(\alpha)^{\top} z_{k}(\alpha)}{n}$ and:

$$
\left(x_{k}(\alpha), y_{k}(\alpha), z_{k}(\alpha)\right)=\left(x_{k}+\alpha \Delta x_{k}, y_{k}+\alpha \Delta y_{k}, z_{k}+\alpha \Delta z_{k}\right) .
$$

Let:

$$
\tilde{r}_{p}(\alpha)=A x_{k}(\alpha)+\mu_{k}(\alpha)\left(y_{k}(\alpha)-\eta_{k}\right)-\left(b+\frac{\mu_{k}(\alpha)}{\mu_{0}} \bar{b}\right)
$$

and

$$
\tilde{r}_{d}(\alpha)=-Q x_{k}(\alpha)+A^{\top} y_{k}(\alpha)+z_{k}(\alpha)-\mu_{k}(\alpha)\left(x_{k}(\alpha)-\zeta_{k}\right)-\left(c+\frac{\mu_{k}(\alpha)}{\mu_{0}} \bar{c}\right) .
$$

In other words, we need to find the maximum $\bar{\alpha} \in\left(0, \alpha^{*}\right]$, such that:

$$
\begin{equation*}
\left\|\tilde{r}_{p}(\alpha), \tilde{r}_{d}(\alpha)\right\|_{2} \leq K_{N} \frac{\mu_{k}(\alpha)}{\mu_{0}},\left\|\tilde{r}_{p}(\alpha), \tilde{r}_{d}(\alpha)\right\|_{\mathcal{A}} \leq \gamma_{\mathcal{A}} \rho \frac{\mu_{k}(\alpha)}{\mu_{0}}, \text { for all } \alpha \in(0, \bar{\alpha}] \tag{3.28}
\end{equation*}
$$

If the latter two conditions hold, then $\left(x_{k}(\alpha), y_{k}(\alpha), z_{k}(\alpha)\right) \in \mathcal{N}_{\mu_{k}(\alpha)}\left(\zeta_{k}, \eta_{k}\right)$, for all
$\alpha \in(0, \bar{\alpha}]$. Then, if Algorithm IP-PMM-QP updates $\zeta_{k}, \eta_{k}$, it does so only when similar conditions (as in (3.28)) hold for the new parameters. If the parameters are not updated, then the new iterate lies in the desired neighbourhood because of (3.28), alongside (3.22)-(3.24).

We start by rearranging $\tilde{r}_{p}(\alpha)$. Specifically, we have that:

$$
\begin{aligned}
\tilde{r}_{p}(\alpha)= & A\left(x_{k}+\alpha \Delta x_{k}\right)-b \\
& +\left(\mu_{k}+\alpha\left(\tau_{k}-1\right) \mu_{k}+\alpha^{2} \frac{\Delta x_{k}^{\top} \Delta z_{k}}{n}\right)\left(\left(y_{k}+\alpha \Delta y_{k}-\eta_{k}\right)-\frac{\bar{b}}{\mu_{0}}\right) \\
= & \left(A x_{k}+\mu_{k}\left(y_{k}-\eta_{k}\right)-b-\frac{\mu_{k}}{\mu_{0}} \bar{b}\right)+\alpha\left(A \Delta x_{k}+\mu_{k} \Delta y_{k}\right)+ \\
& +\left(\alpha\left(\tau_{k}-1\right) \mu_{k}+\alpha^{2} \frac{\Delta x_{k}^{\top} \Delta z_{k}}{n}\right)\left(\left(y_{k}-\eta_{k}+\alpha \Delta y_{k}\right)-\frac{\bar{b}}{\mu_{0}}\right) \\
= & \frac{\mu_{k}}{\mu_{0}} \tilde{b}_{k}+\alpha\left(b-A x_{k}-\tau_{k} \mu_{k}\left(\left(y_{k}-\eta_{k}\right)-\frac{\bar{b}}{\mu_{0}}\right)+\mu_{k}\left(\left(y_{k}-\eta_{k}\right)-\frac{\bar{b}}{\mu_{0}}\right)\right. \\
& \left.-\mu_{k}\left(\left(y_{k}-\eta_{k}\right)-\frac{\bar{b}}{\mu_{0}}\right)\right)+\left(\alpha\left(\tau_{k}-1\right) \mu_{k}+\alpha^{2} \frac{\Delta x_{k}^{\top} \Delta z_{k}}{n}\right) \\
& \cdot\left(\left(y_{k}-\eta_{k}+\alpha \Delta y_{k}\right)-\frac{\bar{b}}{\mu_{0}}\right),
\end{aligned}
$$

where we used that $\mu_{k}(\alpha)=\left(\mu_{k}+\alpha\left(\tau_{k}-1\right) \mu_{k}+\alpha^{2} \frac{\Delta x_{k}^{\top} \Delta z_{k}}{n}\right)$, which can be derived from (3.25), as well as the neighbourhood conditions (3.6), and the second block equation of the Newton system (3.7). By using again the neighbourhood conditions, and then by deleting the opposite terms in the previous equation, we obtain:

$$
\begin{equation*}
\tilde{r}_{p}(\alpha)=(1-\alpha) \frac{\mu_{k} \tilde{b}_{k}+\alpha^{2}\left(\tau_{k}-1\right) \mu_{k} \Delta y_{k}+\alpha^{2} \frac{\Delta x_{k}^{\top} \Delta z_{k}}{n}\left(y_{k}-\eta_{k}+\alpha \Delta y_{k}-\frac{\bar{b}}{\mu_{0}}\right) . . . . . ~}{\text {. }} \tag{3.29}
\end{equation*}
$$

Similarly, we can show that:

$$
\begin{equation*}
\tilde{r}_{d}(\alpha)=(1-\alpha) \frac{\mu_{k}}{\mu_{0}} \tilde{c}_{k}-\alpha^{2}\left(\tau_{k}-1\right) \mu_{k} \Delta x_{k}-\alpha^{2} \frac{\Delta x_{k}^{\top} \Delta z_{k}}{n}\left(x_{k}-\zeta_{k}+\alpha \Delta x_{k}+\frac{\bar{c}}{\mu_{0}}\right) . \tag{3.30}
\end{equation*}
$$

Define the following two quantities:

$$
\begin{align*}
\xi_{2}:= & \mu_{k}\left\|\left(\Delta y_{k}, \Delta x_{k}\right)\right\|_{2} \\
& +K_{\Delta}^{2} n^{3} \mu_{k}\left(\left\|\left(y_{k}-\eta_{k}, x_{k}-\zeta_{k}\right)\right\|_{2}+\alpha^{*}\left\|\left(\Delta y_{k}, \Delta x_{k}\right)\right\|_{2}+\left\|\left(\frac{\bar{b}}{\mu_{0}}, \frac{\bar{c}}{\mu_{0}}\right)\right\|_{2}\right),  \tag{3.31}\\
\xi_{\mathcal{A}}:= & \mu_{k}\left\|\left(\Delta y_{k}, \Delta x_{k}\right)\right\|_{\mathcal{A}} \\
& +K_{\Delta}^{2} n^{3} \mu_{k}\left(\left\|\left(y_{k}-\eta_{k}, x_{k}-\zeta_{k}\right)\right\|_{\mathcal{A}}+\alpha^{*}\left\|\left(\Delta y_{k}, \Delta x_{k}\right)\right\|_{\mathcal{A}}+\left\|\left(\frac{\bar{b}}{\mu_{0}}, \frac{\bar{c}}{\mu_{0}}\right)\right\|_{\mathcal{A}}\right),
\end{align*}
$$

where $\alpha^{*}$ is given by (3.27). Using the definition of the starting point in (3.5), as well as results in Lemmas 3.3.4, 3.3.5, we can observe that $\xi_{2}=\boldsymbol{O}\left(n^{4} \mu_{k}\right)$. On the
other hand, using Assumption 2, we know that for every pair $\left(r_{1}, r_{2}\right) \in \mathbb{R}^{m+n}$, if $\left\|\left(r_{1}, r_{2}\right)\right\|_{2}=\boldsymbol{\Theta}(g(n))$, where $g(\cdot)$ is a positive polynomial function of $n$, then $\left\|\left(r_{1}, r_{2}\right)\right\|_{\mathcal{A}}=\boldsymbol{\Theta}(g(n))$. In other words, we have that $\xi_{\mathcal{A}}=\boldsymbol{O}\left(n^{4} \mu_{k}\right)$. Using the quantities in (3.31), equations (3.29), (3.30), as well as the neighbourhood conditions, we have that:

$$
\begin{aligned}
& \left\|\tilde{r}_{p}(\alpha), \tilde{r}_{d}(\alpha)\right\|_{2} \leq(1-\alpha) K_{N} \frac{\mu_{k}}{\mu_{0}}+\alpha^{2} \mu_{k} \xi_{2}, \\
& \left\|\tilde{r}_{p}(\alpha), \tilde{r}_{d}(\alpha)\right\|_{A} \leq(1-\alpha) \gamma_{\mathcal{A}} \rho \frac{\mu_{k}}{\mu_{0}}+\alpha^{2} \mu_{k} \xi_{\mathcal{A}}
\end{aligned}
$$

for all $\alpha \in\left(0, \alpha^{*}\right]$, where $\alpha^{*}$ is given by (3.27). On the other hand, we know from (3.22), that:

$$
\mu_{k}(\alpha) \geq\left(1-\alpha\left(1-\beta_{1}\right)\right) \mu_{k}, \text { for all } \alpha \in\left(0, \alpha^{*}\right]
$$

By combining the last two inequalities, we get that:

$$
\left\|\tilde{r}_{p}(\alpha), \tilde{r}_{d}(\alpha)\right\|_{2} \leq \frac{\mu_{k}(\alpha)}{\mu_{0}} K_{N}, \quad \text { for all } \alpha \in\left(0, \min \left\{\alpha^{*}, \frac{\beta_{1} K_{N}}{\xi_{2} \mu_{0}}\right\}\right]
$$

Similarly,

$$
\left\|\tilde{r}_{p}(\alpha), \tilde{r}_{d}(\alpha)\right\|_{\mathcal{A}} \leq \frac{\mu_{k}(\alpha)}{\mu_{0}} \gamma_{\mathcal{A}} \rho, \quad \text { for all } \alpha \in\left(0, \min \left\{\alpha^{*}, \frac{\beta_{1} \gamma_{\mathcal{A}} \rho}{\xi_{\mathcal{A}} \mu_{0}}\right\}\right]
$$

Hence, we have that:

$$
\begin{equation*}
\bar{\alpha}:=\min \left\{\alpha^{*}, \frac{\beta_{1} K_{N}}{\xi_{2} \mu_{0}}, \frac{\beta_{1} \gamma_{\mathcal{A}} \rho}{\xi_{\mathcal{A}} \mu_{0}}\right\}, \tag{3.32}
\end{equation*}
$$

where $\beta_{1}=\frac{\tau_{\text {min }}}{2}$. Since $\bar{\alpha}=\boldsymbol{\Omega}\left(\frac{1}{n^{4}}\right)$, we know that there must exist a constant $\bar{\kappa}>0$, independent of $n, m$ and of the iteration $k$, such that $\bar{\alpha} \geq \frac{\bar{\kappa}}{n^{4}}$ for all $k \geq 0$, and this completes the proof.
The following theorem summarizes our results.
Theorem 3.3.1. Given Assumptions 1, 2, the sequence $\left\{\mu_{k}\right\}$ generated by Algorithm IP-PMM-QP converges $Q$-linearly to zero, and the sequences of regularized residual norms

$$
\left\{\left\|A x_{k}+\mu_{k}\left(y_{k}-\eta_{k}\right)-b-\frac{\mu_{k}}{\mu_{0}} \bar{b}\right\|_{2}\right\}
$$

and

$$
\left\{\left\|-Q x_{k}+A^{\top} y_{k}+z_{k}-\mu_{k}\left(x_{k}-\zeta_{k}\right)-c-\frac{\mu_{k}}{\mu_{0}} \bar{c}\right\|_{2}\right\}
$$

converge $R$-linearly to zero.
Proof. From (3.24) we have that:

$$
\mu_{k+1} \leq\left(1-0.01 \alpha_{k}\right) \mu_{k}
$$

while, from (3.32), we know that for all $k \geq 0$, there exists $\bar{\alpha} \geq \frac{\bar{\kappa}}{n^{4}}$ such that
$\alpha_{k} \geq \bar{\alpha}$. Hence, we can easily see that $\mu_{k} \rightarrow 0$. On the other hand, from the neighbourhood conditions, we know that for all $k \geq 0$ :

$$
\left\|A x_{k}+\mu_{k}\left(y_{k}-\eta_{k}\right)-b-\frac{\mu_{k}}{\mu_{0}} \bar{b}\right\|_{2} \leq K_{N} \frac{\mu_{k}}{\mu_{0}}
$$

and

$$
\left\|-Q x_{k}+A^{\top} y_{k}+z_{k}-\mu_{k}\left(x_{k}-\zeta_{k}\right)-c-\frac{\mu_{k}}{\mu_{0}} \bar{c}\right\|_{2} \leq K_{N} \frac{\mu_{k}}{\mu_{0}} .
$$

This completes the proof.

Theorem 3.3.2. Let $\epsilon \in(0,1)$ be a given error tolerance. Choose a starting point for Algorithm IP-PMM-QP as in (3.5), such that $\mu_{0} \leq \frac{K^{*}}{\epsilon^{\omega}}$ for some positive constants $K^{*}, \omega$. Given Assumptions 1 and 2, there exists an index $K$ with:

$$
K=O\left(n^{4}\left|\log \left(\frac{1}{\epsilon}\right)\right|\right)
$$

such that the iterates $\left\{w_{k}\right\}=\left\{\left(x_{k}^{\top}, y_{k}^{\top}, z_{k}^{\top}\right)^{\top}\right\}$ generated from Algorithm IP-PMMQP satisfy:

$$
\mu_{k} \leq \epsilon, \quad \text { for all } k \geq K .
$$

Proof. The proof follows the developments in $[181,186]$ and is only provided here for completeness. Without loss of generality, we can chose $\tau_{\max } \leq 0.5$ and then from Lemma 3.3.6, we know that there is a constant $\bar{\kappa}$ independent of $n$ such that $\bar{a} \geq \frac{\bar{\varepsilon}}{n^{4}}$, where $\bar{a}$ is the worst-case step-length. Given the latter, we know that the new iterate lies in the neighbourhood $\mathcal{N}_{\mu_{k+1}}\left(\zeta_{k+1}, \eta_{k+1}\right)$ defined in (3.6). We also know, from (3.24), that:

$$
\mu_{k+1} \leq(1-0.01 \bar{a}) \mu_{k} \leq\left(1-0.01 \frac{\bar{\kappa}}{n^{4}}\right) \mu_{k}, \quad k=0,1,2, \ldots
$$

By taking logarithms on both sides in the previous inequality, we get:

$$
\log \left(\mu_{k+1}\right) \leq \log \left(1-\frac{\tilde{\kappa}}{n^{4}}\right)+\log \left(\mu_{k}\right)
$$

where $\tilde{\kappa}=0.01 \bar{\kappa}$. By applying repeatedly the previous formula, and using the fact that $\mu_{0} \leq \frac{K^{*}}{\epsilon^{\omega}}$, we have:

$$
\log \left(\mu_{k}\right) \leq k \log \left(1-\frac{\tilde{\kappa}}{n^{4}}\right)+\log \left(\mu_{0}\right) \leq k \log \left(1-\frac{\tilde{\kappa}}{n^{4}}\right)+\omega \log \left(\frac{1}{\epsilon}\right)+\log \left(K^{*}\right) .
$$

We use the fact that $\log (1+\beta) \leq \beta$, for all $\beta>-1$ to obtain:

$$
\log \left(\mu_{k}\right) \leq k\left(-\frac{\tilde{\kappa}}{n^{4}}\right)+\omega \log \left(\frac{1}{\epsilon}\right)+\log \left(K^{*}\right) .
$$

Hence, convergence is attained if:

$$
k\left(-\frac{\tilde{\kappa}}{n^{4}}\right)+\omega \log \left(\frac{1}{\epsilon}\right)+\log \left(K^{*}\right) \leq \log (\epsilon) .
$$

The latter holds for all $k$ satisfying:

$$
k \geq K=\frac{n^{4}}{\tilde{\kappa}}\left((1+\omega) \log \left(\frac{1}{\epsilon}\right)+\log \left(K^{*}\right)\right)
$$

which completes the proof.
Finally, we present the global convergence guarantee of Algorithm IP-PMMQP.

Theorem 3.3.3. Suppose that we allow Algorithm IP-PMM-QP to run indefinitely. Then, if Assumptions 1 and 2 hold, every limit point of $\left\{\left(x_{k}, y_{k}, z_{k}\right)\right\}$ determines a primal-dual solution of the non-regularized pair (CQP)-(CQD).

Proof. From Theorem 3.3.1, we know that $\left\{\mu_{k}\right\} \rightarrow 0$, and hence, there exists a sub-sequence $\mathcal{K} \subseteq \mathbb{N}$, such that:

$$
\begin{gathered}
\left\{A x_{k}+\mu_{k}\left(y_{k}-\eta_{k}\right)-b-\frac{\mu_{k}}{\mu_{0}} \bar{b}\right\}_{\mathcal{K}} \rightarrow 0_{m}, \\
\left\{-Q x_{k}+A^{\top} y_{k}+z_{k}-\mu_{k}\left(x_{k}-\zeta_{k}\right)-c-\frac{\mu_{k}}{\mu_{0}} \bar{c}\right\}_{\mathcal{K}} \rightarrow 0_{n}
\end{gathered}
$$

However, since Assumptions 1 and 2 hold, we know from Lemma 3.3.4 that $\left\{\left(x_{k}, y_{k}, z_{k}\right)\right\}$ is a bounded sequence. Hence, we obtain that:

$$
\left\{A x_{k}-b\right\}_{\mathcal{K}} \rightarrow 0_{m}, \quad\left\{-Q x_{k}+A^{\top} y_{k}+z_{k}-c\right\}_{\mathcal{K}} \rightarrow 0_{n}
$$

One can readily observe that the limit point of the algorithm satisfies the conditions given in (1.4) (where $\left.f(x)=c^{\top} x+(1 / 2) x^{\top} Q x\right)$, since $\mu_{k}=\left(x_{k}^{\top} z_{k}\right) / n$.

Remark 1. Let us notice that the above analysis required Assumption 2, and hence that $\operatorname{rank}(A)=m$. However, one motivation for using regularization is that the resulting algorithm is able to solve rank deficient problems. While the complexity result would not hold in this case, we are able to perform an analysis following exactly the developments in [72], which would guarantee the global convergence of IP-PMM, given certain assumptions (see [72]). The latter is omitted for brevity of presentation.

Remark 2. As mentioned at the end of Section 3.2, we do not study the conditions under which one can guarantee that $x_{k}-\zeta_{k} \rightarrow 0_{n}$ and $y_{k}-\eta_{k} \rightarrow 0_{m}$, although this could be possible. This is because the method is shown to converge globally even if this is not the case. Indeed, notice that if one were to choose $\zeta_{0}=0_{n}$ and $\eta_{0}=0_{m}$, and simply ignore the last conditional statement of Algorithm IP-PMM-QP, the convergence analysis established in this section would still hold. In this case, the method would be interpreted as an interior point-quadratic penalty
method, and we could consider the regularization as a diminishing primal-dual Tikhonov regularizer (i.e. a variant of the regularization proposed in [154]).

### 3.4 Infeasible problems

Let us now drop Assumptions 1, 2, in order to analyze the behaviour of the algorithm in the case where an infeasible problem is tackled. Let us employ the following two premises:

Premise 1. During the iterations of Algorithm IP-PMM-QP, the sequences $\left\{\| y_{k}-\right.$ $\left.\eta_{k} \|_{2}\right\}$ and $\left\{\left\|x_{k}-\zeta_{k}\right\|_{2}\right\}$, remain bounded.

Premise 2. There does not exist a primal-dual triple, satisfying the KKT conditions for the primal-dual pair (CQP)-(CQD).

The following analysis is based on the developments in [4] and [72]. However, in these papers such an analysis is proposed in order to derive convergence of an IPM, while here, we use it as a tool in order to construct a reliable and implementable infeasibility detection mechanism. In what follows, we show that Premises 1 and 2 are contradictory. In other words, if Premise 2 holds (which means that the problem is infeasible), then we will show that Premise 1 cannot hold, and hence the negation of Premise 1 is a necessary condition for infeasibility.

Lemma 3.4.1. Given Premise 1, and by assuming that $x_{k}^{\top} z_{k}>\epsilon$, for some $\epsilon>0$, for all iterations $k$ of Algorithm IP-PMM-QP, the Newton direction produced by (3.7) is uniformly bounded by a constant dependent only on $n$.

Proof. Let us use a variation of Theorem 1 given in [4]. This theorem states that if the following conditions are satisfied,

1. $\mu_{k}>0$,
2. there exists $\bar{\epsilon}>0: x_{k}^{i} z_{k}^{i} \geq \bar{\epsilon}$, for all $i=\{1,2, \ldots, n\}$, and for all $k \geq 0$,
3. and the matrix $H_{k}=\mu_{k} I+Q+X_{k}^{-1} Z_{k}+\frac{1}{\mu_{k}} A^{\top} A$ is positive definite,
then the Jacobian matrix in (3.7) is non-singular and has a uniformly bounded inverse. Note that (1.), (3.) are trivial to verify, based on the our assumption that $x_{k}^{\top} z_{k}=n \mu_{k}>\epsilon$. Condition (2.) follows since we know that our iterates lie in $\mathcal{N}_{\mu_{k}}\left(\zeta_{k}, \eta_{k}\right)$, while we have $x_{k}^{\top} z_{k}>\epsilon$, by assumption. Indeed, from the neighbourhood conditions (3.6), we have that $x_{k}^{i} z_{k}^{i} \geq\left(\left(\gamma_{\mu}\right) / n\right) x_{k}^{\top} z_{k}$. Hence, there exists $\bar{\epsilon}=\left(\gamma_{\mu} \epsilon\right) / n>0$ such that $x_{k}^{i} z_{k}^{i}>\bar{\epsilon}$, for all $k \geq 0$, for all $i=\{1, \ldots, n\}$.

Finally, we have to show that the right hand side of (3.7) is uniformly bounded. To that end, we bound the right-hand side of the second block equation of (3.7) as follows:

$$
\left\|A x_{k}+\tau_{k} \mu_{k}\left(y_{k}-\eta_{k}\right)-b-\frac{\tau_{k} \mu_{k}}{\mu_{0}} \bar{b}+\mu_{k}\left(y_{k}-\eta_{k}-\frac{\bar{b}}{\mu_{0}}\right)-\mu_{k}\left(y_{k}-\eta_{k}-\frac{\bar{b}}{\mu_{0}}\right)\right\|_{2},
$$

where we used the neighbourhood conditions (3.6). Boundedness follows from Premise 1. A similar reasoning applies for bounding the right-hand side of the first block equation, while the right-hand side of the third block equation is bounded directly from the neighbourhood conditions. Combining the previous completes the proof.

In the following lemma, we prove by contradiction that the parameter $\mu_{k}$ of Algorithm IP-PMM-QP converges to zero, given that Premise 1 holds. The proof is based on the developments in $[72,102]$ and is only partially given here, for ease of presentation.

Lemma 3.4.2. Given Premise 1, and a sequence $\left(x_{k}, y_{k}, z_{k}\right) \in \mathcal{N}_{\mu_{k}}\left(\zeta_{k}, \eta_{k}\right)$ produced by Algorithm IP-PMM-QP, the sequence $\left\{\mu_{k}\right\}$ converges to zero.

Proof. Assume, by virtue of contradiction, that $\mu_{k}>\epsilon$, for all $k \geq 0$. Then, we know that the Newton direction obtained by the algorithm at every iteration, after solving (3.7), will be uniformly bounded by a constant dependent only on $n$, that is, there exists a positive constant $K^{\dagger}$, such that $\left\|\left(\Delta x_{k}, \Delta y_{k}, \Delta z_{k}\right)\right\|_{2} \leq K^{\dagger}$. As in Lemma 3.3.6, we define:

$$
\tilde{r}_{p}(\alpha)=A x_{k}(\alpha)+\mu_{k}(\alpha)\left(y_{k}(\alpha)-\eta_{k}\right)-b-\frac{\mu_{k}(\alpha)}{\mu_{0}} \bar{b}
$$

and

$$
\tilde{r}_{d}(\alpha)=-Q x_{k}(\alpha)+A^{\top} y_{k}(\alpha)+z_{k}(\alpha)-\mu_{k}(\alpha)\left(x_{k}(\alpha)-\zeta_{k}\right)-c-\frac{\mu_{k}(\alpha)}{\mu_{0}} \bar{c}
$$

for which we know that equalities (3.29) and (3.30) hold, respectively. Take any $k \geq 0$ and define the following functions:

$$
\begin{aligned}
& f_{1}(\alpha):=\left(x_{k}+\alpha \Delta x_{k}\right)^{\top}\left(z_{k}+\alpha \Delta z_{k}\right)-\left(1-\alpha\left(1-\frac{\tau_{\min }}{2}\right)\right) x_{k}^{\top} z_{k}, \\
& f_{2}^{i}(\alpha):=\left(x_{k}^{i}+\alpha \Delta x_{k}^{i}\right)\left(z_{k}^{i}+\alpha \Delta z_{k}^{i}\right)-\gamma_{\mu} \mu_{k}(\alpha), i=1, \cdots, n, \\
& f_{3}(\alpha):=(1-0.01 \alpha) x_{k}^{\top} z_{k}-\left(x_{k}(\alpha)\right)^{\top}\left(z_{k}(\alpha)\right), \\
& g_{2}(\alpha):=\frac{\mu_{k}(\alpha)}{\mu_{0}} K_{N}-\left\|\left(\tilde{r}_{p}(\alpha), \tilde{r}_{d}(\alpha)\right)\right\|_{2},
\end{aligned}
$$

where $\mu_{k}(\alpha)=\frac{\left(x_{k}+\alpha \Delta x_{k}\right)^{\top}\left(z_{k}+\alpha \Delta z_{k}\right)}{n},\left(x_{k}(\alpha), y_{k}(\alpha), z_{k}(\alpha)\right)=\left(x_{k}+\alpha \Delta x_{k}, y_{k}+\right.$ $\left.\alpha \Delta y_{k}, z_{k}+\alpha \Delta z_{k}\right)$. We would like to show that there exists $\alpha^{*}>0$ such that:

$$
f_{1}(\alpha) \geq 0, f_{2}^{i}(\alpha) \geq 0, \text { for all } i=1, \ldots, n, f_{3}(\alpha) \geq 0, g_{2}(\alpha) \geq 0
$$

for all $\alpha \in\left(0, \alpha^{*}\right]$. These conditions model the requirement that the next iteration of Algorithm IP-PMM-QP must lie in the updated neighbourhood: $\mathcal{N}_{\mu_{k+1}}\left(\zeta_{k}, \eta_{k}\right)$. Note that Algorithm IP-PMM-QP updates the parameters $\eta_{k}, \zeta_{k}$ only if the selected new iterate belongs to the new neighbourhood, defined using the updated parameters. Hence, it suffices to show that $\left(x_{k+1}, y_{k+1}, z_{k+1}\right) \in \mathcal{N}_{\mu_{k+1}}\left(\zeta_{k}, \eta_{k}\right)$. It is important to observe that we do not require that the scaled infeasibilities are
bounded with respect to the semi-norm defined in (3.4). This is because the aforementioned norm requires that $\operatorname{rank}(A)=m$. In other words, we are using a slightly different neighbourhood, and that does not affect the global convergence of the method (but only its complexity).

Proving the existence of $\alpha^{*}>0$, such that each of the aforementioned functions is positive, follows exactly the developments in Lemma 3.3.6, with the only difference being that the bounds on the directions are not explicitly specified in this case. Using the same methodology as in Lemma 3.3.6, while keeping in mind our assumption, namely $x_{k}^{\top} z_{k}>\epsilon$, and hence $x_{k}^{i} z_{k}^{i}>\bar{\epsilon}$, we can show that:

$$
\begin{equation*}
\alpha^{*}:=\min \left\{1, \frac{\tau_{\min } \epsilon}{2\left(K^{\dagger}\right)^{2}}, \frac{\left(1-\gamma_{\mu}\right) \tau_{\min } \bar{\epsilon}}{2\left(K^{\dagger}\right)^{2}}, \frac{0.49 \epsilon}{2\left(K^{\dagger}\right)^{2}}, \frac{\tau_{\min } K_{N} \epsilon}{2 \mu_{0}\left(\xi_{2}\right)}\right\}, \tag{3.33}
\end{equation*}
$$

where $\xi_{2}$ is a bounded constant dependent on $K^{\dagger}$, and defined as in (3.31). However, using the inequality:

$$
\mu_{k+1} \leq(1-0.01 \alpha) \mu_{k}, \text { for all } \alpha \in\left[0, \alpha^{*}\right],
$$

we get that $\mu_{k} \rightarrow 0$, which contradicts our assumption that $\mu_{k}>\epsilon$, for all $k \geq 0$, and completes the proof.

Finally, using the following theorem, we derive a necessary condition for infeasibility.

Theorem 3.4.1. Given Premise 2, i.e. there does not exist a KKT triple for the pair (CQP)-(CQD), then Premise 1 fails to hold.

Proof. By virtue of contradiction, let Premise 1 hold. In Lemma 3.4.2, we proved that given Premise 1, Algorithm IP-PMM-QP produces iterates that belong to the neighbourhood (3.6) and $\mu_{k} \rightarrow 0$. But from the neighbourhood conditions we can observe that:

$$
\left\|A x_{k}+\mu_{k}\left(y_{k}-\eta_{k}\right)-b-\frac{\mu_{k}}{\mu_{0}} \bar{b}\right\|_{2} \leq K_{N} \frac{\mu_{k}}{\mu_{0}},
$$

and

$$
\left\|-Q x_{k}+A^{\top} y_{k}+z_{k}-\mu_{k}\left(x_{k}-\zeta_{k}\right)-c-\frac{\mu_{k}}{\mu_{0}} \bar{c}\right\|_{2} \leq K_{N} \frac{\mu_{k}}{\mu_{0}} .
$$

Hence, we can choose a sub-sequence $\mathcal{K} \subseteq \mathbb{N}$, for which:

$$
\left\{A x_{k}+\mu_{k}\left(y_{k}-\eta_{k}\right)-b-\frac{\mu_{k}}{\mu_{0}} \bar{b}\right\}_{\mathcal{K}} \rightarrow 0_{m},
$$

and

$$
\left\{-Q x_{k}+A^{\top} y_{k}+z_{k}-\mu_{k}\left(x_{k}-\zeta_{k}\right)-c-\frac{\mu_{k}}{\mu_{0}} \bar{c}\right\}_{\mathcal{K}} \rightarrow 0_{n} .
$$

But since $\left\|y_{k}-\eta_{k}\right\|_{2}$ and $\left\|x_{k}-\zeta_{k}\right\|_{2}$ are bounded, while $\mu_{k} \rightarrow 0$, we have that:

$$
\left\{A x_{k}-b\right\}_{\mathcal{K}} \rightarrow 0_{m},\left\{c+Q x_{k}-A^{\top} y_{k}-z_{k}\right\}_{\mathcal{K}} \rightarrow 0_{n}, \text { and }\left\{x_{k}^{\top} z_{k}\right\}_{\mathcal{K}} \rightarrow 0 .
$$

This contradicts Premise 2, i.e. that the pair (CQP)-(CQD) does not have a KKT triple, and completes the proof.

In the previous theorem, we proved that Premise 1 is a necessary condition for infeasibility, since otherwise, we arrive at a contradiction. Nevertheless, this does not mean that the condition is also sufficient. In order to obtain a more reliable test for infeasibility, that uses the previous result, we will have to use the properties of Algorithm IP-PMM-QP. In particular, we can notice that if the primal-dual problem is infeasible, then the PMM sub-problem will stop being updated after a finite number of iterations. In that case, we know from Theorem 3.4.1 that the sequence $\left\|\left(x_{k}-\zeta_{k}, y_{k}-\eta_{k}\right)\right\|_{2}$ will grow unbounded. Hence, we can define a maximum number of iterations per PMM sub-problem, say $k_{\dagger} \geq 0$, as well as a very large constant $K_{\dagger} \geq 0$. Then, if $\left\|\left(x_{k}-\zeta_{k}, y_{k}-\eta_{k}\right)\right\|_{2}>K_{\dagger}$ and $k \geq k_{\dagger}$, the algorithm is terminated. The specific choices for these constants will be given in the next section.

Remark 3. Let us notice that the analysis in Section 3.3 employs the standard assumptions used when analyzing a non-regularized IPM. However, the method could still be useful if these assumptions were not met. Indeed, if for example the constraint matrix was not of full row rank, one could still prove global convergence of the method, using the methodology employed in this section by assuming that Premise 1 holds and Premise 2 does not. Furthermore, in practice the method would not encounter any numerical issues with the inversion of the Newton system (see [4]). Nevertheless, showing polynomial complexity in this case is still an open problem. The aim of this work is to show that under the standard Assumptions 1, 2, Algorithm IP-PMM-QP is able to enjoy polynomial complexity, while having to solve better conditioned systems than those solved by standard IPMs at each iteration, thus ensuring better stability (and as a result better robustness and potentially better efficiency).

### 3.5 Computational experience

In this section, we provide some implementation details and present computational results of the method, over a set of small to large scale linear and convex quadratic programming problems. The code was written in MATLAB and is available via the ERGO webpage ${ }^{1}$.

### 3.5.1 Implementation details

Our implementation deviates from the theory, in order to gain some additional control, as well as computational efficiency. Nevertheless, the theory has served as a guideline to tune the code reliably. There are two major differences between the practical implementation of IP-PMM and its theoretical algorithmic counterpart. Firstly, our implementation uses different penalty parameters for the proximal terms and the logarithmic barrier term. In particular, we define a primal proximal

[^0]penalty $\rho$, a dual proximal penalty $\delta$ and the barrier parameter $\mu$. Using the previous, the PMM Lagrangian function in (3.1), at an arbitrary iteration $k$ of the algorithm, becomes:
$\mathcal{L}_{\mu_{k}, \delta_{k}, \rho_{k}}^{P M M}\left(x ; \zeta_{k}, \lambda_{k}\right)=c^{\top} x+\frac{1}{2} x^{\top} Q x-\lambda_{k}^{\top}(A x-b)+\frac{1}{2 \delta_{k}}\|A x-b\|_{2}^{2}+\frac{\rho_{k}}{2}\left\|x-\zeta_{k}\right\|_{2}^{2}$,
and (3.2) is altered similarly. The second difference lies in the fact that we do not require the iterates of the method to lie in the neighbourhood defined in (3.6) in order to gain efficiency. In what follows, we provide further details concerning our implementation choices.

## Free variables

The method takes as input problems in the following form:

$$
\min _{x}\left(c^{\top} x+\frac{1}{2} x^{\top} Q x\right) \text {, s.t. } A x=b, x^{\mathcal{I}} \geq 0_{|\mathcal{I}|}, x^{\mathcal{F}} \text { free, }
$$

where $\mathcal{I}=\{1, \cdots, n\} \backslash \mathcal{F}$ is the set of indices indicating the non-negative variables. In particular, if a problem instance has only free variables, no logarithmic barrier is employed and the method reduces to a standard proximal method of multipliers. Of course in this case, the derived complexity result does not hold. Nevertheless, a global convergence result holds, as shown in [146]. In general, convex quadratic optimization problems with only equality constraints are usually easy to deal with, and the proposed algorithm behaves very well when solving such problems in practice.

## Constraint matrix scaling

In the pre-processing stage, we check if the constraint matrix is well scaled, i.e if:

$$
\left(\max _{i \in\{1, \ldots, m\}, j \in\{1, \ldots, n\}}\left(\left|A^{(i, j)}\right|\right)<10\right) \wedge\left(\min _{i \in\{1, \ldots, m\}, j \in\{1, \ldots, n\}:\left|A^{(i, j)}\right|>0}\left(\left|A^{(i, j)}\right|\right)>0.1\right) .
$$

If the previous is not satisfied, we apply geometric scaling in the rows of $A$, that is, we multiply each row of $A$ by a scalar of the form:

$$
d^{i}=\frac{1}{\sqrt{\max _{j \in\{1, \cdots, n\}}\left(\left|A^{(i, j)}\right|\right) \cdot \min _{j \in\{1, \ldots, n\}:\left|A^{(i, j)}\right|>0}\left(\left|A^{(i, j)}\right|\right)}}, \quad \text { for all } i \in\{1, \ldots, m\} .
$$

However, for numerical stability, we find the largest integer $p^{i}$, such that $2^{p^{i}} \leq d^{i}$ and we set $d^{i}=2^{p^{i}}$, for all $i \in\{1, \ldots, m\}$. Hence, the scaling factors are powers of two. Based on the IEEE representation of floating point numbers, multiplying by a power of 2 translates into an addition of this power to the exponent, without affecting the mantissa. This scaling technique is based on the one used within the GLPK solver (see [76]).

## Starting point, Newton-step computation and step-length

We use the same starting point as the one in Section 2.4.2. The only difference in this case is that we only shift components of the variables $x$ and $z$ that belong to the index set $\mathcal{I}$.

In order to find the Newton step, we employ a widely used predictor-corrector method. The practical implementation deviates from the theory at this point, in order to gain computational efficiency. We provide the algorithmic scheme in Algorithm PC for completeness, but the reader is referred to [119] for a detailed discussion of this method. Solving two different systems serves as a way of decreasing the infeasibility and allowing $\mu_{k}$ (and hence $\delta_{k}, \rho_{k}$ ) to decrease.

We solve the systems (3.34) and (3.35) (see Algorithm PC, p. 93), using the built-in MATLAB symmetric decomposition (i.e. ldl). In order to exploit quasidefiniteness, we change the default pivot threshold of ldl to a value slightly lower than the minimum allowed regularization value (reg ${ }_{t h r}$; specified in sub-section 3.5.1). Such a small pivot threshold guarantees that no $2 \times 2$ pivots will be employed during the factorization process.

## PMM parameters

At this point, we discuss how we update the PMM sub-problems in practice, as well as how we tune the penalty parameters (regularization parameters) $\delta, \rho$. Notice that in Section 3.3 we set $\delta_{k}=\rho_{k}=\mu_{k}$. While this is beneficial in theory, as it gives us a reliable way of tuning the penalty parameters of the algorithm, it is not very beneficial in practice, as it does not allow us to control the regularization parameters in the cases of extreme ill-conditioning. Hence, we allow the use of different penalty parameters connected to the PMM subproblems, while enforcing both parameters $\left(\delta_{k}, \rho_{k}\right)$ to decrease at the same rate as $\mu_{k}$ (based on the theoretical results in Sections 3.3, 3.4).

On the other hand, the algorithm is more optimistic in practice than it is in theory. In particular, we do not consider the semi-norm (3.4) of the infeasibility, while we allow the update of the estimates $\eta_{k}, \zeta_{k}$, to happen independently. In particular, the former is updated when the 2-norm of the primal infeasibility is sufficiently reduced, while the latter is updated based on the dual infeasibility.

More specifically, at the beginning of the optimization, we set: $\delta_{0}=8, \rho_{0}=8$, $\eta_{0}=y_{0}, \zeta_{0}=x_{0}$. Then, at the end of every iteration, we employ the algorithmic scheme given in Algorithm PEU. In order to ensure numerical stability, we do not allow $\delta$ or $\rho$ to become smaller than a suitable positive value, reg ${ }_{t h r}$. We set: reg $_{\text {thr }}=\max \left\{\frac{\text { tol }}{\max \left\{\|A\|_{\infty}^{2},\|Q\|_{\infty}^{2}\right\}}, 10^{-10}\right\}$. This value is based on the developments in Chapter 2, in order to ensure that we introduce a controlled perturbation in the eigenvalues of the non-regularized linear system. If the factorization fails, we increase the regularization parameters by a factor of 10 and repeat the factorization. If the factorization fails while either $\delta$ or $\rho$ have reached their minimum allowed value $\left(\mathrm{reg}_{t h r}\right)$, then we also increase this value by a factor of 10 . If this occurs 5 consecutive times, the method is terminated with a message indicating ill-conditioning.

## Termination criteria

There are four possible termination criteria. They are summarized in Algorithm TC. In the aforementioned algorithm, tol represents the error tolerance chosen by the user. Similarly, IP $_{\text {maxit }}$ is the maximum number of allowed IPM iterations, also chosen by the user. On the other hand, $\mathrm{PMM}_{\operatorname{maxit}}$ is a threshold indicating that the PMM sub-problem needs too many iterations before being updated (that is if $k_{\mathrm{PMM}}>\mathrm{PMM}_{\operatorname{maxit}}$ ). We set $\mathrm{PMM}_{\operatorname{maxit}}=5$. When either $\eta_{k}$ or $\zeta_{k}$ is updated, we set $k_{\text {PMM }}=0$.

Let us now support the proposed infeasibility detection mechanism. In particular, notice that as long as the penalty parameters do not converge to zero, every PMM-subproblem must have a solution, even in the case of infeasibility. Hence, we expect convergence of the regularized primal (dual, respectively) infeasibility to zero, while from Section 3.4, we know that a necessary condition for infeasibility is that the sequence $\left\|\left(x_{k}-\zeta_{k}, y_{k}-\eta_{k}\right)\right\|_{2}$ diverges. If this behaviour is observed, while the PMM parameters $\eta_{k}, \zeta_{k}$ are not updated (which is not expected to happen in the feasible but rank deficient case), then we can conclude that the problem under consideration is infeasible.

### 3.5.2 Numerical results

At this point, we present the computational results obtained by solving a set of small to large scale linear and convex quadratic problems. In order to stress out the importance of regularization, we compare IP-PMM with a non-regularized IPM. The latter implementation follows exactly from the implementation of IPPMM, simply by fixing $\delta$ and $\rho$ to zero. In the non-regularized case, the minimum pivot of the ldl function is restored to its default value, in order to avoid numerical instability. Throughout all of the presented experiments, we set the number of maximum iterations to 200 . It should be noted here that we expect IP-PMM to require more iterations to converge, as compared to the non-regularized IPM. In turn, the Newton systems arising in IP-PMM have better numerical properties (accelerating the factorization process), while overall the method is expected to be significantly more stable. In what follows, we demonstrate that this increase in the number of iterations is benign, in that it does not make the resulting method inefficient. In contrast, we provide computational evidence that the acceleration of the factorization process more than compensates for the increase in the number of iterations. The experiments were conducted on a PC with a 2.2 GHz Intel Core i7 processor (hexa-core), 16GB RAM, run under Windows 10 operating system. The MATLAB version used was R2018b.

## Linear programming problems

Let us compare the proposed method with the respective non-regularized implementation, over the Netlib collection, [130]. The test set consists of 96 linear programming problems. We set the desired tolerance to tol $=10^{-6}$. Firstly, we compare the two methods, without using the pre-solved version of the problem collection (e.g. allowing rank-deficient matrices). In this case, the non-regularized IPM converged for only 66 out of the total 96 problems. On the other hand, IP-

PMM solved successfully the whole set, in 160 seconds (and a total of 2,609 IPM iterations). Hence, one of the benefits of regularization, that of alleviating rank deficiency of the constraint matrix, becomes immediately obvious.

However, in order to explore more potential benefits of regularization, we run the algorithm on a pre-solved Netlib library. In the pre-solved set, the nonregularized IPM converged for 93 out of 96 problems. The three failures occurred due to instability of the Newton system. The overall time spent was 353 seconds (and a total of 1,871 IPM iterations). On the other hand, IP-PMM solved the whole set in 161 seconds (and a total of 2,367 iterations). Two more benefits of regularization become obvious here. Firstly, we can observe that numerical stability can be a problem in a standard IPM, even if we ensure that the constraint matrices are of full row rank. Secondly, notice that despite the fact that IPPMM required $26 \%$ more iterations, it still solved the whole set in $55 \%$ less CPU time. This is because in IP-PMM, only diagonal pivots are allowed during the factorization. We could enforce the same condition on the non-regularized IPM, but then significantly more problems would fail to converge ( $22 / 96$ ) due to numerical instability (e.g. see the numerical results in Section 2.4.3, where this strategy was adopted).

In Table 3.1, we collect statistics from the runs of the two methods over some medium scale instances of the pre-solved Netlib test set.

Table 3.1: Medium-scale Netlib problems

| Name | $\boldsymbol{n} \boldsymbol{n} \boldsymbol{z}(\boldsymbol{A})$ | IP-PMM |  |  | IPM |  |  |
| ---: | ---: | ---: | ---: | ---: | ---: | ---: | ---: |
|  |  |  | Time (s) | IP-Iter. |  | Time (s) | IP-Iter. |
| 80BAU3B | 29,063 | 1.43 | 44 |  | 9.68 | 40 |  |
| D6CUBE | 43,888 | 1.26 | 25 |  | 9.64 | 22 |  |
| DFL001 | 41,873 | 25.42 | 47 |  | $\dagger^{1}$ | $\dagger$ |  |
| FIT2D | 138,018 | 8.52 | 27 |  | 23.94 | 25 |  |
| FIT2P | 60,784 | 1.24 | 24 |  | 1.56 | 19 |  |
| PILOT87 | 73,804 | 7.21 | 49 |  | 95.04 | 46 |  |
| QAP15 | 110,700 | 91.78 | 23 |  | 93.56 | 18 |  |

${ }^{1} \dagger$ indicates that the solver did not reach the desired accuracy.

From Table 3.1, it becomes obvious that the factorization efficiency is significantly improved by the introduction of the regularization terms. In all of the presented instances, IP-PMM converged needing more iterations, but requiring less CPU time.

In order to summarize the comparison of the two methods, we include Figure 3.1, which contains the performance profiles ([58]), over the pre-solved Netlib set, of the two methods. IP-PMM is represented by the green line (consisting of triangles), while non-regularized IPM by the blue line (consisting of stars). In Figure 3.1a, we present the performance profile with respect to time required for convergence, while in Figure 3.1b, the performance profile with respect to the number of iterations. One can see that all of our previous observations are verified in Figure 3.1.

Figure 3.1: Performance profiles over the pre-solved Netlib test set.


## Infeasible problems

In order to assess the accuracy of the proposed infeasibility detection criteria, we attempt to solve 28 infeasible problems, coming from the Netlib infeasible collection ([130], see also Infeasible Problems). For 22 out of the 28 problems, the method was able to recognize that the problem under consideration is infeasible, and exit before the maximum number of iterations was reached. There were four problems, for which the method terminated after reaching the maximum number of iterations. For one problem the method was terminated early due to numerical instability. Finally, there was one problem for which the method converged to the least squares solution, which satisfied the optimality conditions for a tolerance of $10^{-6}$. Overall, IP-PMM run all 28 infeasible problems in 34 seconds (and a total of 1,813 IPM iterations). The proposed infeasibility detection mechanism had a $78 \%$ rate of success over the infeasible test set, while no feasible problem was misclassified as infeasible. A more accurate infeasibility detection mechanism could be possible, however, the proposed approach is easy to implement and cheap from the computational point of view. Nevertheless, the interested reader is referred to $[8,129,183]$ and the references therein, for various other infeasibility detection methods.

## Quadratic programming problems

Next, we present the comparison of the two methods over the Maros-Mészáros test set ([116]), which is comprised of 122 convex quadratic programming problems. Notice that in the previous experiments we used the pre-solved version of the collection. However, we do not have a pre-solved version of this test set available. Since the focus of this thesis is not on the pre-solve phase of convex problems, we present the comparison of the two methods over the set, without applying any pre-processing. As a consequence, non-regularized IPM fails to solve 27 out of the total 122 problems. However, only 11 out of 27 failed due to rank deficiency. The remaining 16 failures occurred due to numerical instability. On the contrary, IP-PMM solved the whole set successfully in 127 seconds (and a total of 3,014 iterations). As before, the required tolerance was set to $10^{-6}$.

In Table 3.2, we collect statistics from the runs of the two methods over some medium scale instances of the Maros-Mészáros collection.

In order to summarize the comparison of the two methods, we include Figure

Table 3.2: Medium-scale Maros-Mészáros problems

| Name | $\boldsymbol{n} \boldsymbol{n z ( A )}$ | $\boldsymbol{n} \boldsymbol{n}(\boldsymbol{Q})$ | IP-PMM |  |  | IPM |  |
| ---: | ---: | ---: | ---: | ---: | ---: | ---: | ---: | ---: |
|  |  |  | Time (s) | IP-Iter. |  | Time (s) | IP-Iter. |
| AUG2DCQP | 40,000 | 40,400 | 4.70 | 83 |  | 7.21 | 111 |
| CVXQP1L | 14,998 | 69,968 | 25.54 | 38 |  | $\dagger$ | $\dagger$ |
| CVXQP3L | 22,497 | 69,968 | 45.69 | 59 | $\dagger$ | $\dagger$ |  |
| LISWET1 | 30,000 | 10,002 | 1.07 | 30 | 1.86 | 40 |  |
| POWELL20 | 20,000 | 10,000 | 1.26 | 30 | 1.61 | 25 |  |
| QSHIP12L | 16,170 | 122,433 | 0.91 | 23 | $\dagger$ | $\dagger$ |  |
| STCQP1 | 13,338 | 49,109 | 0.38 | 16 | 6.89 | 13 |  |

3.2, which contains the performance profiles, over the Maros-Mészáros test set, of the two methods. IP-PMM is represented by the green line (consisted of triangles), while non-regularized IPM by the blue line (consisted of stars). In Figure 3.2a, we present the performance profile with respect to time required for convergence, while in Figure 3.2b, the performance profile with respect to the number of iterations.

Figure 3.2: Performance profiles over the Maros-Mészáros test set.


Similar remarks can be made here, as those given when summarizing the linear programming experiments. One can readily observe the importance of the stability introduced by the regularization. On the other hand, the overhead in terms of number of iterations that is introduced due to regularization is acceptable due to the acceleration of the factorization (since we are guaranteed to have a quasi-definite augmented system).

## Verification of the theory

We have already presented the benefits of using regularization in interior point methods. A question arises, as to whether a regularized IPM can actually find an exact solution of the problem under consideration. Theoretically, we have proven this to be the case. However, in practice one is not allowed to decrease the regularization parameters indefinitely, since ill-conditioning will become a problem. Based on the theory of augmented Lagrangian methods, one knows that sufficiently small regularization parameters suffice for exactness (see [27,

71], among others). In what follows, we provide a "table of robustness" of IPPMM. We run the method over the Netlib and the Maros-Mészáros collections, for decreasing values of the required tolerance and report the number of problems that converged.

Table 3.3: Table of robustness

| Test Set | Tolerance | Problems Converged |
| ---: | ---: | ---: |
| Netlib (non-presolved) | $10^{-6}$ | $96 / 96$ |
| $"$ | $10^{-8}$ | $95 / 96$ |
| $"$ | $10^{-10}$ | $94 / 96$ |
| Netlib (presolved) | $10^{-6}$ | $96 / 96$ |
| $"$ | $10^{-8}$ | $94 / 96$ |
| $"$ | $10^{-10}$ | $94 / 96$ |
| Maros-Mészáros | $10^{-6}$ | $122 / 122$ |
| $"$ | $10^{-8}$ | $121 / 122$ |
| $"$ | $10^{-10}$ | $112 / 122$ |

One can observe from Table 3.3 that IP-PMM is sufficiently robust. Even in the case where a 10 digit accurate solution is required, the method is able to maintain a success rate larger than $91 \%$.

## Large scale problems

All of our previous experiments were conducted on small to medium scale linear and convex quadratic programming problems. We have shown (both theoretically and practically) that the proposed method is reliable. However, it is worth mentioning the limitations of the current approach. Since we employ exact factorization during the iterations of the IPM, we expect that the method will be limited in terms of the size of the problems it can solve. The main bottleneck arises from the factorization, which does not scale well in terms of processing time and memory requirements. In order to explore the limitations, in Table 3.4 we provide the statistics of the runs of the method over a small set of large scale problems. It contains the number of non-zeros of the constraint matrices, as well as the time needed to solve the problem. The tolerance used in these experiments was $10^{-6}$.

From Table 3.4, it can be observed that as the dimension of the problem grows, the time to convergence is significantly increased. This increase in time is disproportionate for some problems. This is because the required memory could exceed the available RAM, in which case the swap-file is activated. Access to the swap memory is extremely slow and hence the time could potentially increase disproportionately. On the other hand, we retrieve two failures due to lack of available memory. The previous issues could potentially be addressed by the use of iterative methods. Such methods, embedded in the IP-PMM framework, could significantly relax the memory as well as the processing requirements, at the expense of providing inexact directions. Combining IP-PMM (which is stable and reliable) with such an inexact scheme (which could accelerate the IPM

Table 3.4: Large-scale problems

| Name | Collection | $\boldsymbol{n n z}(\boldsymbol{A})$ | time (s) | Status |
| ---: | ---: | ---: | ---: | ---: |
| fome21 | Mittelmann | 751,365 | 567.26 | opt |
| pds-10 | Mittelmann | 250,307 | 40.00 | opt |
| pds-30 | Mittelmann | 564,988 | 447.81 | opt |
| pds-60 | Mittelmann | $1,320,986$ | $2,265.39$ | opt |
| pds-100 | Mittelmann | $1,953,757$ | - | out of memory |
| rail582 | Mittelmann | 402,290 | 91.10 | opt |
| cre-b | Kennington | 347,742 | 24.48 | opt |
| cre-d | Kennington | 323,287 | 23.49 | opt |
| stocfor3 | Kennington | 72,721 | 4.56 | opt |
| ken-18 | Kennington | 667,569 | 77.94 | opt |
| osa-30 | Kennington | 604,488 | 1723.96 | opt |
| nug-20 | QAPLIB | 304,800 | 386.12 | opt |
| nug-30 | QAPLIB | $1,567,800$ | - | out of memory |

iterations) should be a viable and competitive alternative and will be addressed in the following chapters.

### 3.6 Conclusions

In this chapter, we presented an algorithm suitable for solving convex quadratic programs. It arises from the combination of an infeasible interior point method with the proximal method of multipliers (IP-PMM). The method is interpreted as a primal-dual regularized IPM, and we prove that it is guaranteed to converge in a polynomial number of iterations, under standard assumptions. As the algorithm relies only on one penalty parameter, we use the well-known theory of IPMs to tune it. In particular, we treat this penalty as a barrier parameter, and hence the method is well-behaved independently of the problem under consideration. Additionally, we derive a necessary condition for infeasibility and use it to construct an infeasibility detection mechanism. The algorithm is implemented, and the reliability of the method is demonstrated. At the expense of some extra iterations, regularization improves the numerical properties of the interior point iterations. The increase in the number of iterations is benign, since factorization efficiency as well as stability is gained. Not only the method remains efficient, but it outperforms a similar non-regularized IPM scheme.

We observe the limitations of the current approach, due to the cost of factorization, and it is expected that embedding iterative methods in the current scheme might further improve the scalability of the algorithm at the expense of inexact directions. Since the reliability of IP-PMM is demonstrated, it only seems reasonable to allow for approximate Newton directions and still expect fast convergence. Hence, in the following chapters we will extend the theory and the implementation, in order to accommodate the use of iterative methods for solving the associated Newton systems.

## Algorithm PC Predictor-Corrector method

Compute the predictor:

$$
\left[\begin{array}{cc}
-\left(Q+\Theta_{k}^{-1}+\rho_{k} I\right) & A^{\top}  \tag{3.34}\\
A & \delta_{k} I
\end{array}\right]\left[\begin{array}{c}
\Delta_{p} x \\
\Delta_{p} y
\end{array}\right]=\left[\begin{array}{c}
c+Q x_{k}-A^{\top} y_{k}-\rho_{k}\left(x_{k}-\zeta_{k}\right) \\
b-A x_{k}-\delta_{k}\left(y_{k}-\eta_{k}\right)
\end{array}\right],
$$

where $\left(\Theta_{k}^{\mathcal{I}}\right)^{-1}=\left(X_{k}^{\mathcal{I}}\right)^{-1}\left(Z_{k}^{\mathcal{I}}\right),\left(\Theta_{k}^{\mathcal{F}}\right)^{-1}=0_{|\mathcal{F}|,|\mathcal{F}|}$.
Retrieve $\Delta_{p} z$ :

$$
\Delta_{p} z^{I}=\left(-z_{k}^{\mathcal{I}}\right)^{I}-\left(X_{k}^{\mathcal{I}}\right)^{-1}\left(Z_{k}^{\mathcal{I}} \Delta_{p} x^{\mathcal{L}}\right), \quad \Delta_{p} z^{\mathcal{F}}=0_{|\mathcal{F}|} .
$$

Compute the step in the non-negativity orthant:

$$
\alpha_{x}^{\max }=\min _{\left(\Delta_{p} x_{k}^{\mathcal{I}(i)}<0\right)}\left\{1,-\frac{x^{\mathcal{I}(i)}}{\Delta_{p} x^{\mathcal{I}(i)}}\right\}, \quad \alpha_{z}^{\max }=\min _{\left(\Delta_{p} z^{\mathcal{I}(i)}<0\right)}\left\{1,-\frac{z_{k}^{\mathcal{I}(i)}}{\Delta_{p} z^{\mathcal{I}(i)}}\right\},
$$

for $i=1, \cdots,|\mathcal{I}|$, and set:

$$
\alpha_{x}=\nu \alpha_{x}^{\max }, \quad \alpha_{z}=\nu \alpha_{z}^{\max },
$$

with $\nu=0.995 \quad \triangleright$ avoid going too close to the boundary.
Compute a centrality measure:

$$
g_{\alpha}=\left(x^{\mathcal{I}}+\alpha_{x} \Delta_{p} x^{\mathcal{I}}\right)^{\top}\left(z^{\mathcal{I}}+\alpha_{z} \Delta_{p} z^{\mathcal{I}}\right) .
$$

Set: $\mu=\left(\frac{g_{\alpha}}{\left(x_{k}^{T}\right)^{T} z_{k}^{\tau}}\right)^{2} \frac{g_{\alpha}}{|\mathcal{T}|}$
Compute the corrector:

$$
\left[\begin{array}{cc}
-\left(Q+\Theta_{k}^{-1}+\rho_{k} I\right) & A^{\top}  \tag{3.35}\\
A & \delta_{k} I
\end{array}\right]\left[\begin{array}{c}
\Delta_{c} x \\
\Delta_{c} y
\end{array}\right]=\left[\begin{array}{c}
-d_{k} \\
0_{m}
\end{array}\right],
$$

with $d_{k}^{\mathcal{I}}=\mu\left(X_{k}^{\mathcal{I}}\right)^{-1} e_{|\mathcal{I}|}-\left(X_{k}^{\mathcal{I}}\right)^{-1} \Delta_{p} X^{\mathcal{I}} \Delta_{p} z^{\mathcal{I}}, d_{k}^{\mathcal{F}}=0_{|\mathcal{F}|}, \Delta_{p} X=\operatorname{diag}\left(\Delta_{p} x\right)$.
Retrieve $\Delta_{c} z$ :

$$
\Delta_{c} z^{\mathcal{I}}=d_{k}^{\mathcal{I}}-\left(X_{k}^{\mathcal{I}}\right)^{-1}\left(Z_{k}^{\mathcal{I}} \Delta_{c} x^{\mathcal{I}}\right), \quad \Delta_{c} z^{\mathcal{F}}=0_{|\mathcal{F}|} .
$$

$$
(\Delta x, \Delta y, \Delta z)=\left(\Delta_{p} x+\Delta_{c} x, \Delta_{p} y+\Delta_{c} y, \Delta_{p} z+\Delta_{c} z\right) .
$$

Compute the step in the non-negativity orthant:

$$
\alpha_{x}^{\max }=\min _{\Delta x^{\mathcal{I}(i)}<0}\left\{1,-\frac{x_{k}^{\mathcal{I}(i)}}{\Delta x^{\mathcal{I}(i)}}\right\}, \quad \alpha_{z}^{\max }=\min _{\Delta z^{\mathcal{I}(i)}<0}\left\{1,-\frac{z_{k}^{\mathcal{I}(i)}}{\Delta z^{\mathcal{I}(i)}}\right\},
$$

and set:

$$
\alpha_{x}=\tau \alpha_{x}^{\max }, \quad \alpha_{z}=\tau \alpha_{z}^{\max } .
$$

Update:

$$
\left(x_{k+1}, y_{k+1}, z_{k+1}\right)=\left(x_{k}+\alpha_{x} \Delta x, y_{k}+\alpha_{z} \Delta y, z_{k}+\alpha_{z} \Delta z\right) .
$$

```
Algorithm PEU Penalty and Estimate Updates
    \(r=\frac{\left|\mu_{k}-\mu_{k+1}\right|}{\mu_{k}}\) (rate of decrease of \(\mu\) ).
    if \(\left(\left\|A x_{k+1}-b\right\|_{2} \leq 0.95 \cdot\left\|A x_{k}-b\right\|_{2}\right)\) then
        \(\eta_{k+1}=y_{k+1}\),
        \(\delta_{k+1}=(1-r) \cdot \delta_{k}\).
    else
        \(\eta_{k+1}=\eta_{k}\),
        \(\delta_{k+1}=\left(1-\frac{1}{3} r\right) \cdot \delta_{k}, \quad \triangleright\) less aggressive is in this case.
    end if
    \(\delta_{k+1}=\max \left\{\delta_{k+1}\right.\), reg \(\left._{t h r}\right\}, \triangleright\) for numerical stability (ensure quasi-definiteness).
    if \(\left(\left\|c+Q x_{k+1}-A^{\top} y_{k+1}-z_{k+1}\right\|_{2} \leq 0.95 \cdot\left\|c+Q x_{k}-A^{\top} y_{k}-z_{k}\right\|_{2}\right)\) then
        \(\zeta_{k+1}=x_{k+1}\).
        \(\rho_{k+1}=(1-r) \cdot \rho_{k}\).
    else
        \(\zeta_{k+1}=\zeta_{k}\).
        \(\rho_{k+1}=\left(1-\frac{1}{3} r\right) \cdot \rho_{k}\).
    end if
    \(\rho_{k+1}=\max \left\{\rho_{k+1}, \operatorname{reg}_{t h r}\right\}\).
```

```
Algorithm TC Termination Criteria
    Input: tol, \(k_{\mathrm{IP}}, k_{\mathrm{PMM}}, \mathrm{IP}_{\text {maxit }}, \mathrm{PMM}_{\text {maxit }}\).
    if \(\left(\left(\frac{\left\|c-A^{\top} y_{k}+Q x_{k}-z_{k}\right\|_{2}}{\max \left\{\|c\|_{2}, 1\right\}} \leq\right.\right.\) tol \() \wedge\left(\frac{\left\|b-A x_{k}\right\|_{2}}{\max \left\{\|b\|_{2}, 1\right\}} \leq\right.\) tol \() \wedge\left(\mu_{k} \leq\right.\) tol \(\left.)\right)\) then
        return Solution \(\left(x_{k}, y_{k}, z_{k}\right)\).
    else if \(\left(\left(\left\|c+Q x_{k}-A^{\top} y_{k}-z_{k}+\rho_{k}\left(x_{k}-\zeta_{k}\right)\right\|_{2} \leq \mathrm{tol}\right) \wedge\left(\left\|x_{k}-\zeta_{k}\right\|_{2}>10^{10}\right)\right)\)
    then
        if \(\left(k_{\text {PMM }} \geq \mathrm{PMM}_{\text {maxit }}\right)\) then \(\quad \triangleright\) PMM sub-problem not updated for many
    iterations
        Declare Infeasibility.
        end if
    else if \(\left(\left(\left\|b-A x_{k}-\delta_{k}\left(y_{k}-\lambda_{k}\right)\right\|_{2} \leq\right.\right.\) tol \(\left.) \wedge\left(\left\|y_{k}-\eta_{k}\right\|_{2}>10^{10}\right)\right)\) then
        if \(\left(k_{\text {PMM }} \geq \mathrm{PMM}_{\text {maxit }}\right)\) then
            Declare Infeasibility.
        end if
    else if \(\left(k_{\text {IP }} \geq \mathrm{IP}_{\text {maxit }}\right)\) then \(\quad \triangleright\) Maximum IPM iterations reached.
        No Convergence.
    end if
```


## Chapter 4

## An IP-PMM for Linear SDP

### 4.1 Introduction

In this chapter we extend IP-PMM for solving linear positive semi-definite programming (SDP) problems. As in Chapter 3, we apply some iterations of an IPM to each sub-problem of the PMM until a satisfactory solution is found. We then update the PMM parameters, form a new IPM neighbourhood, and repeat this process. Given this framework, we prove polynomial complexity of the algorithm, under mild assumptions, and without requiring exact computations for the Newton directions. We furthermore provide a necessary condition for lack of strong duality, which can be used as a basis for constructing detection mechanisms for identifying pathological cases within IP-PMM.

Positive semi-definite programming problems have attracted a lot of attention in the literature for more more than two decades, and have been used to model a plethora of different problems arising from control theory [11, Chapter 14], power systems [105], stochastic optimization [20], truss optimization [177], and many other application areas (e.g. see [11, 170]). More recently, SDP has been extensively used for building tight convex relaxations of NP-hard combinatorial optimization problems (see [11, Chapter 12], and the references therein).

As a result of the seemingly unlimited applicability of SDP, numerous contributions have been made to optimization techniques suitable for solving such problems. The most remarkable milestone was achieved by Nesterov and Nemirovski [128], who designed a polynomially convergent interior point method for the class of SDP problems. This led to the development of numerous successful IPM variants for SDP; some of theoretical (e.g. [121, 188, 190]) and others of practical nature (e.g. [19, 18, 123]). While IPMs enjoy fast convergence, in theory and in practice, each IPM iteration requires the solution of a very large-scale linear system, even for small-scale SDP problems. What is worse, such linear systems are inherently ill-conditioned. A viable and successful alternative to IPMs for SDP problems (e.g. see [189]), which circumvents the issue of ill-conditioning without significantly compromising convergence speed, is based on the augmented Lagrangian method, which, as already mentioned, can be seen as the dual application of the proximal point method (see [146]). The issue with ALMs is that, unlike IPMs, a consistent strategy for tuning the algorithm parameters is not
known. Furthermore, polynomial complexity is lost, and is replaced with merely a finite termination. An IPM scheme combined with the proximal method of multipliers for solving SDP problems was proposed in [54], and was interpreted as a primal-dual regularized IPM. The authors established global convergence, and numerically demonstrated the efficiency of the method. However, the latter is not guaranteed to converge in a polynomial number of iterations, or even in a finite number of steps. Finally, viable alternatives based on proximal splitting methods have been studied in $[97,162]$. Such methods are very efficient and require significantly less computations and memory per iteration, as compared to IPM or ALM. However, as first-order schemes, their convergence to high accuracy might be slow. Hence, such approaches are suitable for finding approximate solutions with low-accuracy.

In this chapter, we are extending the interior point-proximal method of multipliers presented in Chapter 3. In particular, the algorithm in Chapter 3 was developed for convex quadratic programming problems and assumed that the resulting linear systems are solved exactly. Under this framework, it was proved that IPPMM converges in a polynomial number of iterations, under mild assumptions, and an infeasibility detection mechanism was established. An important feature of this method is that it provides a reliable tuning for the penalty parameters of the PMM; indeed, the reliability of the algorithm is established numerically in Chapter 3 as well as in [23]. In particular, the IP-PMM proposed in [23] uses preconditioned iterative methods for the solution of the resulting linear systems, and is very robust despite the inexact computations. In what follows, we develop an IP-PMM for linear SDP problems, which furthermore allows for inexactness in the solution of the linear systems that have to be solved at every iteration. We show that the method converges polynomially under standard assumptions. Subsequently, we provide a necessary condition for lack of strong duality, which can serve as a basis for constructing implementable detection mechanisms for pathological cases (following the developments in Chapter 3). We note that the chapter is focused on the theoretical aspects of the method, and an efficient, scalable, and reliable implementation would require a separate study.

The rest of this chapter is organized as follows. In Section 4.2, we provide some preliminary background and introduce our notation. Then, in Section 4.3, we provide the algorithmic framework of the method. In Section 4.4, we prove polynomial complexity of the algorithm, and establish its global convergence. Subsequently, in Section 4.5, a necessary condition for lack of strong duality is derived, and we discuss how it can be used to construct an implementable detection mechanism for pathological cases. Finally, we derive some conclusions in Section 4.6.

### 4.2 Preliminaries and notation

### 4.2.1 Primal-dual pair of SDP problems

Let the vector space $\mathcal{S}^{n}:=\left\{B \in \mathbb{R}^{n \times n}: B=B^{\top}\right\}$ be given, endowed with the inner product $\langle A, B\rangle=\operatorname{Tr}(A B)$, where $\operatorname{Tr}(\cdot)$ denotes the trace of a matrix. In this
chapter, we consider the following primal-dual pair of linear positive semi-definite programming problems, in the standard form:

$$
\begin{gather*}
\min _{X \in \mathcal{S}^{n}}\langle C, X\rangle, \text { s.t. } \mathcal{A} X=b, X \in \mathcal{S}_{+}^{n},  \tag{SDP}\\
\max _{y \in \mathbb{R}^{m}, Z \in \mathcal{S}^{n}} b^{\top} y, \quad \text { s.t. } \mathcal{A}^{*} y+Z=C, Z \in \mathcal{S}_{+}^{n}, \tag{SDD}
\end{gather*}
$$

where $\mathcal{S}_{+}^{n}:=\left\{B \in \mathcal{S}^{n}: B \succeq 0_{n, n}\right\}, C, X, Z \in \mathcal{S}^{n}, b, y \in \mathbb{R}^{m}, \mathcal{A}$ is a linear operator on $\mathcal{S}^{n}$, and $\mathcal{A}^{*}$ is the adjoint of $\mathcal{A}$. We note that the norm induced by the inner product $\langle A, B\rangle=\operatorname{Tr}(A B)$ is in fact the Frobenius norm, denoted by $\|\cdot\|_{F}$. Furthermore, the adjoint $\mathcal{A}^{*}: \mathbb{R}^{m} \mapsto \mathcal{S}^{n}$ is such that $y^{\top} \mathcal{A} X=\left\langle\mathcal{A}^{*} y, X\right\rangle, \quad \forall y \in$ $\mathbb{R}^{m}, \forall X \in \mathcal{S}^{n}$.

For the rest of this chapter, except for Section 4.5, we will assume that the linear operator $\mathcal{A}$ is onto and that problems (SDP) and (SDD) are both strictly feasible (that is, Slater's constraint qualification holds for both problems). It is well-known that under the previous assumptions, the primal-dual pair (SDP)(SDD) is guaranteed to have an optimal solution for which strong duality holds (see [128]). Such a solution can be found by solving the Karush-Kuhn-Tucker (KKT) optimality conditions for (SDP)-(SDD), which read as follows:

$$
\left[\begin{array}{c}
\mathcal{A}^{*} y+Z-C  \tag{4.1}\\
\mathcal{A} X-b \\
X Z
\end{array}\right]=\left[\begin{array}{c}
0_{n, n} \\
0_{m} \\
0_{n, n}
\end{array}\right], \quad X, Z \in \mathcal{S}_{+}^{n} .
$$

### 4.2.2 A proximal method of multipliers

Let us derive a proximal method of multipliers (see [147]) for the pair (SDP)(SDD). Given arbitrary starting point $\left(X_{0}, y_{0}\right) \in \mathcal{S}_{+}^{n} \times \mathbb{R}^{m}$, the PMM can be summarized by the following iteration:

$$
\begin{align*}
X_{k+1} & =\underset{X \in \mathcal{S}_{+}^{n}}{\arg \min }\left\{\langle C, X\rangle-y_{k}^{\top}(\mathcal{A} X-b)+\frac{\mu_{k}}{2}\left\|X-X_{k}\right\|_{F}^{2}+\frac{1}{2 \mu_{k}}\|\mathcal{A} X-b\|_{2}^{2}\right\}, \\
y_{k+1} & =y_{k}-\frac{1}{\mu_{k}}\left(\mathcal{A} X_{k+1}-b\right), \tag{4.2}
\end{align*}
$$

where $\mu_{k}$ is a positive penalty parameter. The previous iteration admits a unique solution, for all $k$.

As in Section 1.1.2, we can write (4.2) equivalently by making use of the maximal monotone operator $T_{\mathcal{L}}: \mathbb{R}^{m} \times \mathcal{S}^{n} \rightrightarrows \mathbb{R}^{m} \times \mathcal{S}^{n}$ (see [146, 147]), whose graph is defined as:

$$
\begin{equation*}
T_{\mathcal{L}}(X, y):=\left\{(V, u): V \in C-\mathcal{A}^{*} y+\partial \delta_{S_{+}^{n}}(X), u=\mathcal{A} X-b\right\}, \tag{4.3}
\end{equation*}
$$

where $\delta_{S_{+}^{n}}(\cdot)$ is an indicator function defined as:

$$
\delta_{S_{+}^{n}}(X):= \begin{cases}0, & \text { if } X \in \mathcal{S}_{+}^{n},  \tag{4.4}\\ \infty, & \text { otherwise },\end{cases}
$$

and $\partial(\cdot)$ denotes the sub-differential of a function, hence (from [148, Corollary 23.5.4]):

$$
Z \in \partial \delta_{S_{+}^{n}}(X) \Leftrightarrow-Z \in \mathcal{S}_{+}^{n},\langle X, Z\rangle=0
$$

By convention, we have that $\partial \delta_{\mathcal{S}_{+}^{n}\left(X^{*}\right)}=\emptyset$ if $X^{*} \notin \mathcal{S}_{+}^{n}$. Given a bounded pair $\left(X^{*}, y^{*}\right)$ such that $\left(0_{n, n}, 0_{m}\right) \in T_{\mathcal{L}}\left(X^{*}, y^{*}\right)$, we can retrieve a matrix $Z^{*} \in$ $\partial \delta_{S_{+}^{n}}\left(X^{*}\right)$, using which $\left(X^{*}, y^{*},-Z^{*}\right)$ is an optimal solution for (SDP)-(SDD). By defining the proximal operator:

$$
\begin{equation*}
\mathcal{P}_{k}:=\left(I_{n+m}+\frac{1}{\mu_{k}} T_{\mathcal{L}}\right)^{-1} \tag{4.5}
\end{equation*}
$$

where $I_{n+m}$ is the identity operator of size $n+m$, and describes the direct sum of the idenity operators of $\mathcal{S}_{n}$ and $\mathbb{R}^{m}$, we can express (4.2) as:

$$
\begin{equation*}
\left(X_{k+1}, y_{k+1}\right)=\mathcal{P}_{k}\left(X_{k}, y_{k}\right) \tag{4.6}
\end{equation*}
$$

and it can be shown that $\mathcal{P}_{k}$ is single valued and firmly non-expansive (see [147]).

### 4.2.3 An infeasible interior point method

In what follows we present a basic infeasible IPM suitable for solving the primal-dual pair (SDP)-(SDD). Such methods handle the conic constraints by introducing a suitable logarithmic barrier in the objective (for an extensive study of logarithmic barriers, the reader is referred to [128]). At each iteration, we choose a barrier parameter $\mu>0$ and form the logarithmic barrier primal-dual pair:

$$
\begin{align*}
& \min _{X \in \mathcal{S}^{n}}\langle C, X\rangle-\mu \ln (\operatorname{det}(X)), \text { s.t. } \mathcal{A} X=b,  \tag{4.7}\\
& \max _{y \in \mathbb{R}^{m},}, Z \in \mathcal{S}^{n}  \tag{4.8}\\
& b^{\top} y+\mu \ln (\operatorname{det}(Z)), \text { s.t. } \mathcal{A}^{*} y+Z=C .
\end{align*}
$$

The first-order (barrier) optimality conditions of (4.7)-(4.8) read as follows:

$$
\left[\begin{array}{c}
\mathcal{A}^{*} y+Z-C  \tag{4.9}\\
\mathcal{A} X-b \\
X Z-\mu I_{n}
\end{array}\right]=\left[\begin{array}{c}
0_{n, n} \\
0_{m} \\
0_{n, n}
\end{array}\right], \quad X, Z \in \mathcal{S}_{++}^{n},
$$

where $\mathcal{S}_{++}^{n}:=\left\{B \in \mathcal{S}^{n}: B \succ 0_{n, n}\right\}$. For every chosen value of $\mu$, we want to approximately solve the following non-linear system of equations:

$$
F_{\tau, \mu}^{I P M}(w):=\left[\begin{array}{c}
\mathcal{A}^{*} y+Z-C \\
\mathcal{A} X-b \\
X Z-\tau \mu I_{n}
\end{array}\right]=\left[\begin{array}{c}
0_{n, n} \\
0_{m} \\
0_{n, n}
\end{array}\right]=: \mathbf{0},
$$

where, with a slight abuse of notation, we set $w=(X, y, Z)$.
In IPM literature it is common to apply Newton method to solve approximately the system of non-linear equations $F_{\tau, \mu}^{I P M}(w)=\mathbf{0}$. As already mentioned in the introduction, Newton method is favored for systems of this form due to the self-concordance of the logarithmic barrier (see [128]). However, a well-known issue in the literature is that the matrix $X Z$ is not necessarily symmetric. A common approach to tackle this issue is to employ a symmetrization operator $H_{P}: \mathbb{R}^{n \times n} \mapsto \mathcal{S}^{n}$, such that $H_{P}(X Z)=\mu I_{n}$ if and only if $X Z=\mu I_{n}$, given that $X, Z \in \mathcal{S}_{+}^{n}$. Following Zhang ([188]), we employ the following operator: $H_{P}: \mathbb{R}^{n \times n} \mapsto \mathcal{S}^{n}:$

$$
\begin{equation*}
H_{P}(B):=\frac{1}{2}\left(P B P^{-1}+\left(P B P^{-1}\right)^{\top}\right) \tag{4.10}
\end{equation*}
$$

where $P$ is a non-singular matrix. It can be shown that the central path (a key notion used in IPMs-see [128]) can be equivalently defined as $H_{P}(X Z)=\mu I_{n}$, for any non-singular $P$. In this chapter, we will make use of the choice $P_{k}=Z_{k}^{-\frac{1}{2}}$ (the notation should not be confused with the proximal operator in (1.10)). For a plethora of alternative choices, the reader is referred to [167]. We should note that the analysis in this chapter can be tailored to different symmetrization strategies, and this choice is made for simplicity of exposition.

At the beginning of the $k$-th iteration, we have $w_{k}=\left(X_{k}, y_{k}, Z_{k}\right)$ and $\mu_{k}$ available. The latter is defined as $\mu_{k}=\frac{\left\langle X_{k}, Z_{k}\right\rangle}{n}$. By substituting the symmetrized complementarity in the last block equation and applying Newton method, we obtain the following system of equations:

$$
\left[\begin{array}{ccc}
0_{n, n} & \mathcal{A}^{*} & I_{n}  \tag{4.11}\\
\mathcal{A} & 0_{m, m} & 0_{m, n} \\
\mathcal{E}_{k} & 0_{n, m} & \mathcal{F}_{k}
\end{array}\right]\left[\begin{array}{c}
\Delta X \\
\Delta y \\
\Delta Z
\end{array}\right]=\left[\begin{array}{c}
C-\mathcal{A}^{*} y-Z_{k} \\
b-\mathcal{A} X_{k} \\
\mu I_{n}-H_{P_{k}}\left(X_{k} Z_{k}\right)
\end{array}\right],
$$

where $\mathcal{E}_{k}:=\nabla_{X} H_{P_{k}}\left(X_{k} Z_{k}\right)$, and $\mathcal{F}_{k}:=\nabla_{Z} H_{P_{k}}\left(X_{k} Z_{k}\right)$.

### 4.2.4 Vectorized format

In what follows we vectorize the associated operators, in order to work with matrices. In particular, given any matrix $B \in \mathbb{R}^{m \times n}$, we denote its vectorized form as $\boldsymbol{B}$, which is a vector of size $m n$, obtained by stacking the columns of $B$, from the first to the last. For the rest of this manuscript, any boldface letter (other than $\boldsymbol{O}(\cdot), \boldsymbol{\Omega}(\cdot), \boldsymbol{\Theta}(\cdot)$ which are reserved for the complexity notation) denotes a vectorized matrix. Furthermore, if $\mathcal{A}: \mathcal{S}^{n} \mapsto \mathbb{R}^{m}$ is a linear operator, we can define it component-wise as $(\mathcal{A} X)_{i}:=\left\langle A_{i}, X\right\rangle$, for $i=1, \ldots, m$, and any $X \in \mathcal{S}^{n}$, where $A_{i} \in \mathcal{S}^{n}$. Furthermore, the adjoint of this operator, that is $\mathcal{A}^{*}: \mathbb{R}^{m} \mapsto \mathcal{S}^{n}$
is defined as $\mathcal{A}^{*} y:=\sum_{i=1}^{m} y_{i} A_{i}$, for all $y \in \mathbb{R}^{m}$. Using this notation, we can equivalently write (SDP)-(SDD) in the following form:

$$
\begin{align*}
\min _{X \in \mathcal{S}^{n}}\langle C, X\rangle, \quad \text { s.t. }\left\langle A_{i}, X\right\rangle=b_{i}, \quad i=1, \ldots, m, \quad X \in \mathcal{S}_{+}^{n},  \tag{4.12}\\
\max _{y \in \mathbb{R}^{m}, Z \in \mathcal{S}^{n}} b^{\top} y, \quad \text { s.t. } \sum_{i=1}^{m} y_{i} A_{i}+Z=C, \quad Z \in \mathcal{S}_{+}^{n} . \tag{4.13}
\end{align*}
$$

The first-order optimality conditions can be re-written as:

$$
\left[\begin{array}{c}
A^{\top} y+\boldsymbol{Z}-\boldsymbol{C} \\
A \boldsymbol{X}-b \\
X Z
\end{array}\right]=\left[\begin{array}{c}
0_{n^{2}} \\
0_{m} \\
0_{n^{2}}
\end{array}\right], \quad X, Z \in \mathcal{S}_{+}^{n}
$$

where $A^{\top}=\left[\begin{array}{llll}\boldsymbol{A}_{1} & \boldsymbol{A}_{2} & \cdots & \boldsymbol{A}_{m}\end{array}\right]$.

### 4.3 An interior point-proximal method of multipliers for SDP

In this section we present an inexact extension of IP-PMM presented in Chapter 3, suitable for solving problems of the form of (SDP)-(SDD). Assume that we have available an estimate $\eta_{k}$ for a Lagrange multiplier vector at iteration $k$. Similarly, denote by $\Xi_{k} \in \mathcal{S}_{+}^{n}$ an estimate of a primal solution. During the $k$-th iteration of the PMM, applied to (SDP), the following proximal penalty function has to be minimized:

$$
\begin{equation*}
\mathcal{L}_{\mu_{k}}^{P M M}\left(X ; \Xi_{k}, \eta_{k}\right):=\langle C, X\rangle-\eta_{k}^{\top}(\mathcal{A} X-b)+\frac{1}{2 \mu_{k}}\|\mathcal{A} X-b\|_{2}^{2}+\frac{\mu_{k}}{2}\left\|X-\Xi_{k}\right\|_{F}^{2}, \tag{4.14}
\end{equation*}
$$

with $\left\{\mu_{k}\right\}$ some non-increasing sequence of positive penalty parameters. Notice that this is equivalent to the iteration (4.2). We approximately minimize (4.14) by applying one (or a few) iterations of the previously presented infeasible IPM. We alter (4.14) by adding a logarithmic barrier:

$$
\begin{equation*}
\mathcal{L}_{\mu_{k}}^{I P-P M M}\left(X ; \Xi_{k}, \eta_{k}\right):=\mathcal{L}_{\mu_{k}}^{P M M}\left(X ; \Xi_{k}, \eta_{k}\right)-\mu_{k} \log (\operatorname{det}(X)), \tag{4.15}
\end{equation*}
$$

and we treat $\mu_{k}$ as the barrier parameter. We form the optimality conditions of this sub-problem as:

$$
C-\mathcal{A}^{*} \eta_{k}+\frac{1}{\mu_{k}} \mathcal{A}^{*}(\mathcal{A} X-b)+\mu_{k}\left(X-\Xi_{k}\right)-\mu_{k} X^{-1}=0_{n, n}
$$

Introducing the variables $y=\eta_{k}-\frac{1}{\mu_{k}}(\mathcal{A} X-b)$ and $Z=\mu_{k} X^{-1}$, yields:

$$
\left[\begin{array}{c}
C-\mathcal{A}^{*} y-Z+\mu_{k}\left(X-\Xi_{k}\right)  \tag{4.16}\\
\mathcal{A} X+\mu_{k}\left(y-\eta_{k}\right)-b \\
X Z-\mu_{k} I_{n}
\end{array}\right]=\left[\begin{array}{c}
0_{n, n} \\
0_{m} \\
0_{n, n}
\end{array}\right] \Leftrightarrow\left[\begin{array}{c}
\boldsymbol{C}-A^{\top} y-\boldsymbol{Z}+\mu_{k}\left(\boldsymbol{X}-\boldsymbol{\Xi}_{k}\right) \\
A \boldsymbol{X}+\mu_{k}\left(y-\eta_{k}\right)-b \\
H_{P_{k}}(X Z)-\mu_{k} I_{n}
\end{array}\right]=\left[\begin{array}{c}
0_{n^{2}} \\
0_{m} \\
0_{n^{2}}
\end{array}\right],
$$

where the second system is obtained by introducing the symmetrization in (4.10), and by vectorizing the associated matrices and operators.

Given an arbitrary vector $b \in \mathbb{R}^{m}$, and matrix $C \in \mathbb{R}^{n \times n}$, we define the semi-norm (extending that in [121, Section 4] to the SDP case):

$$
\|(b, \boldsymbol{C})\|_{\mathcal{S}}:=\min _{X, y, Z}\left\{\|(\boldsymbol{X}, \boldsymbol{Z})\|_{2}: \begin{array}{c}
A \boldsymbol{X}=b,  \tag{4.17}\\
A^{\top} y+\boldsymbol{Z}=\boldsymbol{C}
\end{array}\right\} .
$$

Starting point Similar to Section 3.2, we set $\left(X_{0}, Z_{0}\right)=\rho\left(I_{n}, I_{n}\right)$, for some $\rho>0$. We also set $y_{0}$ to some arbitrary value (e.g. $y_{0}=0_{m}$ ), and $\mu_{0}=\frac{\left\langle X_{0}, Z_{0}\right\rangle}{n}$. Using the aforementioned triple, we have:

$$
\begin{equation*}
A \boldsymbol{X}_{0}=b+\bar{b}, A^{\top} y_{0}+\boldsymbol{Z}_{0}=\boldsymbol{C}+\overline{\boldsymbol{C}}, \Xi_{0}=X_{0}, \eta_{0}=y_{0} . \tag{4.18}
\end{equation*}
$$

for some $\bar{b} \in \mathbb{R}^{m}$, and $\bar{C} \in \mathcal{S}^{n}$.
Neighbourhood The neighbourhood is a direct extension of the one employed in Section 3.2. Given (4.18), some $\mu_{k}, \eta_{k}$, and $\Xi_{k}$, we define the regularized set of centers:

$$
\begin{gathered}
\mathscr{P}_{\mu_{k}}\left(\Xi_{k}, \eta_{k}\right):=\left\{(X, y, Z) \in \mathscr{C}_{\mu_{k}}\left(\Xi_{k}, \eta_{k}\right): X \in \mathcal{S}_{++}^{n}, Z \in \mathcal{S}_{++}^{n}, X Z=\mu_{k} I_{n}\right\}, \\
\mathscr{C}_{\mu_{k}}\left(\Xi_{k}, \eta_{k}\right):=\left\{\begin{array}{c}
(X, y, Z): \quad A \boldsymbol{X}+\mu_{k}\left(y-\eta_{k}\right)=b+\frac{\mu_{k}}{\mu_{0}} \bar{b} \\
A^{\top} y+\boldsymbol{Z}-\mu_{k}\left(\boldsymbol{X}-\boldsymbol{\Xi}_{k}\right)=\boldsymbol{C}+\frac{\mu_{k}}{\mu_{0}} \overline{\boldsymbol{C}}
\end{array}\right\},
\end{gathered}
$$

where $\bar{b}, \bar{C}$ are as in (4.18).
We enlarge the previous set, by defining the following set:

$$
\tilde{\mathscr{C}}_{\mu_{k}}\left(\Xi_{k}, \eta_{k}\right):=\left\{\begin{array}{c}
A \boldsymbol{X}+\mu_{k}\left(y-\eta_{k}\right)=b+\frac{\mu_{k}}{\mu_{0}}\left(\bar{b}+\tilde{b}_{k}\right), \\
(X, y, Z): \\
A^{\top} y+\boldsymbol{Z}-\mu_{k}\left(\boldsymbol{X}-\boldsymbol{\Xi}_{k}\right)=\boldsymbol{C}+\frac{\mu_{k}}{\mu_{0}}\left(\overline{\boldsymbol{C}}+\tilde{\boldsymbol{C}}_{k}\right) \\
\left\|\left(\tilde{b}_{k}, \tilde{\boldsymbol{C}}_{k}\right)\right\|_{2} \leq K_{N},\left\|\left(\tilde{b}_{k}, \tilde{\boldsymbol{C}}_{k}\right)\right\|_{\mathcal{S}} \leq \gamma_{\mathcal{S}} \rho
\end{array}\right\},
$$

where $K_{N}>0$ is a constant, $\gamma_{\mathcal{S}} \in(0,1)$ and $\rho>0$ is as defined in the starting point. The vector $\tilde{b}_{k}$ and the matrix $\tilde{C}_{k}$ represent the current scaled (by $\frac{\mu_{0}}{\mu_{k}}$ ) infeasibility and will vary depending on the iteration $k$. We can now define a family of neighbourhoods:

$$
\mathscr{N}_{\mu_{k}}\left(\Xi_{k}, \eta_{k}\right):=\left\{\begin{array}{cc}
(X, y, Z) \in \tilde{\mathscr{C}}_{\mu_{k}}\left(\Xi_{k}, \eta_{k}\right): & X \in \mathcal{S}_{++}^{n}, Z \in \mathcal{S}_{+++}^{n},  \tag{4.19}\\
& \left\|H_{P}(X Z)-\mu I_{n}\right\|_{F} \leq \gamma_{\mu} \mu_{k}
\end{array}\right\},
$$

where $\gamma_{\mu} \in(0,1)$ is a constant restricting the symmetrized complementarity products. Obviously, the starting point defined in (4.18) belongs to the neighbourhood
$\mathscr{N}_{\mu_{0}}\left(\Xi_{0}, \eta_{0}\right)$, with $\left(\tilde{b}_{0}, \tilde{\boldsymbol{C}}_{0}\right)=\left(0_{m}, 0_{n^{2}}\right)$.
Newton system As discussed earlier, we employ the Newton method for approximately solving a perturbed form of system (4.16), for all $k$. As in Section 3.2 , we perturb (4.16) in order to take into consideration the target reduction of the barrier parameter $\mu_{k}$ (by introducing the centering parameter $\tau_{k}$ ), as well as to incorporate the initial infeasibility, given our starting point in (4.18). In particular, we would like to solve the following system:

$$
\left[\begin{array}{ccc}
-\left(\mu_{k} I_{n}\right) & \mathcal{A}^{*} & I_{n}  \tag{4.20}\\
\mathcal{A} & \mu_{k} I_{m} & 0_{n, m} \\
Z_{k} & 0_{n, m} & X_{k}
\end{array}\right]\left[\begin{array}{c}
\Delta X_{k} \\
\Delta y_{k} \\
\Delta Z_{k}
\end{array}\right]=\left[\begin{array}{c}
\left(C+\frac{\tau_{k} \mu_{k}}{\mu_{0}} \bar{C}\right)-\mathcal{A}^{*} y_{k}-Z_{k}+\tau_{k} \mu_{k}\left(X_{k}-\Xi_{k}\right) \\
-\mathcal{A} X_{k}-\tau_{k} \mu_{k}\left(y_{k}-\eta_{k}\right)+\left(b+\frac{\left.\tau_{k} \mu_{k} \bar{b}\right)}{\mu_{0}}\right. \\
-X_{k} Z_{k}+\tau_{k} \mu_{k} I_{n}
\end{array}\right],
$$

where $\bar{b}, \bar{C}$ are as in (4.18). We note that we could either first linearize the last block equation of (4.16) and then apply the symmetrization, defined in (4.10), or first apply the symmetrization directly to the last block equation of (4.16) and then linearize it. Both approaches are equivalent. Hence, following the former approach, we obtain the vectorized Newton system, that has to be solved at every iteration of IP-PMM:

$$
\begin{align*}
& {\left[\begin{array}{ccc}
-\left(\mu_{k} I_{n^{2}}\right) & A^{\top} & I_{n^{2}} \\
A & \mu_{k} I_{m} & 0_{m, n^{2}} \\
E_{k} & 0_{n^{2}, m} & F_{k}
\end{array}\right]\left[\begin{array}{c}
\boldsymbol{\Delta} \boldsymbol{X}_{k} \\
\Delta y_{k} \\
\boldsymbol{\Delta} \boldsymbol{Z}_{k}
\end{array}\right]} \\
& \quad=\left[\begin{array}{c}
\left(\boldsymbol{C}+\frac{\tau_{k} \mu_{k}}{\mu_{0}} \overline{\boldsymbol{C}}\right)-A^{\top} y_{k}-\boldsymbol{Z}_{k}+\tau_{k} \mu_{k}\left(\boldsymbol{X}_{k}-\mathbf{\Xi}_{k}\right) \\
-A \boldsymbol{X}_{k}-\tau_{k} \mu_{k}\left(y_{k}-\eta_{k}\right)+\left(b+\frac{\tau_{k} \mu_{k}}{\mu_{0}} \bar{b}\right) \\
- \\
-\left(Z_{k}^{\frac{1}{2}} \otimes Z_{k}^{\frac{1}{2}}\right) \boldsymbol{X}_{k}+\tau_{k} \mu_{k} \boldsymbol{I}_{n}
\end{array}\right]+\left[\begin{array}{c}
\mathbf{E}_{d, k} \\
\epsilon_{p, k} \\
\mathbf{E}_{\mu, k}
\end{array}\right], \tag{4.21}
\end{align*}
$$

where $E_{k}=\left(Z_{k}^{\frac{1}{2}} \otimes Z_{k}^{\frac{1}{2}}\right), F_{k}=\frac{1}{2}\left(Z_{k}^{\frac{1}{2}} X_{k} \otimes Z_{k}^{-\frac{1}{2}}+Z^{-\frac{1}{2}} \otimes Z_{k}^{\frac{1}{2}} X_{k}\right)$, and $\left(\mathrm{E}_{d, k}, \epsilon_{p, k}, \mathrm{E}_{\mu, k}\right)$ models potential errors, occurring by solving the symmetrized version of system (4.20) inexactly (e.g. by using a Krylov subspace method). In order to make sure that the computed direction is accurate enough, we impose the following accuracy conditions:

$$
\begin{equation*}
\left\|\mathbf{E}_{\mu, k}\right\|_{2}=0, \quad\left\|\left(\epsilon_{p, k}, \mathbf{E}_{d, k}\right)\right\|_{2} \leq \frac{\tau_{\min }}{4 \mu_{0}} K_{N} \mu_{k}, \quad\left\|\left(\epsilon_{p, k}, \mathbf{E}_{d, k}\right)\right\|_{\mathcal{S}} \leq \frac{\tau_{\min }}{4 \mu_{0}} \gamma_{\mathcal{S}} \rho \mu_{k} \tag{4.22}
\end{equation*}
$$

where $\tau_{\text {min }}$ is the minimum allowed value for $\tau_{k}, K_{N}, \gamma_{\mathcal{S}}$ are constants defined in (4.19), and $\rho$ is defined in the starting point in (4.18). Notice that the condition $\left\|\mathbf{E}_{\mu, k}\right\|_{2}=0$ is imposed without loss of generality, since it can be easily satisfied in practice. For more on this, see the discussions in [190, Section 3] and [88, Lemma 4.1]. Furthermore, as we will observe in Section 4.4, the bound on the error with respect to the semi-norm defined in (4.17) is required to ensure polynomial complexity of the method. While evaluating this semi-norm is not particularly practical (and is never performed in practice, e.g. see Section 3.5.1), it can be done in polynomial time (see [121, Section 4]), and hence does not affect the polynomial nature of the algorithm. The algorithmic scheme of the method is summarized in Algorithm IP-PMM-SDP.

```
Algorithm IP-PMM-SDP Interior Point-Proximal Method of Multipliers
Input: \(\mathcal{A}, b, C\), tol.
Parameters: \(0<\tau_{\min } \leq \tau_{\max } \leq 0.5, k^{\dagger}, \bar{K}, K_{N}>0,0<\gamma_{\mathcal{S}}<1,0<\gamma_{\mu}<1\).
Starting point: Set as in (4.18), \(\bar{k}=0\).
    for \((k=0,1,2, \cdots)\) do
        if \(\left(\left(\left\|A \boldsymbol{X}_{k}-b\right\|_{2}<\operatorname{tol}\right) \wedge\left(\left\|\boldsymbol{C}-A^{\top} y_{k}-\boldsymbol{Z}_{k}\right\|_{2}<\mathrm{tol}\right) \wedge\left(\frac{\left\langle X_{k}, Z_{k}\right\rangle}{n}<\mathrm{tol}\right)\right)\)
    then
            return \(\left(X_{k}, y_{k}, Z_{k}\right)\).
        else
            Choose \(\tau_{k} \in\left[\tau_{\min }, \tau_{\max }\right]\) and solve (4.21) so that (4.22) holds.
            Choose \(\alpha_{k}\), as the largest \(\alpha \in(0,1]\), s.t. \(\mu_{k}(\alpha) \leq(1-0.01 \alpha) \mu_{k}\), and:
\[
\begin{aligned}
& \left(X_{k}+\alpha_{k} \Delta X_{k}, y_{k}+\alpha_{k} \Delta y_{k}, Z_{k}+\alpha_{k} \Delta Z_{k}\right) \in \mathscr{N}_{\mu_{k}(\alpha)}\left(\Xi_{k}, \eta_{k}\right), \\
& \text { where, } \quad \mu_{k}(\alpha)=\frac{\left\langle X_{k}+\alpha_{k} \Delta X_{k}, Z_{k}+\alpha_{k} \Delta Z_{k}\right\rangle}{n} .
\end{aligned}
\]
Set \(\left(X_{k+1}, y_{k+1}, Z_{k+1}\right)=\left(X_{k}+\alpha_{k} \Delta X_{k}, y_{k}+\alpha_{k} \Delta y_{k}, Z_{k}+\alpha_{k} \Delta Z_{k}\right)\).
Set \(\mu_{k+1}=\frac{\left\langle X_{k+1}, Z_{k+1}\right\rangle}{n}\).
Let \(r_{p}=A \boldsymbol{X}_{k+1}-\left(b+\frac{\mu_{k+1}}{\mu_{0}} \bar{b}\right), \boldsymbol{R}_{d}=\left(\boldsymbol{C}+\frac{\mu_{k+1}}{\mu_{0}} \overline{\boldsymbol{C}}\right)-A^{\top} y_{k+1}-\boldsymbol{Z}_{k+1}\).
Set \(\left(\Xi_{k+1}, \eta_{k+1}\right)=\left(\Xi_{k}, \eta_{k}\right)\).
if \(\left(\left(\left\|\left(r_{p}, \boldsymbol{R}_{d}\right)\right\|_{2} \leq K_{N} \frac{\mu_{k+1}}{\mu_{0}}\right) \wedge\left(\left\|\left(r_{p}, \boldsymbol{R}_{d}\right)\right\|_{\mathcal{S}} \leq \gamma_{\mathcal{S}} \rho \frac{\mu_{k+1}}{\mu_{0}}\right)\right)\) then
if \(\left(\left\|\left(\boldsymbol{X}_{k+1}, y_{k+1}\right)\right\|_{2} \leq \bar{K} \sqrt{n}\right)\) then \(\left(\Xi_{k+1}, \eta_{k+1}\right)=\left(X_{k+1}, y_{k+1}\right)\).
else if \(\left(\bar{k} \leq k^{\dagger}\right)\) then \(\left(\Xi_{k+1}, \eta_{k+1}\right)=\left(X_{k+1}, y_{k+1}\right), \bar{k}=\bar{k}+1\).
end if
end if end if
end for
```


### 4.4 Convergence analysis

In this section we prove polynomial complexity of Algorithm IP-PMM-SDP, and establish its global convergence. The analysis is modeled after that in Chapter 3. We make use of the following two standard assumptions, generalizing those employed in Chapter 3 to the SDP case.

Assumption 3. The problems (SDP) and (SDD) are strictly feasible, that is, Slater's constraint qualification holds for both problems. Furthermore, there exists an optimal solution $\left(X^{*}, y^{*}, Z^{*}\right)$ and a constant $K_{*}>0$ independent of $n$ and $m$ such that $\left\|\left(\boldsymbol{X}^{*}, y^{*}, \boldsymbol{Z}^{*}\right)\right\|_{2} \leq K_{*} \sqrt{n}$.

Assumption 4. The vectorized constraint matrix $A$ of (SDP) has full row rank, that is $\operatorname{rank}(A)=m$. Moreover, there exist constants $K_{A, 1}>0, K_{A, 2}>0$,
$K_{r, 1}>0$, and $K_{r, 2}>0$, independent of $n$ and $m$, such that:

$$
\sigma_{\min }(A) \geq K_{A, 1}, \quad \sigma_{\max }(A) \leq K_{A, 2}, \quad\|b\|_{\infty} \leq K_{r, 1}, \quad\|\boldsymbol{C}\|_{2} \leq K_{r, 2} \sqrt{n}
$$

Remark 4.4.1. Assumption 3 is a direct extension of Assumption 1. From positive semi-definiteness of $X^{*}$ and $Z^{*}$, we can show that it implies that $\operatorname{Tr}\left(X^{*}\right)+$ $\operatorname{Tr}\left(Z^{*}\right) \leq 2 C_{*} n$, which is one of the assumptions employed in $[188,190]$. Notice that we assume $n>m$, without loss of generality. The theory in this section would hold if $m>n$, simply by replacing $n$ by $m$ in the upper bound of the norm of the optimal solution as well as of the problem data.

Before proceeding with the convergence analysis, we briefly provide an outline of it, for the convenience of the reader. Firstly, it should be noted that polynomial complexity as well as global convergence of Algorithm IP-PMM-SDP follows exactly the developments in Section 3.3. To that end, we provide some necessary technical results in Lemmas 4.4.1-4.4.3. Then, in Lemma 4.4 .4 we are able to show that the iterates $\left(X_{k}, y_{k}, Z_{k}\right)$ of Algorithm IP-PMM-SDP will remain bounded for all $k$. Subsequently, we provide some additional technical results in Lemmas 4.4.5-4.4.7, which are then used in Lemma 4.4.8, where we show that the Newton direction computed at every iteration $k$ is also bounded. All the previous are utilized in Lemma 4.4.9, where we provide a lower bound for the step-length $\alpha_{k}$ chosen by Algorithm IP-PMM-SDP at every iteration $k$. Then, $Q$-linear convergence of $\mu_{k}$ (with $R$-linear convergence of the regularized residuals) is shown in Theorem 4.4.1. Polynomial complexity is proven in Theorem 4.4.2, and finally, global convergence is established in Theorem 4.4.3. Any proof that is a direct generalization from a respective proof in Section 3.3, will be given in the appendix (see Appendix A) for the convenience of the reader.

Let us now use the properties of the proximal operator defined in (4.5).
Lemma 4.4.1. Given Assumption 3, and for all $\eta \in \mathbb{R}^{m}, \Xi \in \mathcal{S}_{+}^{n}$ and $0 \leq \mu<$ $\infty$, there exists a unique pair $\left(X_{r}^{*}, y_{r}^{*}\right)$, such that $\left(X_{r}^{*}, y_{r}^{*}\right)=\mathcal{P}(\Xi, \eta), X_{r}^{*} \in \mathcal{S}_{+}^{n}$, and

$$
\begin{equation*}
\left\|\left(\boldsymbol{X}_{r}^{*}, y_{r}^{*}\right)-\left(\boldsymbol{X}^{*}, y^{*}\right)\right\|_{2} \leq\left\|(\boldsymbol{\Xi}, \eta)-\left(\boldsymbol{X}^{*}, y^{*}\right)\right\|_{2}, \tag{4.23}
\end{equation*}
$$

where $\mathcal{P}(\cdot)$ is defined as in (4.5), and $\left(X^{*}, y^{*}\right)$ is such that $\left(0_{n, n}, 0_{m}\right) \in T_{\mathcal{L}}\left(X^{*}, y^{*}\right)$. Proof. The thesis follows from the developments in [147, Proposition 1].
Lemma 4.4.2. Given Assumptions 3, 4, there exists a triple $\left(X_{r_{k}}^{*}, y_{r_{k}}^{*}, Z_{r_{k}}^{*}\right)$, satisfying:

$$
\begin{align*}
A \boldsymbol{X}_{r_{k}}^{*}+\mu\left(y_{r_{k}}^{*}-\eta_{k}\right)-b & =0_{m}, \\
-\boldsymbol{C}+A^{\top} y_{r_{k}}^{*}+\boldsymbol{Z}_{r_{k}}^{*}-\mu\left(\boldsymbol{X}_{r_{k}}^{*}-\mathbf{\Xi}_{k}\right) & =0_{n^{2}},  \tag{4.24}\\
\left\langle X_{r_{k}}^{*}, Z_{r_{k}}^{*}\right\rangle & =0
\end{align*}
$$

with $X_{r_{k}}^{*}, Z_{r_{k}}^{*} \in \mathcal{S}_{+}^{n}$, and $\left\|\left(\boldsymbol{X}_{r_{k}}^{*}, y_{r_{k}}^{*}, \boldsymbol{Z}_{r_{k}}^{*}\right)\right\|_{2}=\boldsymbol{O}(\sqrt{n})$, for all $\eta_{k} \in \mathbb{R}^{m}, \Xi_{k} \in$ $\mathcal{S}_{+}^{n}$, produced by Algorithm IP-PMM-SDP, and any $\mu \in[0, \infty)$. Moreover, $\left\|\left(\boldsymbol{\Xi}_{k}, \eta_{k}\right)\right\|_{2}=\boldsymbol{O}(\sqrt{n})$, for all $k \geq 0$.

Proof. See Appendix A.1.

Lemma 4.4.3. Given Assumptions 3, 4, $\left(\Xi_{k}, \eta_{k}\right)$, produced at an arbitrary iteration $k \geq 0$ of Algorithm IP-PMM-SDP, and any $\mu \in[0, \infty)$, there exists a triple $(\tilde{X}, \tilde{y}, \tilde{Z})$ which satisfies the following system of equations:

$$
\begin{align*}
A \tilde{\boldsymbol{X}}+\mu \tilde{y} & =b+\bar{b}+\mu \eta_{k}+\tilde{b}_{k}, \\
A^{\top} \tilde{y}+\tilde{\boldsymbol{Z}}-\mu \tilde{\boldsymbol{X}} & =\boldsymbol{C}+\overline{\boldsymbol{C}}-\mu \mathbf{\Xi}_{k}+\tilde{\boldsymbol{C}}_{k},  \tag{4.25}\\
\tilde{X} \tilde{Z} & =\theta I_{n},
\end{align*}
$$

for some arbitrary $\theta>0(\theta=\boldsymbol{\Theta}(1))$, with $\tilde{X}, \tilde{Z} \in \mathcal{S}_{++}^{n}$ and $\|(\tilde{\boldsymbol{X}}, \tilde{y}, \tilde{\boldsymbol{Z}})\|_{2}=$ $\boldsymbol{O}(\sqrt{n})$, where $\tilde{b}_{k}, \tilde{C}_{k}$ are defined in (4.19), while $\bar{b}, \bar{C}$ are defined with the starting point in (4.18). Furthermore, $\lambda_{\min }(\tilde{X}) \geq \xi$ and $\lambda_{\min }(\tilde{Z}) \geq \xi$, for some positive $\xi=\boldsymbol{\Theta}(1)$.

Proof. Let $k \geq 0$ denote an arbitrary iteration of Algorithm IP-PMM-SDP. Let also $\bar{b}, \bar{C}$ as defined in (4.18), and $\tilde{b}_{k}, \tilde{C}_{k}$, as defined in the neighbourhood conditions in (4.19). Given an arbitrary positive constant $\theta>0$, we consider the following barrier primal-dual pair:

$$
\begin{align*}
& \left.\min _{X \in \mathcal{S}^{n}}\left(\left\langle C+\bar{C}+\tilde{C}_{k}\right), X\right\rangle-\theta \ln (\operatorname{det}(X))\right) \text {, s.t. } \mathcal{A} X=b+\bar{b}+\tilde{b}_{k},  \tag{4.26}\\
& \max _{y \in \mathbb{R}^{m}, Z \in \mathcal{S}^{n}}\left(\left(b+\bar{b}+\tilde{b}_{k}\right)^{\top} y+\theta \ln (\operatorname{det}(Z))\right) \text {, s.t. } \mathcal{A}^{*} y+Z=C+\bar{C}+\tilde{C}_{k} . \tag{4.27}
\end{align*}
$$

Let us now define the following triple:

$$
(\hat{X}, \hat{y}, \hat{Z}):=\arg \min _{(X, y, Z)}\left\{\|(\boldsymbol{X}, \boldsymbol{Z})\|_{2}: A \boldsymbol{X}=\tilde{b}_{k}, A^{\top} y+\boldsymbol{Z}=\tilde{C}_{k}\right\} .
$$

From the neighbourhood conditions (4.19), we know that $\left\|\left(\tilde{b}_{k}, \tilde{\boldsymbol{C}}_{k}\right)\right\|_{\mathcal{S}} \leq \gamma_{\mathcal{S}} \rho$, and from the definition of the semi-norm in (4.17), we have that $\|(\hat{\boldsymbol{X}}, \hat{\boldsymbol{Z}})\|_{2} \leq \gamma_{\mathcal{S}} \rho$. Using (4.17) alongside Assumption 4, we can also show that $\|\hat{y}\|_{2}=\boldsymbol{\Theta}\left(\|(\hat{\boldsymbol{X}}, \hat{\boldsymbol{Z}})\|_{2}\right)$. On the other hand, from the definition of the starting point, we have that $\left(X_{0}, Z_{0}\right)=\rho\left(I_{n}, I_{n}\right)$. By defining the following auxiliary point:

$$
(\bar{X}, \bar{y}, \bar{Z})=\left(X_{0}, y_{0}, Z_{0}\right)+(\hat{X}, \hat{y}, \hat{Z})
$$

we have that $\left(1+\gamma_{\mathcal{S}}\right) \rho\left(I_{n}, I_{n}\right) \succeq(\bar{X}, \bar{Z}) \succeq\left(1-\gamma_{\mathcal{S}}\right) \rho\left(I_{n}, I_{n}\right)$, that is, the eigenvalues of these matrices are bounded by constants that are independent of the problem under consideration. By construction, the triple ( $\bar{X}, \bar{y}, \bar{Z}$ ) is a feasible solution for the primal-dual pair in (4.26)-(4.27), giving bounded primal and dual objective values, respectively. This, alongside Weierstrass's theorem on a potential function can be used to show that the solution of problem (4.26)-(4.27) is bounded. In other words, for any choice of $\theta>0$, there must exist a bounded triple ( $X_{s}^{*}, y_{s}^{*}, Z_{s}^{*}$ ) solving (4.26)-(4.27), i.e.:

$$
A \boldsymbol{X}_{s}^{*}=b+\bar{b}+\tilde{b}_{k}, \quad A^{\top} y_{s}^{*}+\boldsymbol{Z}_{s}^{*}=\boldsymbol{C}+\overline{\boldsymbol{C}}+\tilde{\boldsymbol{C}}_{k}, \quad X_{s}^{*} Z_{s}^{*}=\theta I_{n},
$$

such that $\lambda_{\max }\left(X_{s^{*}}\right) \leq K_{s^{*}}$ and $\lambda_{\max }\left(Z_{s^{*}}\right) \leq K_{s^{*}}$, where $K_{s^{*}}>0$ is a positive constant. In turn, combining this with Assumption 4 implies that $\left\|\left(\boldsymbol{X}_{s}^{*}, y_{s}^{*}, \boldsymbol{Z}_{s}^{*}\right)\right\|_{2}=$
$\boldsymbol{O}(\sqrt{n})$.
Let us now apply the PMM to (4.26)-(4.27), given the estimates $\Xi_{k}, \eta_{k}$. We should note at this point that the proximal operator used here is different from that in (4.5), since it is based on a different maximal monotone operator to that in (4.3). In particular, we associate a single-valued maximal monotone operator to (4.26)-(4.27), with graph:
$\tilde{T}_{\mathcal{L}}(X, y):=\left\{(V, u): V=\left(C+\bar{C}+\tilde{C}_{k}\right)-\mathcal{A}^{*} y-\theta X^{-1}, u=\mathcal{A} X-\left(b+\bar{b}+\tilde{b}_{k}\right)\right\}$.
As before, the proximal operator is defined as $\tilde{\mathcal{P}}:=\left(I_{n+m}+\tilde{T}_{\mathcal{L}}\right)^{-1}$, and is singlevalued and non-expansive. We let any $\mu \in[0, \infty)$ and define the following penalty function:

$$
\begin{aligned}
\tilde{\mathcal{L}}_{\mu, \theta}\left(X ; \Xi_{k}, \eta_{k}\right):= & \left\langle C+\bar{C}+\tilde{C}_{k}, X\right\rangle+\frac{1}{2} \mu\left\|X-\Xi_{k}\right\|_{F}^{2}+\frac{1}{2 \mu}\left\|\mathcal{A} X-\left(b+\bar{b}+\tilde{b}_{k}\right)\right\|_{2}^{2} \\
& -\left(\eta_{k}\right)^{\top}\left(\mathcal{A} X-\left(b+\bar{b}+\tilde{b}_{k}\right)\right)-\theta \ln (\operatorname{det}(X))
\end{aligned}
$$

By defining the variables $y=\eta_{k}-\frac{1}{\mu}\left(\mathcal{A} X-\left(b+\bar{b}+\tilde{b}_{k}\right)\right)$ and $Z=\theta X^{-1}$, we can see that the optimality conditions of this PMM sub-problem are exactly those stated in (4.25). Equivalently, we can find a pair $(\tilde{X}, \tilde{y})$ such that $(\tilde{X}, \tilde{y})=\tilde{\mathcal{P}}\left(\Xi_{k}, \eta_{k}\right)$ and set $\tilde{Z}=\theta \tilde{X}^{-1}$. We can now use the non-expansiveness of $\tilde{\mathcal{P}}$, as in Lemma 4.4.1, to obtain:

$$
\left\|(\tilde{\boldsymbol{X}}, \tilde{y})-\left(\boldsymbol{X}_{s}^{*}, y_{s}^{*}\right)\right\|_{2} \leq\left\|\left(\Xi_{k}, \eta_{k}\right)-\left(\boldsymbol{X}_{s}^{*}, y_{s}^{*}\right)\right\|_{2}
$$

But we know, from Lemma 4.4.2, that $\left\|\left(\boldsymbol{\Xi}_{k}, \eta_{k}\right)\right\|_{2}=\boldsymbol{O}(\sqrt{n}), \forall k \geq 0$. Combining this with our previous observations, yields that $\|(\tilde{\boldsymbol{X}}, \tilde{y})\|_{2}=\boldsymbol{O}(\sqrt{n})$. Setting $\tilde{Z}=$ $\theta \tilde{X}^{-1}$, gives a triple $(\tilde{X}, \tilde{y}, \tilde{Z})$ that satisfies (4.25), while $\|(\tilde{\boldsymbol{X}}, \tilde{y}, \tilde{\boldsymbol{Z}})\|_{2}=\boldsymbol{O}(\sqrt{n})$ (from dual feasibility).

To conclude the proof, let us notice that the value of $\tilde{\mathcal{L}}_{\mu, \theta}\left(X ; \Xi_{k}, \eta_{k}\right)$ will grow unbounded as $\lambda_{\min }(X) \rightarrow 0$ or $\lambda_{\max }(X) \rightarrow \infty$. Hence, there must exist a constant $\tilde{K}>0$, such that the minimizer of this function satisfies $\frac{1}{\tilde{K}} \leq \lambda_{\min }(\tilde{X}) \leq$ $\lambda_{\max }(\tilde{X}) \leq \tilde{K}$. The relation $\tilde{X} \tilde{Z}=\theta I_{n}$ then implies that $\frac{\theta}{\tilde{K}} \leq \lambda_{\min }(\tilde{Z}) \leq$ $\lambda_{\max }(\tilde{Z}) \leq \theta \tilde{K}$. Hence, there exists some $\xi=\boldsymbol{\Theta}(1)$ such that $\lambda_{\min }(\tilde{X}) \geq \xi$ and $\lambda_{\min }(\tilde{Z}) \geq \xi$.

In the following lemma, we derive boundedness of the iterates of Algorithm IP-PMM-SDP.

Lemma 4.4.4. Given Assumptions 3 and 4, the iterates $\left(X_{k}, y_{k}, Z_{k}\right)$ produced by Algorithm IP-PMM-SDP, for all $k \geq 0$, are such that:

$$
\operatorname{Tr}\left(X_{k}\right)=\boldsymbol{O}(n), \quad \operatorname{Tr}\left(Z_{k}\right)=\boldsymbol{O}(n), \quad\left\|\left(\boldsymbol{X}_{k}, y_{k}, \boldsymbol{Z}_{k}\right)\right\|_{2}=\boldsymbol{O}(n)
$$

Proof. See Appendix A.2.
In what follows, we provide Lemmas 4.4.5-4.4.7, which we use to prove boundedness of the Newton direction computed at every iteration of Algorithm IP-PMM-SDP, in Lemma 4.4.8.

Lemma 4.4.5. Let $D_{k}=S_{k}^{-\frac{1}{2}} F_{k}=S_{k}^{\frac{1}{2}} E_{k}^{-1}$, where $S_{k}=E_{k} F_{k}$, and $E_{k}, F_{k}$ are defined as in the Newton system in (4.21). Then, for any $M \in \mathbb{R}^{n \times n}$,

$$
\left\|D_{k}^{-T} \boldsymbol{M}\right\|_{2}^{2} \leq \frac{1}{\left(1-\gamma_{\mu}\right) \mu_{k}}\left\|Z_{k}^{\frac{1}{2}} M Z_{k}^{\frac{1}{2}}\right\|_{F}^{2}, \quad\left\|D_{k} \boldsymbol{M}\right\|_{2}^{2} \leq \frac{1}{\left(1-\gamma_{\mu}\right) \mu_{k}}\left\|X_{k}^{\frac{1}{2}} M X_{k}^{\frac{1}{2}}\right\|_{F}^{2}
$$

where $\gamma_{\mu}$ is defined in (4.19). Moreover, we have that:
$\left\|D_{k}^{-T}\right\|_{2}^{2} \leq \frac{1}{\left(1-\gamma_{\mu}\right) \mu_{k}}\left\|Z_{k}\right\|_{F}^{2}=\boldsymbol{O}\left(\frac{n^{2}}{\mu_{k}}\right), \quad\left\|D_{k}\right\|_{2}^{2} \leq \frac{1}{\left(1-\gamma_{\mu}\right) \mu_{k}}\left\|X_{k}\right\|_{F}^{2}=\boldsymbol{O}\left(\frac{n^{2}}{\mu_{k}}\right)$.
Proof. The proof of the first two inequalities follows exactly the developments in [190, Lemma 5]. The bound on the 2-norm of the matrix $D_{k}^{-T}$ follows by choosing $M$ such that $M$ is a unit eigenvector, corresponding to the largest eigenvalue of $D_{k}^{-T}$. Then, $\left\|D_{k}^{-T} \boldsymbol{M}\right\|_{2}^{2}=\left\|D_{k}^{-T}\right\|_{2}^{2}$. But, we have that:

$$
\begin{aligned}
\left\|D_{k}^{-T} \boldsymbol{M}\right\|_{2}^{2} & \leq \frac{1}{\left(1-\gamma_{\mu}\right) \mu_{k}}\left\|Z_{k}^{\frac{1}{2}} M Z_{k}^{\frac{1}{2}}\right\|_{F}^{2} \\
& =\frac{1}{\left(1-\gamma_{\mu}\right) \mu_{k}} \operatorname{Tr}\left(Z_{k} M^{\top} Z_{k} M\right) \\
& \leq \frac{1}{\left(1-\gamma_{\mu}\right) \mu_{k}}\left\|Z_{k}\right\|_{F}^{2}=\boldsymbol{O}\left(\frac{n^{2}}{\mu_{k}}\right)
\end{aligned}
$$

where we used the cyclic property of the trace as well as Lemma 4.4.4. The same reasoning applies to deriving the bound for $\left\|D_{k}\right\|_{2}^{2}$.

Lemma 4.4.6. Let $D_{k}$ and $S_{k}$ as defined in Lemma 4.4.5. Then, we have that:

$$
\left\|D_{k}^{-T} \boldsymbol{\Delta} \boldsymbol{X}_{k}\right\|_{2}^{2}+\left\|D_{k} \boldsymbol{\Delta} \boldsymbol{Z}_{k}\right\|_{2}^{2}+2\left\langle\Delta X_{k}, \Delta Z_{k}\right\rangle=\left\|S_{k}^{-\frac{1}{2}} \boldsymbol{R}_{\mu, k}\right\|_{2}^{2}
$$

where $R_{\mu, k}=\tau_{k} \mu_{k} I_{n}-Z_{k}^{\frac{1}{2}} X_{k} Z_{k}^{\frac{1}{2}}$. Furthermore,

$$
\left\|H_{P_{k}}\left(\Delta X_{k} \Delta Z_{k}\right)\right\|_{F} \leq \frac{\sqrt{\frac{1+\gamma_{\mu}}{1-\gamma_{\mu}}}}{2}\left(\left\|D_{k}^{-T} \boldsymbol{\Delta} \boldsymbol{X}_{k}\right\|_{2}^{2}+\left\|D_{k} \boldsymbol{\Delta} \boldsymbol{Z}_{k}\right\|_{2}^{2}\right)
$$

where $\gamma_{\mu}$ is defined in (4.19).
Proof. The equality follows directly by pre-multiplying by $S^{-\frac{1}{2}}$ on both sides of the third block equation of the Newton system in (4.21) and by then taking the 2 -norm (see [188, Lemma 3.1]). For a proof of the inequality, we refer the reader to [188, Lemma 3.3].

Lemma 4.4.7. Let $S_{k}$ as defined in Lemma 4.4.5, and $R_{\mu, k}$ as defined in Lemma 4.4.6. Then,

$$
\left\|S_{k}^{-\frac{1}{2}} \boldsymbol{R}_{\mu, k}\right\|_{2}^{2}=\boldsymbol{O}\left(n \mu_{k}\right)
$$

Proof. The proof is omitted since it follows exactly the developments in [190, Lemma 7].

Lemma 4.4.8. Given Assumptions 3 and 4, and the Newton direction, denoted as $\left(\Delta X_{k}, \Delta y_{k}, \Delta Z_{k}\right)$, obtained by solving system (4.21) during an arbitrary iteration $k \geq 0$ of Algorithm IP-PMM-SDP, we have that:

$$
\left\|H_{P_{k}}\left(\Delta X_{k} \Delta Z_{k}\right)\right\|_{F}=\boldsymbol{O}\left(n^{4} \mu\right), \quad\left\|\left(\boldsymbol{\Delta} \boldsymbol{X}_{k}, \Delta y_{k}, \boldsymbol{\Delta} \boldsymbol{Z}_{k}\right)\right\|_{2}=\boldsymbol{O}\left(n^{3}\right)
$$

Proof. Consider an arbitrary iteration $k$ of Algorithm IP-PMM-SDP. We invoke Lemmas 4.4.2, 4.4.3, for $\mu=\tau_{k} \mu_{k}$. That is, there exists a triple $\left(X_{r_{k}}^{*}, y_{r_{k}}^{*}, Z_{r_{k}}^{*}\right)$ satisfying (4.24), and a triple ( $\tilde{X}, \tilde{y}, \tilde{Z})$ satisfying (4.25), for $\mu=\tau_{k} \mu_{k}$. Using the centering parameter $\tau_{k}$, define:

$$
\begin{align*}
\hat{\boldsymbol{C}} & =-\left(\frac{\tau_{k}}{\mu_{0}} \overline{\boldsymbol{C}}-\left(1-\tau_{k}\right)\left(\boldsymbol{X}_{k}-\boldsymbol{\Xi}_{k}+\frac{\mu_{k}}{\mu_{0}}\left(\tilde{\boldsymbol{X}}-\boldsymbol{X}_{r_{k}}^{*}\right)\right)+\frac{1}{\mu_{k}} \mathbf{E}_{d, k}\right),  \tag{4.28}\\
\hat{b} & =-\left(\frac{\tau_{k}}{\mu_{0}} \bar{b}+\left(1-\tau_{k}\right)\left(y_{k}-\eta_{k}+\frac{\mu_{k}}{\mu_{0}}\left(\tilde{y}-y_{r_{k}}^{*}\right)\right)+\frac{1}{\mu_{k}} \epsilon_{p, k}\right)
\end{align*}
$$

where $\bar{b}, \bar{C}, \mu_{0}$ are defined in (4.18) and $\epsilon_{p, k}, \mathrm{E}_{d, k}$ model the errors which occur when system (4.20) is solved inexactly. Notice that these errors are required to satisfy (4.22) at every iteration $k$. Using Lemmas 4.4.2, 4.4.3, 4.4.4, relation (4.22), and Assumption 4, we know that $\|(\hat{\boldsymbol{C}}, \hat{b})\|_{2}=\boldsymbol{O}(n)$. Then, by applying again Assumption 4, we know that there must exist a matrix $\hat{X} \in \mathbb{R}^{n \times n}$ such that $\mathcal{A} \hat{X}=\hat{b},\|\hat{X}\|_{F}=\boldsymbol{O}(n)$, and by setting $\hat{Z}=\hat{C}+\mu \hat{X}$, we have that $\|\hat{Z}\|_{F}=\boldsymbol{O}(n)$ and:

$$
\begin{equation*}
\mathcal{A} \hat{X}=\hat{b}, \quad \hat{Z}-\mu_{k} \hat{X}=\hat{C} . \tag{4.29}
\end{equation*}
$$

Using $\left(X_{r_{k}}^{*}, y_{r_{k}}^{*}, Z_{r_{k}}^{*}\right),(\tilde{X}, \tilde{y}, \tilde{Z})$, as well as the triple $\left(\hat{X}, 0_{m}, \hat{Z}\right)$, where $(\hat{X}, \hat{Z})$ is defined in (4.29), we can define the following auxiliary triple:

$$
\begin{equation*}
(\bar{X}, \bar{y}, \bar{Z})=\left(\Delta X_{k}, \Delta y_{k}, \Delta Z_{k}\right)+\frac{\mu_{k}}{\mu_{0}}(\tilde{X}, \tilde{y}, \tilde{Z})-\frac{\mu_{k}}{\mu_{0}}\left(X_{r_{k}}^{*}, y_{r_{k}}^{*}, Z_{r_{k}}^{*}\right)+\mu_{k}\left(\hat{X}, 0_{m}, \hat{Z}\right) \tag{4.30}
\end{equation*}
$$

Using (4.30), (4.28), and the second block equation of (4.21):

$$
\begin{aligned}
A \overline{\boldsymbol{X}}+\mu_{k} \bar{y}= & \left(A \boldsymbol{\Delta} \boldsymbol{X}_{k}+\mu_{k} \Delta y_{k}\right)+\frac{\mu_{k}}{\mu_{0}}\left(\left(A \tilde{\boldsymbol{X}}+\mu_{k} \tilde{y}\right)-\left(A \boldsymbol{X}_{r_{k}}^{*}+\mu_{k} y_{r_{k}}^{*}\right)\right)+\mu_{k} A \hat{\boldsymbol{X}} \\
= & \left(b+\tau_{k} \frac{\mu_{k}}{\mu_{0}} \bar{b}-A \boldsymbol{X}_{k}-\tau_{k} \mu_{k}\left(y_{k}-\eta_{k}\right)+\epsilon_{p, k}\right) \\
& +\frac{\mu_{k}}{\mu_{0}}\left(\left(A \boldsymbol{X}+\mu_{k} \tilde{y}\right)-\left(A \boldsymbol{X}_{r_{k}}^{*}+\mu_{k} y_{r_{k}}^{*}\right)\right) \\
& -\mu_{k}\left(\tau_{k} \frac{\bar{b}}{\mu_{0}}+\left(1-\tau_{k}\right)\left(y_{k}-\eta_{k}\right)\right)-\frac{\mu_{k}}{\mu_{0}}\left(1-\tau_{k}\right) \mu_{k}\left(\tilde{y}-y_{r_{k}}^{*}\right)-\epsilon_{p, k}
\end{aligned}
$$

Then, by deleting opposite terms in the right-hand side, and employing (4.24)(4.25) (evaluated at $\mu=\tau_{k} \mu_{k}$ from the definition of $\left(X_{r_{k}}^{*}, y_{r_{k}}^{*}, Z_{r_{k}}^{*}\right)$ and $\left.(\tilde{X}, \tilde{y}, \tilde{Z})\right)$,
we have

$$
\begin{aligned}
A \overline{\boldsymbol{X}}+\mu_{k} \bar{y}= & \left(b+\tau_{k} \frac{\mu_{k}}{\mu_{0}} \bar{b}-A \boldsymbol{X}_{k}-\tau_{k} \mu_{k}\left(y_{k}-\eta_{k}\right)\right)+\frac{\mu_{k}}{\mu_{0}}\left(b+\tau_{k} \mu_{k} \eta_{k}+\bar{b}+\tilde{b}_{k}\right) \\
& -\frac{\mu_{k}}{\mu_{0}}\left(\tau_{k} \mu_{k} \eta_{k}+b\right)-\mu_{k}\left(\tau_{k} \frac{\bar{b}}{\mu_{0}}+\left(1-\tau_{k}\right)\left(y_{k}-\eta_{k}\right)\right) \\
= & b+\frac{\mu_{k}}{\mu_{0}}\left(\bar{b}+\tilde{b}_{k}\right)-A \boldsymbol{X}_{k}-\mu_{k}\left(y_{k}-\eta_{k}\right) \\
= & 0_{m},
\end{aligned}
$$

where the last equation follows from the neighbourhood conditions $\left(\left(X_{k}, y_{k}, Z_{k}\right) \in\right.$ $\left.\mathscr{N}_{\mu_{k}}\left(\Xi_{k}, \eta_{k}\right)\right)$. Similarly, we can show that:

$$
A^{\top} \bar{y}+\bar{Z}-\mu_{k} \bar{X}=0_{n^{2}} .
$$

The previous two equalities imply that:

$$
\begin{equation*}
\langle\bar{X}, \bar{Z}\rangle=\left\langle\bar{X},-\mathcal{A}^{*} \bar{y}+\mu_{k} \bar{X}\right\rangle=\mu_{k}\langle\bar{X}, \bar{X}\rangle+\mu_{k} \bar{y}^{\top} \bar{y} \geq 0 . \tag{4.31}
\end{equation*}
$$

On the other hand, using the last block equation of the Newton system (4.21), we have:

$$
E_{k} \overline{\boldsymbol{X}}+F_{k} \overline{\boldsymbol{Z}}=\boldsymbol{R}_{\mu, k}+\frac{\mu_{k}}{\mu_{0}} E_{k}\left(\tilde{\boldsymbol{X}}-\boldsymbol{X}_{r_{k}}^{*}+\mu_{0} \hat{\boldsymbol{X}}\right)+\frac{\mu_{k}}{\mu_{0}} F_{k}\left(\tilde{\boldsymbol{Z}}-\boldsymbol{Z}_{r_{k}}^{*}+\mu_{0} \hat{\boldsymbol{Z}}\right)
$$

where $R_{\mu, k}$ is defined as in Lemma 4.4.6. Let $S_{k}$ be defined as in Lemma 4.4.5. By multiplying both sides of the previous equation by $S_{k}^{-\frac{1}{2}}$, we get:
$D_{k}^{-T} \overline{\boldsymbol{X}}+D_{k} \overline{\boldsymbol{Z}}=S_{k}^{-\frac{1}{2}} \boldsymbol{R}_{\mu, k}+\frac{\mu_{k}}{\mu_{0}}\left(D_{k}^{-T}\left(\tilde{\boldsymbol{X}}-\boldsymbol{X}_{r_{k}}^{*}+\mu_{0} \hat{\boldsymbol{X}}\right)+D_{k}\left(\tilde{\boldsymbol{Z}}-\boldsymbol{Z}_{r_{k}}^{*}+\mu_{0} \hat{\boldsymbol{Z}}\right)\right)$.
But from (4.31) we know that $\langle\bar{X}, \bar{Z}\rangle \geq 0$ and hence:

$$
\left\|D_{k}^{-T} \overline{\boldsymbol{X}}+D_{k} \overline{\boldsymbol{Z}}\right\|_{2}^{2} \geq\left\|D_{k}^{-T} \overline{\boldsymbol{X}}\right\|_{2}^{2}+\left\|D_{k} \overline{\boldsymbol{Z}}\right\|_{2}^{2} .
$$

Combining (4.32) with the previous inequality, gives:

$$
\begin{gathered}
\left\|D_{k}^{-T} \overline{\boldsymbol{X}}\right\|_{2}^{2} \leq\left\{\left\|S_{k}^{-\frac{1}{2}} \boldsymbol{R}_{\mu, k}\right\|_{2}+\frac{\mu_{k}}{\mu_{0}}\left(\left\|D_{k}^{-T}\left(\tilde{\boldsymbol{X}}-\boldsymbol{X}_{r_{k}}^{*}+\mu_{0} \hat{\boldsymbol{X}}\right)\right\|_{2}\right.\right. \\
\left.\left.+\left\|D_{k}\left(\tilde{\boldsymbol{Z}}-\boldsymbol{Z}_{r_{k}}^{*}+\mu_{0} \hat{\boldsymbol{Z}}\right)\right\|_{2}\right)\right\}^{2}
\end{gathered}
$$

We take square roots, use (4.30) and apply the triangle inequality, to get:

$$
\begin{align*}
\left\|D_{k}^{-T} \boldsymbol{\Delta} \boldsymbol{X}_{k}\right\|_{2} \leq & \left\|S_{k}^{-\frac{1}{2}} \boldsymbol{R}_{\mu, k}\right\|_{2}+\frac{\mu_{k}}{\mu_{0}}\left(2\left\|D_{k}^{-T}\left(\tilde{\boldsymbol{X}}-\boldsymbol{X}_{r_{k}}^{*}+\mu_{0} \hat{\boldsymbol{X}}\right)\right\|_{2}\right. \\
& \left.+\left\|D_{k}\left(\tilde{\boldsymbol{Z}}-\boldsymbol{Z}_{r_{k}}^{*}+\mu_{0} \hat{\boldsymbol{Z}}\right)\right\|_{2}\right) . \tag{4.33}
\end{align*}
$$

We now proceed to bounding the terms in the right hand side of (4.33). A bound for the first term of the right hand side is given by Lemma 4.4.7, that is:

$$
\left\|S_{k}^{-\frac{1}{2}} \boldsymbol{R}_{\mu, k}\right\|_{2}=\boldsymbol{O}\left(n^{\frac{1}{2}} \mu_{k}^{\frac{1}{2}}\right)
$$

On the other hand, we have (from Lemma 4.4.5) that

$$
\left\|D_{k}^{-T}\right\|_{2}=\boldsymbol{O}\left(\frac{n}{\mu_{k}^{\frac{1}{2}}}\right), \quad\left\|D_{k}\right\|_{2}=\boldsymbol{O}\left(\frac{n}{\mu_{k}^{\frac{1}{2}}}\right)
$$

Hence, using the previous bounds, as well as Lemmas 4.4.2, 4.4.3, and (4.29), we obtain:

$$
2 \frac{\mu_{k}}{\mu_{0}}\left\|D_{k}^{-T}\left(\tilde{\boldsymbol{X}}-\boldsymbol{X}_{r_{k}}^{*}+\mu_{0} \hat{\boldsymbol{X}}\right)\right\|_{2}+\frac{\mu_{k}}{\mu_{0}}\left\|D_{k}\left(\tilde{\boldsymbol{Z}}-\boldsymbol{Z}_{r_{k}}^{*}+\mu_{0} \hat{\boldsymbol{Z}}\right)\right\|_{2}=\boldsymbol{O}\left(n^{2} \mu_{k}^{\frac{1}{2}}\right)
$$

Combining all the previous bounds yields that $\left\|D_{k}^{-T} \boldsymbol{\Delta} \boldsymbol{X}_{k}\right\|_{2}=\boldsymbol{O}\left(n^{2} \mu_{k}^{\frac{1}{2}}\right)$. One can bound $\left\|D_{k} \boldsymbol{\Delta} \boldsymbol{Z}_{k}\right\|_{2}$ in the same way. The latter is omitted for ease of presentation.

Furthermore, we have that:

$$
\left\|\boldsymbol{\Delta} \boldsymbol{X}_{k}\right\|_{2}=\left\|D_{k} D_{k}^{-T} \boldsymbol{\Delta} \boldsymbol{X}_{k}\right\|_{2} \leq\left\|D_{k}\right\|_{2}\left\|D_{k}^{-T} \boldsymbol{\Delta} \boldsymbol{X}_{k}\right\|_{2}=\boldsymbol{O}\left(n^{3}\right)
$$

Similarly, we can show that $\left\|\boldsymbol{\Delta} \boldsymbol{Z}_{k}\right\|_{2}=\boldsymbol{O}\left(n^{3}\right)$. From the first block equation of the Newton system in (4.21), alongside Assumption 4, we can show that $\left\|\Delta y_{k}\right\|_{2}=$ $\boldsymbol{O}\left(n^{3}\right)$.

Finally, using the previous bounds, as well as Lemma 4.4.6, we obtain the desired bound on $\left\|H_{P_{k}}\left(\Delta X_{k} \Delta Z_{k}\right)\right\|_{F}$, that is:

$$
\left\|H_{P_{k}}\left(\Delta X_{k} \Delta Z_{k}\right)\right\|_{F}=\boldsymbol{O}\left(n^{4} \mu_{k}\right)
$$

which completes the proof.
As in Section 3.3, we can now prove that at every iteration of Algorithm IP-PMMSDP there exists a step-length $\alpha_{k}>0$, using which, the new iterate satisfies the conditions required by the algorithm. To that end, we assume the following notation:

$$
\left(X_{k}(\alpha), y_{k}(\alpha), Z_{k}(\alpha)\right) \equiv\left(X_{k}+\alpha \Delta X_{k}, y_{k}+\alpha \Delta y_{k}, Z_{k}+\alpha \Delta Z_{k}\right)
$$

Lemma 4.4.9. Given Assumptions 3, 4, and by letting $P_{k}(\alpha)=Z_{k}(\alpha)^{\frac{1}{2}}$, there exists a step-length $\bar{\alpha} \in(0,1)$, such that for all $\alpha \in[0, \bar{\alpha}]$ and for all iterations $k \geq 0$ of Algorithm IP-PMM-SDP, the following relations hold:

$$
\begin{gather*}
\left\langle X_{k}+\alpha \Delta X_{k}, Z_{k}+\alpha \Delta Z_{k}\right\rangle \geq\left(1-\alpha\left(1-\beta_{1}\right)\right)\left\langle X_{k}, Z_{k}\right\rangle,  \tag{4.34}\\
\left\|H_{P_{k}(\alpha)}\left(X_{k}(\alpha) Z_{k}(\alpha)\right)-\mu_{k}(\alpha)\right\|_{F} \leq \gamma_{\mu} \mu_{k}(\alpha),  \tag{4.35}\\
\left\langle X_{k}+\alpha \Delta X_{k}, Z_{k}+\alpha \Delta Z_{k}\right\rangle \leq\left(1-\alpha\left(1-\beta_{2}\right)\right)\left\langle X_{k}, Z_{k}\right\rangle, \tag{4.36}
\end{gather*}
$$

where, without loss of generality, $\beta_{1}=\frac{\tau_{\min }}{2}$ and $\beta_{2}=0.99$. Moreover, $\bar{\alpha} \geq \frac{\bar{\varepsilon}}{n^{4}}$ for
all $k \geq 0$, where $\bar{\kappa}>0$ is independent of $n$, $m$, and if $\left(X_{k}, y_{k}, Z_{k}\right) \in \mathscr{N}_{\mu_{k}}\left(\Xi_{k}, \eta_{k}\right)$, then letting:

$$
\left(X_{k+1}, y_{k+1}, Z_{k+1}\right)=\left(X_{k}+\alpha \Delta X_{k}, y_{k}+\alpha \Delta y_{k}, Z_{k}+\alpha \Delta Z_{k}\right), \mu_{k+1}=\frac{\left\langle X_{k+1}, Z_{k+1}\right\rangle}{n}
$$

for any $\alpha \in(0, \bar{\alpha}]$, gives $\left(X_{k+1}, y_{k+1}, Z_{k+1}\right) \in \mathscr{N}_{\mu_{k+1}}\left(\Xi_{k+1}, \eta_{k+1}\right)$, where $\Xi_{k}$, and $\eta_{k}$ are updated as in Algorithm IP-PMM-SDP.

Proof. We proceed by proving the first three inequalities stated in the Lemma. From Lemma 4.4.8, there exist constants $K_{\Delta}>0$ and $K_{H \Delta}>0$, independent of $n$ and $m$, such that:

$$
\begin{gathered}
\left\langle\Delta X_{k}, \Delta Z_{k}\right\rangle=\left(D_{k}^{-T} \boldsymbol{\Delta} \boldsymbol{X}_{k}\right)^{\top}\left(D_{k} \boldsymbol{\Delta} \boldsymbol{Z}_{k}\right) \leq\left\|D_{k}^{-T} \boldsymbol{\Delta} \boldsymbol{X}_{k}\right\|_{2}\left\|D_{k} \boldsymbol{\Delta} \boldsymbol{Z}_{k}\right\|_{2} \leq K_{\Delta}^{2} n^{4} \mu_{k}, \\
\left\|H_{P_{k}}\left(\Delta X_{k} \Delta Z_{k}\right)\right\|_{F} \leq K_{H \Delta} n^{4} \mu_{k} .
\end{gathered}
$$

From the last block equation of the Newton system (4.20), we can show that:

$$
\begin{equation*}
\left\langle Z_{k}, \Delta X_{k}\right\rangle+\left\langle X_{k}, \Delta Z_{k}\right\rangle=\left(\tau_{k}-1\right)\left\langle X_{k}, Z_{k}\right\rangle . \tag{4.37}
\end{equation*}
$$

The latter can also be obtained from (4.21), since we require $\left\|\mathrm{E}_{\mu, k}\right\|=0$. Furthermore:

$$
\begin{equation*}
H_{P_{k}}\left(X_{k}(\alpha) Z_{k}(\alpha)\right)=(1-\alpha) H_{P_{k}}\left(X_{k} Z_{k}\right)+\alpha \tau_{k} \mu_{k} I_{n}+\alpha^{2} H_{P_{k}}\left(\Delta X_{k} \Delta Z_{k}\right) . \tag{4.38}
\end{equation*}
$$

We proceed by proving (4.34). Using (4.37), we have:

$$
\begin{aligned}
& \left\langle X_{k}+\alpha \Delta X_{k}, Z_{k}+\alpha \Delta Z_{k}\right\rangle-\left(1-\alpha\left(1-\beta_{1}\right)\right)\left\langle X_{k}, Z_{k}\right\rangle \\
& \quad=\left\langle X_{k}, Z_{k}\right\rangle+\alpha\left(\tau_{k}-1\right)\left\langle X_{k}, Z_{k}\right\rangle+\alpha^{2}\left\langle\Delta X_{k}, \Delta Z_{k}\right\rangle-(1-\alpha)\left\langle X_{k}, Z_{k}\right\rangle-\alpha \beta_{1}\left\langle X_{k}, Z_{k}\right\rangle \\
& \geq \alpha\left(\tau_{k}-\beta_{1}\right)\left\langle X_{k}, Z_{k}\right\rangle-\alpha^{2} K_{\Delta}^{2} n^{4} \mu_{k} \geq \alpha\left(\frac{\tau_{\min }}{2}\right) n \mu_{k}-\alpha^{2} K_{\Delta}^{2} n^{4} \mu_{k},
\end{aligned}
$$

where we set (without loss of generality) $\beta_{1}=\frac{\tau_{\text {min }}}{2}$. The rightmost term of the previous inequality will be non-negative for every $\alpha$ satisfying:

$$
\alpha \leq \frac{\tau_{\min }}{2 K_{\Delta}^{2} n^{3}} .
$$

In order to prove (4.35), we will use (4.38) and the fact that from the neighbourhood conditions we have $\left\|H_{P_{k}}\left(X_{k} Z_{k}\right)-\mu_{k}\right\|_{F} \leq \gamma_{\mu} \mu_{k}$. To that end, we use the result in [188, Lemma 4.2], stating that:

$$
\left\|H_{P_{k}(\alpha)}\left(X_{k}(\alpha) Z_{k}(\alpha)\right)-\mu_{k}(\alpha) I_{n}\right\|_{F} \leq\left\|H_{P_{k}}\left(X_{k}(\alpha) Z_{k}(\alpha)\right)-\mu_{k}(\alpha) I_{n}\right\|_{F} .
$$

The latter yields

$$
\begin{aligned}
& \left\|H_{P_{k}(\alpha)}\left(X_{k}(\alpha) Z_{k}(\alpha)\right)-\mu_{k}(\alpha) I_{n}\right\|_{F}-\gamma_{\mu} \mu_{k}(\alpha) \\
& \quad \leq\left\|H_{P_{k}}\left(X_{k}(\alpha) Z_{k}(\alpha)\right)-\mu_{k}(\alpha) I_{n}\right\|_{F}-\gamma_{\mu} \mu_{k}(\alpha)
\end{aligned}
$$

By combining all the previous, we have:

$$
\begin{aligned}
&\left\|H_{P_{k}}\left(X_{k}(\alpha) Z_{k}(\alpha)\right)-\mu_{k}(\alpha) I_{n}\right\|_{F}-\gamma_{\mu} \mu_{k}(\alpha) \\
&= \|(1-\alpha)\left(H_{P_{k}}\left(X_{k} Z_{k}\right)-\mu_{k} I_{n}\right)+\alpha^{2} H_{P_{k}}\left(\Delta X_{k}, \Delta Z_{k}\right) \\
&-\frac{\alpha^{2}}{n}\left\langle\Delta X_{k}, \Delta Z_{k}\right\rangle I_{n} \|_{F}-\gamma_{\mu} \mu_{k}(\alpha) \\
& \leq(1-\alpha)\left\|H_{P_{k}}\left(X_{k} Z_{k}\right)-\mu_{k} I_{n}\right\|_{F}+\alpha^{2} \mu_{k}\left(\frac{K_{\Delta}^{2}}{n}+K_{H \Delta}\right) n^{4} \\
&-\gamma_{\mu}\left((1-\alpha) \mu_{k}+\alpha \tau_{k} \mu_{k}+\frac{\alpha^{2}}{n}\left\langle\Delta X_{k}, \Delta Z_{k}\right\rangle\right) \\
& \leq-\gamma_{\mu} \alpha \tau_{\min } \mu_{k}+\alpha^{2} \mu_{k}\left(\frac{2 K_{\Delta}^{2}}{n}+K_{H \Delta}\right) n^{4}
\end{aligned}
$$

where we used the neighbourhood conditions in (4.19), the equality $\mu_{k}(\alpha)=$ $(1-\alpha) \mu_{k}+\alpha \tau_{k} \mu_{k}+\frac{\alpha^{2}}{n}\left\langle\Delta X_{k}, \Delta Z_{k}\right\rangle$ (which can be derived from (4.37)), and the third block equation of the Newton system (4.21). The rightmost term of the previous inequality is non-positive for every $\alpha$ satisfying:

$$
\alpha \leq \frac{\tau_{\min } \gamma_{\mu}}{\left(\frac{2 K_{\Delta}^{2}}{n}+K_{H \Delta}\right) n^{4}} .
$$

Finally, to prove (4.36), we set (without loss of generality) $\beta_{2}=0.99$. We know, from Algorithm IP-PMM-SDP, that $\tau_{\max } \leq 0.5$. With the previous two remarks in mind, we have:

$$
\begin{aligned}
\frac{1}{n}\left\langle X_{k}\right. & \left.+\alpha \Delta X_{k}, Z_{k}+\alpha \Delta Z_{k}\right\rangle-(1-0.01 \alpha) \mu_{k} \\
& \leq(1-\alpha) \mu_{k}+\alpha \tau_{k} \mu_{k}+\alpha^{2} \frac{K_{\Delta}^{2} n^{4}}{n} \mu_{k}-(1-0.01 \alpha) \mu_{k} \\
& \leq-0.99 \alpha \mu_{k}+0.5 \alpha \mu_{k}+\alpha^{2} \frac{K_{\Delta}^{2} n^{4}}{n} \mu_{k} \\
& =-0.49 \alpha \mu_{k}+\alpha^{2} \frac{K_{\Delta}^{2} n^{4}}{n} \mu_{k}
\end{aligned}
$$

The last term will be non-positive for every $\alpha$ satisfying:

$$
\alpha \leq \frac{0.49}{K_{\Delta}^{2} n^{3}} .
$$

By combining all the previous bounds on the step-length, we have that (4.34)(4.36) hold for every $\alpha \in\left(0, \alpha^{*}\right]$, where:

$$
\begin{equation*}
\alpha^{*}:=\min \left\{\frac{\tau_{\min }}{2 K_{\Delta}^{2} n^{3}}, \frac{\tau_{\min } \gamma_{\mu}}{\left(\frac{2 K_{\Delta}^{2}}{n}+K_{H \Delta}\right) n^{4}}, \frac{0.49}{K_{\Delta}^{2} n^{3}}, 1\right\} \tag{4.39}
\end{equation*}
$$

Next, we would like to find the maximum $\bar{\alpha} \in\left(0, \alpha^{*}\right]$, such that:

$$
\left(X_{k}(\alpha), y_{k}(\alpha), Z_{k}(\alpha)\right) \in \mathscr{N}_{\mu_{k}(\alpha)}\left(\Xi_{k}, \eta_{k}\right), \text { for all } \alpha \in(0, \bar{\alpha}),
$$

where $\mu_{k}(\alpha)=\frac{\left\langle X_{k}(\alpha), Z_{k}(\alpha)\right\rangle}{n}$. Let:

$$
\begin{equation*}
\tilde{r}_{p}(\alpha)=A \boldsymbol{X}_{k}(\alpha)+\mu_{k}(\alpha)\left(y_{k}(\alpha)-\eta_{k}\right)-\left(b+\frac{\mu_{k}(\alpha)}{\mu_{0}} \bar{b}\right), \tag{4.40}
\end{equation*}
$$

and

$$
\begin{equation*}
\tilde{\boldsymbol{R}}_{d}(\alpha)=A^{\top} y_{k}(\alpha)+\boldsymbol{Z}_{k}(\alpha)-\mu_{k}(\alpha)\left(\boldsymbol{X}_{k}(\alpha)-\boldsymbol{\Xi}_{k}\right)-\left(\boldsymbol{C}+\frac{\mu_{k}(\alpha)}{\mu_{0}} \overline{\boldsymbol{C}}\right) . \tag{4.41}
\end{equation*}
$$

In other words, we need to find the maximum $\bar{\alpha} \in\left(0, \alpha^{*}\right]$, such that:

$$
\begin{equation*}
\left\|\tilde{r}_{p}(\alpha), \tilde{\boldsymbol{R}}_{d}(\alpha)\right\|_{2} \leq K_{N} \frac{\mu_{k}(\alpha)}{\mu_{0}},\left\|\tilde{r}_{p}(\alpha), \tilde{\boldsymbol{R}}_{d}(\alpha)\right\|_{\mathcal{S}} \leq \gamma_{\mathcal{S}} \rho \frac{\mu_{k}(\alpha)}{\mu_{0}}, \text { for all } \alpha \in(0, \bar{\alpha}) \tag{4.42}
\end{equation*}
$$

If the latter two conditions hold, then $\left(X_{k}(\alpha), y_{k}(\alpha), Z_{k}(\alpha)\right) \in \mathscr{N}_{\mu_{k}(\alpha)}\left(\Xi_{k}, \eta_{k}\right)$, for all $\alpha \in(0, \bar{\alpha})$. Then, if Algorithm IP-PMM-SDP updates $\Xi_{k}$, and $\eta_{k}$, it does so only when similar conditions (as in (4.42)) hold for the new parameters. If the parameters are not updated, the new iterate lies in the desired neighbourhood because of (4.42), alongside (4.34)-(4.36).

We start by rearranging $\tilde{r}_{p}(\alpha)$. Specifically, we have that:

$$
\begin{aligned}
\tilde{r}_{p}(\alpha)= & A\left(\boldsymbol{X}_{k}+\alpha \boldsymbol{\Delta} \boldsymbol{X}_{k}\right)+\left(\mu_{k}+\alpha\left(\tau_{k}-1\right) \mu_{k}\right. \\
& \left.+\frac{\alpha^{2}}{n}\left\langle\Delta X_{k}, \Delta Z_{k}\right\rangle\right)\left(\left(y_{k}+\alpha \Delta y_{k}-\eta_{k}\right)-\frac{\bar{b}}{\mu_{0}}\right)-b \\
= & \left(A \boldsymbol{X}_{k}+\mu_{k}\left(y_{k}-\eta_{k}\right)-b-\frac{\mu_{k}}{\mu_{0}} \bar{b}\right)+\alpha\left(A \boldsymbol{\Delta} \boldsymbol{X}_{k}+\mu_{k} \Delta y_{k}\right) \\
& +\left(\alpha\left(\tau_{k}-1\right) \mu_{k}+\frac{\alpha^{2}}{n}\left\langle\Delta X_{k}, \Delta Z_{k}\right\rangle\right)\left(\left(y_{k}-\eta_{k}+\alpha \Delta y_{k}\right)-\frac{\bar{b}}{\mu_{0}}\right) \\
= & \frac{\mu_{k}}{\mu_{0}} \tilde{b}_{k}+\alpha\left(b-A \boldsymbol{X}_{k}-\tau_{k} \mu_{k}\left(\left(y_{k}-\eta_{k}\right)-\frac{\bar{b}}{\mu_{0}}\right)\right. \\
& \left.+\epsilon_{p, k}+\mu_{k}\left(\left(y_{k}-\eta_{k}\right)-\frac{\bar{b}}{\mu_{0}}\right)-\mu_{k}\left(\left(y_{k}-\eta_{k}\right)-\frac{\bar{b}}{\mu_{0}}\right)\right) \\
& +\left(\alpha\left(\tau_{k}-1\right) \mu_{k}+\frac{\alpha^{2}}{n}\left\langle\Delta X_{k}, \Delta Z_{k}\right\rangle\right)\left(\left(y_{k}-\eta_{k}+\alpha \Delta y_{k}\right)-\frac{\bar{b}}{\mu_{0}}\right) .
\end{aligned}
$$

where we used the definition of $\tilde{b}_{k}$ in the neighbourhood conditions in (4.19), and the second block equation in (4.21). By using again the neighbourhood conditions, and then by deleting the opposite terms in the previous equation, we obtain:

$$
\begin{align*}
\tilde{r}_{p}(\alpha)= & (1-\alpha) \frac{\mu_{k}}{\mu_{0}} \tilde{b}_{k}+\alpha \epsilon_{p, k}+\alpha^{2}\left(\tau_{k}-1\right) \mu_{k} \Delta y_{k} \\
& +\frac{\alpha^{2}}{n}\left\langle\Delta X_{k}, \Delta Z_{k}\right\rangle\left(y_{k}-\eta_{k}+\alpha \Delta y_{k}-\frac{\bar{b}}{\mu_{0}}\right) . \tag{4.43}
\end{align*}
$$

Similarly, we can show that:

$$
\begin{align*}
\tilde{\boldsymbol{R}}_{d}(\alpha)= & (1-\alpha) \frac{\mu_{k}}{\mu_{0}} \tilde{\boldsymbol{C}}_{k}+\alpha \mathbf{E}_{d, k}-\alpha^{2}\left(\tau_{k}-1\right) \mu_{k} \boldsymbol{\Delta} \boldsymbol{X}_{k} \\
& -\frac{\alpha^{2}}{n}\left\langle\Delta X_{k}, \Delta Z_{k}\right\rangle\left(\boldsymbol{X}_{k}-\mathbf{\Xi}_{k}+\alpha \boldsymbol{\Delta} \boldsymbol{X}_{k}+\frac{1}{\mu_{0}} \overline{\boldsymbol{C}}\right) \tag{4.44}
\end{align*}
$$

Recall (Lemma 4.4.8) that $\left\langle\Delta X_{k}, \Delta Z_{k}\right\rangle \leq K_{\Delta}^{2} n^{4} \mu_{k}$, and define the following quantities

$$
\begin{align*}
\xi_{2}= & \mu_{k}\left\|\left(\Delta y_{k}, \boldsymbol{\Delta} \boldsymbol{X}_{k}\right)\right\|_{2}+K_{\Delta}^{2} n^{3} \mu_{k}\left(\left\|\left(y_{k}-\eta_{k}, \boldsymbol{X}_{k}-\boldsymbol{\Xi}_{k}\right)\right\|_{2}+\right. \\
& \left.\alpha^{*}\left\|\left(\Delta y_{k}, \boldsymbol{\Delta} \boldsymbol{X}_{k}\right)\right\|_{2}+\frac{1}{\mu_{0}}\|(\bar{b}, \overline{\boldsymbol{C}})\|_{2}\right)  \tag{4.45}\\
\xi_{\mathcal{S}}= & \mu_{k}\left\|\left(\Delta y_{k}, \boldsymbol{\Delta} \boldsymbol{X}_{k}\right)\right\|_{\mathcal{S}}+K_{\Delta}^{2} n^{3} \mu_{k}\left(\left\|\left(y_{k}-\eta_{k}, \boldsymbol{X}_{k}-\boldsymbol{\Xi}_{k}\right)\right\|_{\mathcal{S}}+\right. \\
& \left.\alpha^{*}\left\|\left(\Delta y_{k}, \boldsymbol{\Delta} \boldsymbol{X}_{k}\right)\right\|_{\mathcal{S}}+\frac{1}{\mu_{0}}\|(\bar{b}, \overline{\boldsymbol{C}})\|_{\mathcal{S}}\right),
\end{align*}
$$

where $\alpha^{*}$ is given by (4.39). Using the definition of the starting point in (4.18), as well as results in Lemmas 4.4.4, 4.4.8, we can observe that $\xi_{2}=\boldsymbol{O}\left(n^{4} \mu_{k}\right)$. On the other hand, using Assumption 4, we know that for every pair $\left(r_{1}, \boldsymbol{R}_{2}\right) \in \mathbb{R}^{m+n^{2}}$ (where $R_{2} \in \mathbb{R}^{n \times n}$ is an arbitrary matrix), if $\left\|\left(r_{1}, \boldsymbol{R}_{2}\right)\right\|_{2}=\boldsymbol{\Theta}(f(n))$, where $f(\cdot)$ is a positive polynomial function of $n$, then $\left\|\left(r_{1}, R_{2}\right)\right\|_{\mathcal{S}}=\boldsymbol{\Theta}(f(n))$. Hence, we have that $\xi_{\mathcal{S}}=\boldsymbol{O}\left(n^{4} \mu_{k}\right)$. Using the quantities in (4.45), equations (4.43), (4.44), as well as the neighbourhood conditions, we have that:

$$
\begin{aligned}
& \left\|\tilde{r}_{p}(\alpha), \tilde{\boldsymbol{R}}_{d}(\alpha)\right\|_{2} \leq(1-\alpha) K_{N} \frac{\mu_{k}}{\mu_{0}}+\alpha \mu_{k}\left\|\left(\epsilon_{p, k}, \mathbf{E}_{d, k}\right)\right\|_{2}+\alpha^{2} \mu_{k} \xi_{2} \\
& \left\|\tilde{r}_{p}(\alpha), \tilde{\boldsymbol{R}}_{d}(\alpha)\right\|_{S} \leq(1-\alpha) \gamma_{\mathcal{S}} \rho \frac{\mu_{k}}{\mu_{0}}++\alpha \mu_{k}\left\|\left(\epsilon_{p, k}, \mathbf{E}_{d, k}\right)\right\|_{\mathcal{S}}+\alpha^{2} \mu_{k} \xi_{\mathcal{S}}
\end{aligned}
$$

for all $\alpha \in\left(0, \alpha^{*}\right]$, where $\alpha^{*}$ is given by (4.39) and the error occurring from the inexact solution of (4.20), $\left(\epsilon_{p, k}, \mathrm{E}_{d, k}\right)$, satisfies (4.22). From (4.34), we know that:

$$
\mu_{k}(\alpha) \geq\left(1-\alpha\left(1-\beta_{1}\right)\right) \mu_{k}, \text { for all } \alpha \in\left(0, \alpha^{*}\right)
$$

By combining the last three inequalities, using (4.22) and setting $\beta_{1}=\frac{\tau_{\min }}{2}$, we obtain that:

$$
\left\|\tilde{r}_{p}(\alpha), \tilde{\boldsymbol{R}}_{d}(\alpha)\right\|_{2} \leq \frac{\mu_{k}(\alpha)}{\mu_{0}} K_{N}, \text { for all } \alpha \in\left(0, \min \left\{\alpha^{*}, \frac{\tau_{\min } K_{N}}{4 \xi_{2} \mu_{0}}\right\}\right]
$$

Similarly,

$$
\left\|\tilde{r}_{p}(\alpha), \tilde{\boldsymbol{R}}_{d}(\alpha)\right\|_{\mathcal{S}} \leq \frac{\mu_{k}(\alpha)}{\mu_{0}} \gamma_{\mathcal{S}} \rho, \text { for all } \alpha \in\left(0, \min \left\{\alpha^{*}, \frac{\tau_{\min } \gamma_{\mathcal{S}} \rho}{4 \xi_{\mathcal{S}} \mu_{0}}\right\}\right]
$$

Hence, we have that:

$$
\begin{equation*}
\bar{\alpha}:=\min \left\{\alpha^{*}, \frac{\tau_{\min } K_{N}}{4 \xi_{2} \mu_{0}}, \frac{\tau_{\min } \gamma_{\mathcal{S}} \rho}{4 \xi_{\mathcal{S}} \mu_{0}}\right\} . \tag{4.46}
\end{equation*}
$$

Since $\bar{\alpha}=\Omega\left(\frac{1}{n^{4}}\right)$, we know that there must exist a constant $\bar{\kappa}>0$, independent of $n, m$ and of the iteration $k$, such that $\bar{\alpha} \geq \frac{\kappa}{n^{4}}$, for all $k \geq 0$, and this completes the proof.

The following theorem summarizes our results.
Theorem 4.4.1. Given Assumptions 3, 4, the sequence $\left\{\mu_{k}\right\}$ generated by Algorithm IP-PMM-SDP converges Q-linearly to zero, and the sequences of regularized residual norms

$$
\left\{\left\|A \boldsymbol{X}_{k}+\mu_{k}\left(y_{k}-\eta_{k}\right)-b-\frac{\mu_{k}}{\mu_{0}} \bar{b}\right\|_{2}\right\} \text { and }\left\{\left\|A^{\top} y_{k}+\boldsymbol{Z}_{k}-\mu_{k}\left(\boldsymbol{X}_{k}-\boldsymbol{\Xi}_{k}\right)-\boldsymbol{C}-\frac{\mu_{k}}{\mu_{0}} \overline{\boldsymbol{C}}\right\|_{2}\right\}
$$

converge $R$-linearly to zero.
Proof. See Appendix A.3.
Theorem 4.4.2. Let $\varepsilon \in(0,1)$ be a given error tolerance. Choose a starting point for Algorithm IP-PMM-SDP as in (4.18), such that $\mu_{0} \leq \frac{K}{\varepsilon \omega}$ for some positive constants $K$, $\omega$. Given Assumptions 3 and 4, there exists an index $k_{0} \geq 0$ with:

$$
k_{0}=\boldsymbol{O}\left(n^{4}\left|\log \frac{1}{\varepsilon}\right|\right)
$$

such that the iterates $\left\{\left(X_{k}, y_{k}, Z_{k}\right)\right\}$ generated from Algorithm IP-PMM-SDP satisfy:

$$
\mu_{k} \leq \varepsilon, \quad \text { for all } k \geq k_{0} .
$$

Proof. The proof can be found in Theorem 3.3.2.
Finally, we present the global convergence guarantee of Algorithm IP-PMMSDP.

Theorem 4.4.3. Suppose that Algorithm IP-PMM-SDP terminates when a limit point is reached. Then, if Assumptions 3 and 4 hold, every limit point of the sequence $\left\{\left(X_{k}, y_{k}, Z_{k}\right)\right\}$ determines a primal-dual solution of the non-regularized pair (SDP)-(SDD).

Proof. See Appendix A.4.

### 4.5 A sufficient condition for strong duality

We now drop Assumptions 3, 4, in order to analyze the behaviour of the algorithm when solving problems that are strongly (or weakly) infeasible, problems for which strong duality does not hold (weakly feasible), or problems for which the primal or the dual solution is not attained. For a formal definition and a
comprehensive study of the previous types of problems we refer the reader to [112], and the references therein. Below we provide a well-known result, stating that strong duality holds if and only if there exists a KKT point.

Proposition 1. Let (SDP)-(SDD) be given. Then, val(SDP) $\geq \operatorname{val}(\mathrm{SDD})$, where $\operatorname{val}(\cdot)$ denotes the optimal objective value of a problem. Moreover, $\operatorname{val}(\mathrm{SDP})=$ $\operatorname{val}(\mathrm{SDD})$ and $\left(X^{*}, y^{*}, Z^{*}\right)$ is an optimal solution for (SDP)-(SDD), if and only if $\left(X^{*}, y^{*}, Z^{*}\right)$ satisfies the (KKT) optimality conditions in (4.1).

Proof. This is a well-known fact, the proof of which can be found in [160, Proposition 2.1].

Let us employ the following two premises:
Premise 3. During the iterations of Algorithm IP-PMM-SDP, the sequences $\left\{\left\|y_{k}-\eta_{k}\right\|_{2}\right\}$ and $\left\{\left\|X_{k}-\Xi_{k}\right\|_{F}\right\}$, remain bounded.

Premise 4. There does not exist a primal-dual triple, satisfying the KKT conditions in (4.1) associated with the primal-dual pair (SDP)-(SDD).

The following analysis extends the result presented in Section 3.4, and is based on the developments in [54, Sections $10 \& 11]$. In what follows, we show that Premises 3 and 4 are contradictory. In other words, if Premise 4 holds (which means that strong duality does not hold for the problem under consideration), then Premise 3 cannot hold, and hence Premise 3 is a sufficient condition for strong duality (and its negation is a necessary condition for Premise 4). We show that if Premise 3 holds, then the algorithm converges to an optimal solution. If not, however, it does not necessarily mean that the problem under consideration is infeasible. For example, this could happen if either (SDP) or (SDD) is strongly infeasible, weakly infeasible, and in some cases even if either of the problems is weakly feasible (e.g. see $[112,160]$ ). As we discuss later, the knowledge that Premise 3 does not hold could be useful in detecting pathological problems.

Lemma 4.5.1. Given Premise 3, and by assuming that $\left\langle X_{k}, Z_{k}\right\rangle>\varepsilon$, for some $\varepsilon>0$, for all iterations $k$ of Algorithm IP-PMM-SDP, the Newton direction produced by (4.21) is uniformly bounded by a constant dependent only on $n$ and/or $m$.

Proof. The proof is omitted since it follows exactly the developments in [54, Lemma 10.1]. We notice that the regularization terms (blocks $(1,1)$ and $(2,2)$ in the Jacobian matrix in (4.21)) depend on $\mu_{k}$ which by assumption is always bounded away from zero: $\mu_{k} \geq \frac{\epsilon}{n}$.

In the following lemma, we prove by contradiction that the parameter $\mu_{k}$ of Algorithm IP-PMM-SDP converges to zero, given that Premise 3 holds.

Lemma 4.5.2. Given Premise 3, and a sequence $\left(X_{k}, y_{k}, Z_{k}\right) \in \mathcal{N}_{\mu_{k}}\left(\Xi_{k}, \eta_{k}\right)$ produced by Algorithm IP-PMM-SDP, the sequence $\left\{\mu_{k}\right\}$ converges to zero.

Proof. Assume, by virtue of contradiction, that $\mu_{k}>\varepsilon>0$, for all $k \geq 0$. Then, we know (from Lemma 4.5.1) that the Newton direction obtained by the algorithm at every iteration, after solving (4.21), will be uniformly bounded by a constant dependent only on $n$, that is, there exists a positive constant $K^{\dagger}$, such that $\left\|\left(\Delta X_{k}, \Delta y_{k}, \Delta Z_{k}\right)\right\|_{2} \leq K^{\dagger}$. We define $\tilde{r}_{p}(\alpha)$ and $\tilde{\boldsymbol{R}}_{d}(\alpha)$ as in (4.40) and (4.41), respectively, for which we know that equalities (4.43) and (4.44) hold, respectively. Take any $k \geq 0$ and define the following functions:

$$
\begin{aligned}
f_{1}(\alpha) & :=\left\langle X_{k}(\alpha), Z_{k}(\alpha)\right\rangle-\left(1-\alpha\left(1-\frac{\tau_{\min }}{2}\right)\right)\left\langle X_{k}, Z_{k}\right\rangle, \\
f_{2}(\alpha) & :=\gamma_{\mu} \mu_{k}(\alpha)-\left\|H_{P_{k}(\alpha)}\left(X_{k}(\alpha) Z_{k}(\alpha)\right)-\mu_{k}(\alpha)\right\|_{F} \\
f_{3}(\alpha) & :=(1-0.01 \alpha)\left\langle X_{k}, Z_{k}\right\rangle-\left\langle X_{k}(\alpha), Z_{k}(\alpha)\right\rangle, \\
g_{2}(\alpha) & :=\frac{\mu_{k}(\alpha)}{\mu_{0}} K_{N}-\left\|\left(\tilde{r}_{p}(\alpha), \tilde{\boldsymbol{R}}_{d}(\alpha)\right)\right\|_{2},
\end{aligned}
$$

where $\mu_{k}(\alpha)=\frac{\left\langle X_{k}+\alpha \Delta X_{k}, Z_{k}+\alpha \Delta Z_{k}\right\rangle}{n},\left(X_{k}(\alpha), y_{k}(\alpha), Z_{k}(\alpha)\right)=\left(X_{k}+\alpha \Delta X_{k}, y_{k}+\right.$ $\left.\alpha \Delta y_{k}, Z_{k}+\alpha \Delta Z_{k}\right)$. We would like to show that there exists $\alpha^{*}>0$, such that:

$$
f_{1}(\alpha) \geq 0, \quad f_{2}(\alpha) \geq 0, \quad f_{3}(\alpha) \geq 0, \quad g_{2}(\alpha) \geq 0, \text { for all } \alpha \in\left(0, \alpha^{*}\right] .
$$

These conditions model the requirement that the next iteration of Algorithm IP-PMM-SDP must lie in the updated neighbourhood $\mathcal{N}_{\mu_{k+1}}\left(\Xi_{k}, \eta_{k}\right)$ (again, notice that the restriction with respect to the semi-norm defined in (4.17) is not required here, and indeed it cannot be incorporated unless rank $(A)=m$ ). Since Algorithm IP-PMM-SDP updates the parameters $\eta_{k}, \Xi_{k}$ only if the selected new iterate belongs to the new neighbourhood, defined using the updated parameters (again, ignoring the restrictions with respect to the semi-norm), it suffices to show that $\left(X_{k+1}, y_{k+1}, Z_{k+1}\right) \in \mathcal{N}_{\mu_{k+1}}\left(\Xi_{k}, \eta_{k}\right)$.

Proving the existence of $\alpha^{*}>0$, such that each of the aforementioned functions is positive, follows exactly the developments in Lemma 4.4.9, with the only difference being that the bounds on the directions are not explicitly specified in this case. Using the same methodology as in Lemma 4.4.9, while keeping in mind our assumption, namely $\left\langle X_{k}, Z_{k}\right\rangle>\varepsilon$, we can show that:

$$
\begin{equation*}
\alpha^{*}:=\min \left\{1, \frac{\tau_{\min } \epsilon}{2\left(K^{\dagger}\right)^{2}}, \frac{\left(1-\gamma_{\mu}\right) \tau_{\min } \bar{\epsilon}}{2\left(K^{\dagger}\right)^{2}}, \frac{0.49 \epsilon}{2\left(K^{\dagger}\right)^{2}}, \frac{\tau_{\min } K_{N} \epsilon}{4 \mu_{0}\left(\xi_{2}\right)}\right\} \tag{4.47}
\end{equation*}
$$

where $\xi_{2}$ is a bounded constant, defined as in (4.45), and dependent on $K^{\dagger}$. However, using the inequality:

$$
\mu_{k+1} \leq(1-0.01 \alpha) \mu_{k}, \text { for all } \alpha \in\left[0, \alpha^{*}\right]
$$

we get that $\mu_{k} \rightarrow 0$, which contradicts our assumption that $\mu_{k}>\varepsilon, \forall k \geq 0$, and completes the proof.

Finally, using the following theorem, we derive a necessary condition for lack of strong duality.

Theorem 4.5.1. Given Premise 4, i.e. there does not exist a KKT triple for the pair (SDP)-(SDD), then Premise 3 fails to hold.

Proof. See Appendix A.5.
In the previous theorem, we proved that the negation of Premise 3 is a necessary condition for Premise 4. In order to obtain a more reliable algorithmic test for lack of strong duality, we can employ a similar heuristic as the one presented in the end of Section 3.4.

### 4.6 Conclusions

In this chapter we developed and analyzed an interior point-proximal method of multipliers, suitable for solving linear positive semi-definite programs, without requiring the exact solution of the associated Newton systems. By generalizing appropriately some previous results on convex quadratic programming, we show that IP-PMM inherits the polynomial complexity of standard non-regularized IPM schemes when applied to SDP problems, under standard assumptions, while having to approximately solve better-conditioned Newton systems, compared to their non-regularized counterparts. Furthermore, we provide a tuning for the proximal penalty parameters based on the well-studied barrier parameter, which can be used to guide any subsequent implementation of the method. Finally, we study the behaviour of the algorithm when applied to problems for which no KKT point exists, and give a necessary condition which can be used to construct detection mechanisms for identifying such pathological cases.

A future research direction would be to construct a robust and efficient implementation of the method, which should utilize some Krylov subspace solver alongside an appropriate preconditioner for the solution of the associated linear systems. Given several implementations of IP-PMM for other classes of problems appearing in this thesis as well as in the literature, we expect that the theory can successfully guide the implementation, yielding a competitive, robust, and efficient method.

## Chapter 5

## Preconditioning for Regularized IPMs

### 5.1 Introduction

In this chapter, we are concerned with applying Krylov subspace methods for the efficient solution of systems of the following form:

$$
\underbrace{\left[\begin{array}{cc}
-\left(Q+\rho I_{n}\right) & A^{\top}  \tag{5.1}\\
A & \delta I_{m}
\end{array}\right]}_{K}\left[\begin{array}{l}
\Delta x \\
\Delta y
\end{array}\right]=\left[\begin{array}{l}
\xi_{1} \\
\xi_{2}
\end{array}\right],
$$

where $A \in \mathbb{R}^{m \times n}$ (with $m \leq n$ ), $Q \succeq 0_{n, n} \in \mathbb{R}^{n \times n}$, and $\delta, \rho>0$. Such systems arise in a plethora of applications [21], which go far beyond optimization. However, in this chapter we restrict the discussion to the case of regularized systems arising in interior point methods for optimization [1, 134, 154, 171]. Due to the potential large dimensions of the systems, they are often solved by means of iterative techniques, usually from the family of Krylov subspace methods [85]. To guarantee efficiency of such methods the possibly ill-conditioned system (5.1) often needs to be appropriately preconditioned and, indeed, there exists a rich literature which addresses the issue (see the discussions in [21, 24, 25, 59, 60, 139], and the references therein).

As already stated in Chapter 1, a wide range of preconditioners have been proposed in the literature, which can be divided into symmetric (e.g. [21, 74, 126, $132,161]$ ) and non-symmetric ones (e.g. [59, 94, 100, 126, 153]). A comprehensive study on saddle point systems and their associated "optimal" preconditioners can be found in [21]. Non-symmetric preconditioners are significantly more difficult to analyze and a simple spectral analysis is not sufficient to deduce their effectiveness (see [86]). On the other hand, symmetric preconditioners are often significantly easier to analyze, and the eigenvalues of the preconditioned matrices allow one to theoretically compare different preconditioning approaches. A range of general preconditioners have been proposed for systems of the form of (5.1) arising from optimization (see [30, 40, 63, 133, 156]). However, as is the case within the field of preconditioning in general, these are typically sensitive to changes in the structure of the matrices involved, and can have substantial memory requirements. Precon-
ditioners have also been successfully devised for specific classes of optimization problems solved using similar optimization methods: applications include those arising from multicommodity network flow problems (e.g. [37]), stochastic programming problems (e.g. [36]), formulations within which the constraint matrix has a primal block-angular structure (e.g. [38]), and PDE-constrained optimization problems (e.g. [138, 140, 144]). Nevertheless, such preconditioners exploit particular structures arising from specific applications; unless there exists such a structure which hints as to the appropriate way to develop a solver, the design of bespoke preconditioners remains a challenge.

It is therefore clear that a completely robust preconditioner for convex programming does not currently exist, as available preconditioners are either problemsensitive (with a possibility of failure when problem parameters or structures are modified), or are tailored towards specific classes of problems. In this chapter, we aim to provide a first step towards the construction of generalizable preconditioners for saddle point systems arising from the application of a regularized IPM to convex programming. In this case, $Q$ represents the Hessian of the primal barrier problem's objective function (or the Hessian of the Lagrangian in the nonlinear programming case), $A$ represents the constraint matrix (or the Jacobian of the constraints in the nonlinear programming case), while $\rho$ and $\delta$ are the primal and dual regularization parameters, respectively. We note that the IPM may contribute a term to the $(1,1)$ or the $(2,2)$ block of $(5.1)$, depending on the form of the constraints and non-negativity variables. Here we assume that the term is added in the $(1,1)$ block.

We present symmetric preconditioning approaches that can be used within the MINRES [135] or the CG method [93], and we provide some spectral analysis results for the associated preconditioned systems. More specifically, we consider preconditioners which are derived by "sparsifications" of system (5.1), that is, by dropping specific entries from sparse matrices $Q$ and $A$, thus making them more sparse and hence easier to factorize. Various such approaches have been proposed to date and include: the preconditioners which exploit an early guess of a basic-nonbasic partition of variables to drop columns from $A$ [133], constraint preconditioners [25, 59, 60], inexact constraint preconditioners [24] which drop specific entries in matrices $Q$ and $A$, and of course a plethora of preconditioners which involve various levels of incomplete Cholesky factorizations of the Schur complement of the matrix in (5.1), see for example [30]. The literature on preconditioners is growing rapidly and we refer the interested reader to [21, 50, 139] and the references therein.

We consider dropping off-diagonal entries of $Q$, but restrict the elimination of entries in $A$ only to the removal of complete columns or parts of rows of it. Such a strategy guarantees that we avoid situations in which eigenvalues of the preconditioned matrix may become complex (such as those employed in [24]), which as a consequence would have required employing non-symmetric Krylov methods. In order to construct the preconditioners, we take advantage of the properties of the logarithmic barrier, that allow us to know in advance which columns of the problem matrix are important and which are less influential. Furthermore, we discuss some approaches for dealing with problems for which the constraint matrix $A$ may contain a subset of dense columns or rows. All such "sparsifica-
tions" are captured in a general result presented in Section 5.2 which provides the spectral analysis of the preconditioned Schur complement matrix. The main theorem sheds light on consequences of sparsifying rows or dropping columns of $A$, and demonstrates that the former might produce a larger number of non-unit eigenvalues.

All of the preconditioning approaches discussed are compared numerically on various real-life linear and convex quadratic programming problems. In particular, we present some numerical results on certain test problems taken from the Netlib and the Maros-Mészáros collections, and subsequently we apply the methodologies on certain $L^{1}$-regularized PDE-constrained optimization problems. We should mention that the presented preconditioners will be later utilized (in Chapter 6), when solving image reconstruction, as well as classification problems. All preconditioning approaches have been implemented within an IP-PMM framework, i.e. the polynomially convergent primal-dual regularized IPM, based on the developments in Chapters 3, 4. A robust implementation is provided.

It is worth stressing that the proposed preconditioners are general and do not assume the knowledge of special structures which might be present in matrices $Q$ and $A$ (such as block-diagonal, block-angular, network, PDE-induced, etc.). Therefore they may be applied within general-purpose IPM solvers for convex programming problems.

The rest of this chapter is organized as follows. In Section 5.2 we present some preconditioners suitable for the normal equations. Then, in Section 5.3, we adapt these preconditioners to regularized saddle point systems. Subsequently, in Section 5.4 we focus on saddle point systems arising from the applications of regularized IPMs to convex programming problems, and present some numerical results. Finally, in Section 5.5, we provide some conclusions.

### 5.2 Regularized normal equations

We begin by defining the regularized normal equations matrix (or Schur complement) $M:=A G A^{\top}+\delta I_{m} \in \mathbb{R}^{m \times m}$, corresponding to (5.1), where $G \equiv$ $\left(Q+\rho I_{n}\right)^{-1} \succ 0_{n, n}$. For the discussion of this section, we assume that $G$ (and hence $Q$ ) is a diagonal matrix. Permuting the normal equations matrix suitably (see the discussion at the end of this section), we can write:

$$
B:=\mathscr{P}_{r} A G^{\frac{1}{2}} \mathscr{P}_{c}=\left[\begin{array}{ll}
B_{11} & B_{12} \\
B_{21} & B_{22}
\end{array}\right],
$$

where $\mathscr{P}_{r}, \mathscr{P}_{c}$ are two permutation matrices, $B_{11} \in \mathbb{R}^{k_{r} \times k_{c}}, B_{12} \in \mathbb{R}^{k_{r} \times\left(n-k_{c}\right)}$, $B_{21} \in \mathbb{R}^{\left(m-k_{r}\right) \times k_{c}}$, and $B_{22} \in \mathbb{R}^{\left(m-k_{r}\right) \times\left(n-k_{c}\right)}$, with $0 \leq k_{r} \leq m$ and $0 \leq k_{c} \leq n$. Let us further introduce the following notation:

$$
\mathscr{P}_{r} M \mathscr{P}_{r}^{\top} \equiv\left[\begin{array}{ll}
M_{11} & M_{21}^{\top} \\
M_{21} & M_{22}
\end{array}\right],
$$

where $M_{11}, M_{21}$, and $M_{22}$ are defined as:

$$
\begin{aligned}
& M_{11}:=B_{11} B_{11}^{\top}+B_{12} B_{12}^{\top}+\delta I_{k_{r}} \in \mathbb{R}^{k_{r} \times k_{r}}, \\
& M_{21}:=B_{21} B_{11}^{\top}+B_{22} B_{12}^{\top} \in \mathbb{R}^{\left(m-k_{r}\right) \times k_{r}}, \\
& M_{22}:=B_{21} B_{21}^{\top}+B_{22} B_{22}^{\top}+\delta I_{m-k_{r}} \in \mathbb{R}^{\left(m-k_{r}\right) \times\left(m-k_{r}\right)} .
\end{aligned}
$$

In what follows, we present two preconditioning strategies for systems involving matrix $M$. Both approaches are based on a sparsification of matrix $M$. The first approach relies on a Cholesky decomposition of a sparsified matrix, while the second approach is based on a $L D L^{\top}$ decomposition of a sparsified augmented system matrix, which is used to implicitly derive a preconditioner for $M$.

### 5.2.1 A Cholesky-based preconditioner

Our first proposal is to consider preconditioning $\mathscr{P}_{r} M \mathscr{P}_{r}^{\top}$ with the following matrix:

$$
P_{N E}:=\left[\begin{array}{cc}
M_{11} & 0_{k_{r},\left(m-k_{r}\right)}  \tag{5.2}\\
0_{\left(m-k_{r}\right), k_{r}} & \widetilde{M}_{22}
\end{array}\right], \quad \widetilde{M}_{22}=M_{22}-B_{21} B_{21}^{\top} .
$$

In the following theorem, we will analyze the spectrum of the preconditioned matrix $P_{N E}^{-1} \mathscr{P}_{r} M \mathscr{P}_{r}^{\top}$, with respect to the spectrum of the associated matrices. Before proceeding with the theorem, we should mention that dropping $M_{21}$ and $M_{21}^{\top}$ from $M$, produces some small eigenvalue outliers (smaller than or equal to 1 ), as well as certain eigenvalues in the interval ( 1,2 ), while dropping $B_{21} B_{21}^{\top}$ from the $(2,2)$ block of $M$ produces some large eigenvalue outliers (larger than or equal to 1). For ease of presentation, the two eigenvalue intervals are given separately.
Theorem 5.2.1. The preconditioned matrix $P_{N E}^{-1} \mathscr{P}_{r} M \mathscr{P}_{r}^{\top}$ has at least $\max \{m-$ $\left.\left(2 k_{r}+k_{c}\right), 0\right\}$ eigenvalues equal to 1 . All remaining eigenvalues lie in $I_{1} \cup I_{2}$, where

$$
I_{1}:=\left[\frac{\delta}{\delta+\sigma_{\max }^{2}(B)}, 1\right], \quad I_{2}:=\left[1,2+\frac{\lambda_{\max }\left(B_{21} B_{21}^{\top}\right)}{\delta+\lambda_{\min }\left(B_{22} B_{22}^{\top}\right)}\right]
$$

Proof. Firstly, given an arbitrary eigenvalue $\lambda$ (which must be positive since $P_{N E} \succ 0_{m, m}$ and $M \succ 0_{m, m}$ ) corresponding to a unit eigenvector $v$, let us write the generalized eigenproblem as:

$$
\left[\begin{array}{ll}
M_{11} & M_{21}^{\top}  \tag{5.3}\\
M_{21} & M_{22}
\end{array}\right]\left[\begin{array}{l}
v_{1} \\
v_{2}
\end{array}\right]=\lambda\left[\begin{array}{l}
M_{11} v_{1} \\
\widetilde{M}_{22} v_{2}
\end{array}\right]
$$

We separate the analysis into two cases.
Case 1: Let $v_{2} \in \operatorname{Null}\left(M_{21}^{\top}\right)$. Firstly, we notice that:

$$
\operatorname{dim}\left(\operatorname{Null}\left(M_{21}^{\top}\right)\right)=\left(m-k_{r}\right)-\operatorname{rank}\left(M_{21}^{\top}\right) \geq \max \left\{m-2 k_{r}, 0\right\} .
$$

Two sub-cases arise here. For the first sub-case, we notice that if $v_{1} \neq 0_{k_{r}}$, then
from positive definiteness of $M_{11}$, combined with the first block equation of (5.3), we obtain that $\lambda=1$. In turn, we claim that this implies that $v_{2} \in \operatorname{Null}\left(B_{21} B_{21}^{\top}\right)$ and $v_{1} \in \operatorname{Null}\left(M_{21}\right)$. To see this, assume that $v_{2} \notin \operatorname{Null}\left(B_{21} B_{21}^{\top}\right)$. Then from the second block equation of (5.3) we obtain:

$$
M_{21} v_{1}+M_{22} v_{2}=\widetilde{M}_{22} v_{2} \Rightarrow M_{21} v_{1}=-B_{21} B_{21}^{\top} v_{2},
$$

where we used the definition of $\widetilde{M}_{22}$. If $v_{2} \notin \operatorname{Null}\left(B_{21} B_{21}^{\top}\right)$, this implies that $v_{2}^{\top} B_{21} B_{21}^{\top} v_{2}>0$. The previous equation then yields that

$$
v_{2}^{\top} M_{21} v_{1}=-v_{2}^{\top} B_{21} B_{21}^{\top} v_{2} \Rightarrow 0=-v_{2}^{\top} B_{21} B_{21}^{\top} v_{2}<0,
$$

which follows from the base assumption (i.e. $v_{2} \in \operatorname{Null}\left(M_{21}^{\top}\right)$ ), and results in a contradiction. Hence, $v_{2} \in \operatorname{Null}\left(B_{21} B_{21}^{\top}\right)$. On the other hand, if $v_{1} \notin \operatorname{Null}\left(M_{21}\right)$ then the second block equation yields directly a contradiction, since we have shown that $v_{2} \in \operatorname{Null}\left(B_{21} B_{21}^{\top}\right)$.

Next we consider the second sub-case, i.e. $v_{1}=0_{k_{r}}$. Combined with our base assumption, the first block equation of (5.3) becomes redundant. From the second block equation of the eigenproblem, and using $v_{1}=0_{k_{r}}$, we obtain:

$$
\begin{align*}
v_{2}^{\top} M_{22} v_{2} & =\lambda v_{2}^{\top} \widetilde{M}_{22} v_{2} \Rightarrow \\
v_{2}^{\top}\left(\widetilde{M}_{22}+B_{21} B_{21}^{\top}\right) v_{2} & =\lambda v_{2}^{\top} \widetilde{M}_{22} v_{2} . \tag{5.4}
\end{align*}
$$

Hence we have that:

$$
\lambda=1+\frac{v_{2}^{\top}\left(B_{21} B_{21}^{\top}\right) v_{2}}{v_{2}^{\top}\left(\widetilde{M}_{22}\right) v_{2}} \leq 1+\frac{\lambda_{\max }\left(B_{21} B_{21}^{\top}\right)}{\delta+\lambda_{\min }\left(B_{22} B_{22}^{\top}\right)} .
$$

All eigenvalues in this case can be bounded by the previous inequality and there will be at most $\operatorname{rank}\left(B_{21} B_{21}^{\top}\right)$ non-unit eigenvalues. On the other hand, if $v_{2} \in \operatorname{Null}\left(B_{21} B_{21}^{\top}\right)$, then trivially $\lambda=1$. This concludes the first case.

Case 2: In this case, we assume that $v_{2} \notin \operatorname{Null}\left(M_{21}^{\top}\right)$. In what follows we assume $\lambda \neq 1$ (noting that $\lambda=1$ would only occur if $v_{1} \in \operatorname{Null}\left(M_{21}\right)$ and $\left.v_{2} \in \operatorname{Null}\left(B_{21} B_{21}^{\top}\right)\right)$, and there are at most $2 k_{r}$ such eigenvalues. Given the previous assumption, and using the first block equation in (5.3), we obtain:

$$
v_{1}=\frac{1}{\lambda-1} M_{11}^{-1} M_{21}^{\top} v_{2} .
$$

Substituting the previous to the second block equation of (5.3) yields the following generalized eigenproblem:

$$
\begin{equation*}
\left(M_{21} M_{11}^{-1} M_{21}^{\top}+(\lambda-1) B_{21} B_{21}^{\top}\right) v_{2}=(\lambda-1)^{2} \widetilde{M}_{22} v_{2} \tag{5.5}
\end{equation*}
$$

where we used the definitions of $\widetilde{M}_{22}$ and $M_{22}$. Multiplying (5.5) by $v_{2}^{\top}$ and rearranging yields the following quadratic algebraic equation that $\lambda$ must satisfy in this case:

$$
\begin{equation*}
\lambda^{2}+\beta \lambda+\gamma=0, \tag{5.6}
\end{equation*}
$$

where

$$
\beta:=-2-\frac{v_{2}^{\top} B_{21} B_{21}^{\top} v_{2}}{v_{2}^{\top} \widetilde{M}_{22} v_{2}},
$$

and

$$
\gamma:=1-\frac{v_{2}^{\top}\left(M_{21} M_{11}^{-1} M_{21}^{\top}-B_{21} B_{21}^{\top}\right) v_{2}}{v_{2}^{\top} \widetilde{M}_{22} v_{2}}
$$

Let us notice that the smallest eigenvalue is at least as large as $\frac{\delta}{\delta+\sigma_{\max }(B)}$. This follows from positive definiteness of $P_{N E}$ and $M$, and the bound can be deduced by inspecting the Rayleigh quotients. Hence, we would like to find an upper bound for the largest eigenvalue. To that end, notice that:

$$
\gamma=\frac{v_{2}^{\top}\left(M_{22}-M_{21} M_{11}^{-1} M_{21}^{\top}\right) v_{2}}{v_{2}^{\top} \widetilde{M}_{22} v_{2}}
$$

which follows from the definition of $\widetilde{M}_{22}$. Positive definiteness of $M$ then implies that $\gamma>0$. From the last relation we also have that:

$$
0<\gamma \leq \frac{v_{2}^{\top} M_{22} v_{2}}{v_{2}^{\top} \widetilde{M}_{22} v_{2}} \leq 1+\frac{\lambda_{\max }\left(B_{21} B_{21}^{\top}\right)}{\delta+\lambda_{\min }\left(B_{22} B_{22}^{\top}\right)}=: \gamma_{u}
$$

Furthermore, $\beta_{l}:=-\left(2+\frac{\lambda_{\max }\left(B_{21} B_{21}^{\top}\right)}{\delta+\lambda_{\min }\left(B_{22} B_{22}\right)}\right) \leq \beta \leq-2$. From the previous, one can also observe that $\gamma \leq-\beta-1$.

Returning to (5.6), we first consider the following solution:

$$
\lambda_{-}=\frac{1}{2}\left(-\beta-\sqrt{\beta^{2}-4 \gamma}\right)
$$

Obviously, $\beta^{2}-4 \gamma$ is always greater than 0 , since $-\beta \geq 2$ and $\gamma \leq-\beta-1$. Next, we notice that the expression for $\lambda_{-}$is increasing with respect to $\gamma$. We omit finding a lower bound for $\lambda_{-}$since this was established earlier. For the upper bound, we use the fact that $\gamma \leq-\beta-1$, to obtain:

$$
\lambda_{-} \leq \frac{1}{2}\left(-\beta-\sqrt{\beta^{2}+4(\beta+1)}\right)=\frac{1}{2}(|\beta|-|\beta+2|)=1
$$

since $\beta \leq-2$ (also, in the beginning of this case, we have treated $\lambda_{-}=1$ separately).

Finally, we consider the symmetric equation, which reads:

$$
\lambda_{+}=\frac{1}{2}\left(-\beta+\sqrt{\beta^{2}-4 \gamma}\right)
$$

Firstly, we can easily notice that $\lambda_{+}>1$. Subsequently, upon noticing that $\lambda_{+}$is decreasing with respect to $\gamma$, we can obtain the following obvious bound:

$$
\lambda_{+} \leq|\beta| \leq-\beta_{l}
$$

To conclude the proof, we observe that dropping $M_{21}$ and $M_{21}^{\top}$ yields at most $k_{r}+\operatorname{rank}\left(M_{21}^{\top}\right) \leq 2 k_{r}$ eigenvalue outliers. Similarly, dropping $B_{21} B_{21}^{\top}$ from the $(2,2)$ block of $M$ yields at most $\operatorname{rank}\left(B_{21}\right) \leq k_{c}$ eigenvalue outliers. Hence, there will be at least max $\left\{m-\left(2 k_{r}+k_{c}\right), 0\right\}$ eigenvalues of the preconditioned matrix equal to 1 , and this concludes the proof.

Remark 4. As we will discuss later, a case of interest would be to only drop certain $k_{c}$ columns of $A$. In that case, we have $I_{1}=\{1\}$ and $I_{2}$ as in Theorem 5.2.1, and at most $k_{c}$ eigenvalue outliers. A similar observation can be made for the case of only sparsifying certain $\left(k_{r}\right)$ rows of $A$. In that case, it can easily be shown that $I_{2}=[1,2]$ (with $I_{1}$ as in in Theorem 5.2.1), and there will be at most $2 k_{r}$ non-unit eigenvalues. Notice that dropping columns is expected to be more useful in general, since this results in fewer outliers and possibly in greater gains (either in terms of processing time or memory requirements). However, in certain special applications one has to resort to sparsifying problematic rows. The suitability of each such choice should depend on the problem under consideration.

Remark 5. Now that we have presented the spectral properties of the preconditioned system, let us discuss the use of such a preconditioning strategy. In practice, one often has to solve regularized normal equations arising from a plethora of problems, where $A$ (and B) is usually a sparse matrix.

- Firstly, it is common in many application areas to have a small number of columns or rows of A that are dense. Such columns (or rows) could pose significant difficulties as they produce dense factors when one tries to factorize the normal equations (e.g. using a Cholesky decomposition). This is especially the case for dense columns, which can produce up to $(m-p) \times(m-p)$ fill-in, where $p$ is the pivot order of the respective dense column (we refer the reader to the discussion in [2, Section 4]). The use of a preconditioner like the one defined in (5.2) serves the purpose of dropping (sparsifying, respectively) such columns (rows, respectively), thus making the Cholesky factors of $P_{N E}$ significantly more sparse. For example, we may find two permutation matrices $\mathscr{P}_{r}, \mathscr{P}_{c}$ which sort the rows and columns, respectively, of $A$ in descending order of their number of non-zeros, and write $\widehat{A}=\mathscr{P}_{r} A \mathscr{P}_{c}$. Then, the resulting normal equations read as $\mathcal{P}_{r}^{\top} B B^{\top} \mathcal{P}_{r}+\delta I_{m}$. As long as the number of columns or rows dropped is low (which is observed in several applications), the number of outliers produced by this dropping strategy is manageable. While some of these outliers will be dangerously close to zero (given that the regularization parameter $\delta>0$ is small), they can be dealt with efficiently.
- Secondly, it often happens in optimization, and especially when solving systems arising from the application of an interior point method, to have certain diagonal elements of $G$ that are very small. In view of this property, and given the bound presented in Theorem 5.2.1, we can observe that dropping all columns corresponding to small diagonal elements in $G$ results in manageable and not too sizeable outliers. Such a preconditioner was proposed in [23], and arises as a special case of $P_{N E}$ in (5.2), by choosing $k_{r}=0$ and a suitable permutation matrix $\mathscr{P}_{c}$.

Remark 6. Finally, we should note that the preconditioner in (5.2) would still be meaningful if $Q$ had the form:

$$
\mathscr{P}_{c} Q \mathscr{P}_{c}^{\top}=\left[\begin{array}{cc}
Q_{1} & 0_{k_{c},\left(n-k_{c}\right)} \\
0_{\left(n-k_{c}\right), k_{c}} & Q_{2}
\end{array}\right] .
$$

Obviously such a structure is not expected to be present in general. However, this suggests an extension of the column dropping strategy to the case of a non-diagonal matrix $Q$. In particular, when a column is dropped, we could drop the respective row and column of matrix $Q$, keeping only its diagonal. We will return to this observation later. We should note that a similar methodology has been employed and analyzed before in Chapter 2 in the context of non-diagonal regularization.

## Convergence guarantees

The idea of splitting the matrix $A$ into sparse and dense columns is not new. Indeed, there is an extensive literature for the solution of least-squares problems, where $A^{\top}$ is split based on the density of its columns (notice that in the leastsquares literature it is assumed that $n \leq m$, and hence one resorts to the normal equations which result from pivoting the $(2,2)$ block in (5.1)). In what follows we briefly discuss a widely-used approach that is employed to deal with the dense and the sparse columns separately, focusing on the case where $m \leq n$, inspired by the developments in [158]. For an in-depth analysis, we refer the interested reader to [158] and the references therein.

Assuming that the number of dense columns $k_{c}$ of $A$ is large, but significantly smaller than $n-k_{c}$, we expect that as the IPM progresses, and the conditioning of the associated Newton systems is deteriorating, the approximation in (5.2) of $M$ could eventually be insufficient to ensure the convergence of the Krylov method. In that case, if the number of Krylov iterations required in the previous IPM iteration surpasses a certain threshold, we could switch to the following methodology (to ensure the method remains efficient without requiring additional memory). Using the splitting $B:=A G^{\frac{1}{2}} \mathscr{P}_{c}=\left[\begin{array}{ll}B_{1} & B_{2}\end{array}\right]\left(B_{1}\right.$ is dense and $B_{2}$ is sparse), we can write the normal equations as:

$$
\left(B_{1} B_{1}^{\top}+B_{2} B_{2}^{\top}+\delta I_{m}\right) y=\left(B_{1} B_{1}^{\top}+L_{2}^{\top} L_{2}\right) y=\xi
$$

where $L_{2}$ is the Cholesky decomposition of $B_{2} B_{2}^{\top}+\delta I_{m}$, and $\xi$ is the right hand side. Define $C_{1}=L_{2}^{-1} B_{1}, \widehat{\xi}=L_{2}^{-1} \xi$, and $\widehat{y}=L_{2}^{\top} y$. Then, the normal equations can be written as:

$$
\left(I_{m}+C_{1} C_{1}^{\top}\right) \widehat{y}=\widehat{\xi}
$$

Using the Woodbury formula (see [180]), we can retrieve $\widehat{y}$ as:

$$
\widehat{y}=\widehat{\xi}-C_{1}\left(I_{k_{c}}+C_{1}^{\top} C_{1}\right)^{-1} C_{1}^{\top} \widehat{\xi}
$$

Subsequently, $y$ is retrieved by backward substitution using the Cholesky factor $L_{2}$. Assuming that $k_{c} \ll n-k_{c}$, we expect that the cost of inverting $I_{k_{c}}+C_{1}^{\top} C_{1}$ is reasonable (and we can employ dense linear algebra operations to avoid using a sparse representation, if necessary). The main computational bottleneck of this
approach lies with forming $C_{1}$, since this requires $k_{c}$ backward substitutions. The cost of this can help us determine the threshold on the Krylov iterations that should force the algorithm to switch to this strategy.

Finally, note that all of the previous steps can be further approximated by using $\widetilde{L}_{2}$, and $\widetilde{L}_{1}$ such that $\widetilde{L}_{2} \widetilde{L}_{2}^{\top} \approx B_{2} B_{2}^{\top}+\delta I_{m}$ and $\widetilde{L}_{1} \widetilde{L}_{1}^{\top} \approx I_{m}+C_{1} C_{1}^{\top}$. In that case, the previous process (involving the inexact factorizations) is used as preconditioner for an inner layer Krylov method. This methodology has been used to solve least-squares problems in [158], and hence is not treated any further in this thesis.

### 5.2.2 $\quad \mathrm{A} \mathrm{LDL}^{\top}$-based preconditioner

Next, we present an alternative to the preconditioner in (5.2). More specifically, let us divide the columns of matrix $A$ into two mutually exclusive sets $\mathcal{B}$ and $\mathcal{N}$. Then, using the column-dropping strategy presented in the previous section, and assuming that the variables corresponding to $\mathcal{N}$ are not especially important (e.g. they correspond to the smallest diagonal elements of $G$, hence an IPM for linear programming considers them inactive, i.e. $x^{\mathcal{N}} \rightarrow 0_{|\mathcal{N}|}$ and $\left.G^{(\mathcal{N}, \mathcal{N})} \rightarrow \rho_{k} I_{|\mathcal{N}|}\right)$, we propose approximating the normal equations matrix $M$ by the following preconditioner:

$$
\begin{equation*}
\widehat{P}_{N E}=A^{(:, \mathcal{B})} G^{(\mathcal{B}, \mathcal{B})}\left(A^{(:, \mathcal{B})}\right)^{\top}+\delta I_{m}, \tag{5.7}
\end{equation*}
$$

which is the preconditioner proposed in [23]. Notice that our assumption that $Q$ (and hence $G$ ) is diagonal implies that $\left(G^{(\mathcal{B}, \mathcal{B})}\right)^{-1}=Q^{(\mathcal{B}, \mathcal{B})}+\rho I_{|\mathcal{B}|}$. Given our previous discussion, we would like to avoid inverting this preconditioner by means of a Cholesky decomposition, as a single dense column in $\mathcal{B}$ could result in dense Cholesky factors. Instead, we form an appropriate saddle point system to compute the action of the approximated normal equations. More specifically, given an arbitrary vector $y \in \mathbb{R}^{m}$, instead of computing $\widehat{P}_{N E}^{-1} y$ using a Cholesky decomposition, we can compute

$$
\underbrace{\left[\begin{array}{cc}
-\left(Q^{(\mathcal{B}, \mathcal{B})}+\rho I_{|\mathcal{B}|}\right) & \left(A^{(:, \mathcal{B})}\right)^{\top}  \tag{5.8}\\
A^{(,, \mathcal{B})} & \delta I_{m}
\end{array}\right]}_{\widetilde{P}_{N E}}\left[\begin{array}{l}
w_{1} \\
w_{2}
\end{array}\right]=\left[\begin{array}{c}
0_{|\mathcal{B}|} \\
y
\end{array}\right],
$$

by means of a $L D L^{\top}$ decomposition of the previous saddle point matrix. Then, we notice that returning $w_{2}$ is equivalent to computing $\widehat{P}_{N E}^{-1} y$.

Following the discussion in [2, Section 4], we know that using $L D L^{\top}$ to factorize the matrix in (5.8) can result in significant memory savings compared to the Cholesky decomposition of $\widehat{P}_{N E}$. Notice that in view of the regularized nature of the systems under consideration (indeed, we have assumed that $G$ is positive definite), we can use the result in [171], stating that matrices like the one in (5.8) are quasi-definite; any symmetric permutation of such matrices admits a $L D L^{\top}$ decomposition.

While this approach might seem expensive, it can provide significant time and
memory savings, especially in cases where $A^{(:, \mathcal{B})}$ contains dense columns. In the previous section we discussed a strategy for alleviating this issue, noting however that such a strategy can only be used to deal with a small number of dense columns. On the contrary, if we have a sizeable subset of the columns of $A^{(:, \mathcal{B})}$ that are dense, we could delay their pivot order within the $L D L^{\top}$, thus significantly reducing the overall fill-in of the decomposition factors, without introducing any eigenvalue outliers in the preconditioned system. Of course, finding the optimal permutation for the $L D L^{\top}$ decomposition is a NP-hard problem, however, there have been developed several effective permutation heuristics tailored to such symmetric decompositions. In particular, the pivots are computed dynamically to ensure both stability and sparsity. In view of the previous, the preconditioner based on solving (5.8) is expected to be more stable than its counterpart based on the Cholesky decomposition. Finally, difficulties arising from dense rows or in general "problematic" rows can also be alleviated using a heuristic proposed in [115].

As in Section 5.2.1, there is an extensive literature for the solution of saddlepoint systems where $A^{\top}$ is split based on the density of its columns or rows. Using a suitable symmetric decomposition one can separate the sparse from the dense linear algebra in order to save both memory and time. We refer the interested reader to [159] and the references therein.

### 5.2.3 BFGS-like low-rank updates of the preconditioner

In certain cases, it might be beneficial to improve the spectral properties of the preconditioned system by attempting to capture and correct some of its eigenvalue outliers. Thus, we briefly mention a possible way of achieving this by means of BFGS-like low-rank updates. The potential usefulness of this is later verified numerically.

Given a rectangular (tall) matrix $V \in \mathbb{R}^{m \times p}$ with maximum column rank, it is possible to define a generalized block-tuned preconditioner $P$ satisfying the property

$$
P^{-1} \mathscr{P}_{r} M \mathscr{P}_{r}^{\top} V=\nu V
$$

so that the columns of $V$ become eigenvectors of the preconditioned matrix corresponding to the eigenvalue $\nu$. A way to construct $P$ (or its explicit inverse) is suggested by the BFGS-based preconditioners used e.g. in [22] for accelerating Newton linear systems or analyzed in [117] for general sequences of linear systems, that is

$$
P^{-1}=\nu V \Pi V^{\top}+\left(I_{m}-V \Pi V^{\top} \mathscr{P}_{r} M \mathscr{P}_{r}^{\top}\right) P_{N E}^{-1}\left(I_{m}-\mathscr{P}_{r} M \mathscr{P}_{r}^{\top} V \Pi V^{\top}\right)
$$

with $\Pi=\left(V^{\top} \mathscr{P}_{r} M \mathscr{P}_{r}^{\top} V\right)^{-1}$. Notice that if the columns of $V$ would be chosen as e.g. the $p$ exact rightmost eigenvectors of $P_{N E}^{-1} \mathscr{P}_{r} M \mathscr{P}_{r}^{\top}$ (corresponding to the $p$ largest eigenvalues) then all the other eigenpairs,

$$
\left(\lambda_{1}, z_{1}\right), \ldots,\left(\lambda_{m-p}, z_{m-p}\right)
$$

of the new preconditioned matrix $P^{-1} \mathscr{P}_{r} M \mathscr{P}_{r}^{\top}$ would remain unchanged as stated in the following theorem.

Theorem 5.2.2. If the columns of $V$ are the exact rightmost eigenvectors of $P_{N E}^{-1} \mathscr{P}_{r} M \mathscr{P}_{r}^{\top}$ then for every $j=1, \ldots m-p$ there holds

$$
P^{-1} \mathscr{P}_{r} M \mathscr{P}_{r}^{\top} z_{j}=P_{N E}^{-1} \mathscr{P}_{r} M \mathscr{P}_{r}^{\top} z_{j}=\lambda_{j} z_{j} .
$$

Proof. The eigenvectors of the symmetric generalized eigenproblem $\mathscr{P}_{r} M \mathscr{P}_{r}^{\top} x=$ $\lambda P_{N E} x$ form a $P_{N E}$-orthonormal basis and therefore $V^{T} P_{N E} z_{j}=V^{\top} \mathscr{P}_{r} M \mathscr{P}_{r}^{\top} z_{j}=$ $0, j=1, \ldots m-p$. Moreover, denoting as $\Lambda_{p}=\operatorname{Diag}\left(\lambda_{m-p+1}, \cdots, \lambda_{m}\right)$ the diagonal matrix with the largest eigenvalues, it turns out that

$$
\Pi=\left(V^{\top} \mathscr{P}_{r} M \mathscr{P}_{r}^{\top} V\right)^{-1}=\Lambda_{p}^{-1} .
$$

Then

$$
\begin{aligned}
P^{-1} \mathscr{P}_{r} M \mathscr{P}_{r}^{\top} z_{j}= & \nu V \Lambda_{p}^{-1} V^{\top} \mathscr{P}_{r} M \mathscr{P}_{r}^{\top} z_{j} \\
& +\left(I_{m}-V \Lambda_{p}^{-1} V^{\top} \mathscr{P}_{r} M \mathscr{P}_{r}^{\top}\right) P_{N E}^{-1} \\
& \cdot\left(\mathscr{P}_{r} M \mathscr{P}_{r}^{\top} z_{j}-\mathscr{P}_{r} M \mathscr{P}_{r}^{\top} V \Lambda_{p}^{-1} V^{\top} \mathscr{P}_{r} M \mathscr{P}_{r}^{\top} z_{j}\right) \\
= & \left(I_{m}-V \Lambda_{p}^{-1} V^{\top} \mathscr{P}_{r} M \mathscr{P}_{r}^{\top}\right) P_{N E}^{-1} \mathscr{P}_{r} M \mathscr{P}_{r}^{\top} z_{j} \\
= & \left(I_{m}-V \Lambda_{p}^{-1} V^{\top} \mathscr{P}_{r} M \mathscr{P}_{r}^{\top}\right) \lambda_{j} z_{j}=\lambda_{j} z_{j} .
\end{aligned}
$$

This completes the proof.
Usually the columns of $V$ are chosen as the (approximate) eigenvectors of $P_{N E}^{-1} \mathscr{P}_{r} M \mathscr{P}_{r}^{\top}$ corresponding to the smallest eigenvalues of this matrix. However, we could choose instead, as the columns of $V$, the rightmost eigenvectors of $P_{N E}^{-1} \mathscr{P}_{r} M \mathscr{P}_{r}^{\top}$, approximated with low accuracy by the function eigs of MATLAB. In this case, the $\nu$ value must be selected to satisfy $\lambda_{\min }\left(P_{N E}^{-1} \mathscr{P}_{r} M \mathscr{P}_{r}^{\top}\right)<\nu \ll$ $\lambda_{\max }\left(P_{N E}^{-1} \mathscr{P}_{r} M \mathscr{P}_{r}^{\top}\right)$. For example, we could choose $\nu=10$, and the column size of $V$ as $p=10$.

Finally, by computing approximately the rightmost eigenvectors, we would expect a slight perturbation of $\lambda_{1}, \ldots, \lambda_{m-p}$, depending on the accuracy of this approximation.

### 5.3 Regularized saddle point matrices

Let us now consider the regularized saddle point system in (5.1). Here we drop the assumption used in the previous section, and consider general, possibly non-diagonal, matrices $Q$. In what follows, we discuss two families of preconditioning strategies, noting their advantages and disadvantages. All presented preconditioners will be positive definite in order to be usable within the MINRES method, which is a short-recurrence iterative solver, suitable for solving symmetric indefinite or quasidefinite systems. This allows us to avoid non-symmetric long-recurrence solvers like the GMRES method.

### 5.3.1 Block diagonal preconditioners

The most common approach is to employ a block diagonal preconditioner (see $[21,23,132,161])$. In particular, given an approximation for matrix $Q$, say $\widetilde{Q}$, and the approximation for matrix $\widetilde{M} \equiv A\left(\widetilde{Q}+\rho I_{n}\right)^{-1} A^{\top}+\delta I_{m}$, given either in (5.2) or in (5.7), say $P_{N E}$, we can define the following preconditioner for the matrix in (5.1):

$$
P_{A S}=\left[\begin{array}{cc}
\widetilde{Q}+\rho I_{n} & 0_{n, m}  \tag{5.9}\\
0_{m, n} & P_{N E}
\end{array}\right] .
$$

From Remark 6, we know that if $\widetilde{Q}=\operatorname{Diag}(Q)$, or if we choose $\widetilde{Q}$ as

$$
\mathscr{P}_{c}^{\top} \widetilde{Q} \mathscr{P}_{c}=\left[\begin{array}{cc}
\operatorname{Diag}\left(Q^{(\mathcal{N}, \mathcal{N})}\right) & 0_{|\mathcal{N}|| | \mathcal{B} \mid}  \tag{5.10}\\
0_{|\mathcal{B}|,|\mathcal{N}|} & Q^{(\mathcal{B}, \mathcal{B})}
\end{array}\right] .
$$

where $\mathcal{N}$ corresponds to the columns of $A$ that are dropped (as in Section 5.2.2), we can directly use Theorem 5.2.1 to bound the eigenvalues of the preconditioned matrix $P_{N E}^{-1} \widetilde{M}$. In the latter case, notice that $\left(Q^{(\mathcal{B}, \mathcal{B})}+\rho I_{|\mathcal{B}|}\right)^{-1}$ does not introduce significant fill-in in the $(2,2)$ block of the preconditioner in (5.9), as we can implicitly invert this block using the methodology presented in Section 5.2.2. The aforementioned bounds can be used to bound the spectrum of the preconditioned system $P_{A S}^{-1} K$, where $K$ is the matrix in (5.1). For the rest of this subsection, we assume that $\widetilde{Q}$ is either diagonal, or is given by (5.10), and $P_{N E}$ is given either as in (5.2) or as in (5.7).

We proceed with the spectral analysis. To that end, let $F:=Q+\rho I_{n}$ and $\widetilde{F}:=\widetilde{Q}+\rho I_{n}$. The following theorem will characterize the eigenvalues of $P_{A S}^{-1} K$ in terms of the extremal eigenvalues of the preconditioned $(1,1)$ block of $(5.1)$, $\widetilde{F}^{-1} F_{k}$, and of $P_{N E}^{-1} \widetilde{M}$ (which can be described by Theorem 5.2.1). We will work with (SPD) similarity transformations of these matrices defined as

$$
\begin{equation*}
\hat{F}=\widetilde{F}^{-1 / 2} F \widetilde{F}^{-1 / 2}, \quad \hat{M}_{N E}=P_{N E}^{-1 / 2} \widetilde{M} P_{N E}^{-1 / 2} \tag{5.11}
\end{equation*}
$$

and set

$$
\begin{aligned}
& \alpha_{N E}=\lambda_{\min }\left(\hat{M}_{N E}\right), \quad \beta_{N E}=\lambda_{\max }\left(\hat{M}_{N E}\right), \quad \kappa_{N E}=\frac{\beta_{N E}}{\alpha_{N E}}, \\
& \alpha_{F}=\lambda_{\min }(\hat{F}), \quad \beta_{F}=\lambda_{\max }(\hat{F}), \quad \kappa_{F}=\frac{\beta_{F}}{\alpha_{F}}
\end{aligned}
$$

Hence, an arbitrary element of the Rayleigh quotient (or numerical range) of these matrices is represented as:

$$
\gamma_{N E} \in W\left(\hat{M}_{N E}\right)=\left[\alpha_{N E}, \beta_{N E}\right], \quad \gamma_{F} \in W(\hat{F})=\left[\alpha_{F}, \beta_{F}\right]
$$

Similarly, an arbitrary element of $W\left(P_{N E}\right)$ is denoted by

$$
\gamma_{p} \in\left[\lambda_{\min }\left(P_{N E}\right), \lambda_{\max }\left(P_{N E}\right)\right] \subseteq\left[\delta, \frac{\sigma_{\max }^{2}(A)}{\rho}+\delta\right)
$$

Since we either have $\widetilde{Q}=\operatorname{Diag}(Q)$, or $\widetilde{Q}$ given as in (5.10), we can observe that $\alpha_{F} \leq 1 \leq \beta_{F}$ as

$$
\frac{1}{n} \sum_{i=1}^{n} \lambda_{i}\left(\widetilde{F}^{-1} F\right)=\frac{1}{n} \operatorname{Tr}\left(\widetilde{F}^{-1} F\right)=1
$$

Theorem 5.3.1. The eigenvalues of $P_{A S}^{-1} K$ lie in the union of the following intervals:

$$
I_{-}=\left[-\beta_{F}-\sqrt{\beta_{N E}},-\alpha_{F}\right] ; \quad I_{+}=\left[\frac{1}{2}\left(-\beta_{F}+\sqrt{\beta_{F}^{2}+4 \alpha_{N E}}\right), 1+\sqrt{\beta_{N E}-1}\right] .
$$

Proof. The eigenvalues of $P_{A S}^{-1} K$ are the same as those of

$$
P_{A S}^{-1 / 2} K P_{A S}^{-1 / 2}=\left[\begin{array}{cc}
\widetilde{F}^{-1 / 2} & 0_{n, m} \\
0_{m, n} & P_{N E}^{-1 / 2}
\end{array}\right]\left[\begin{array}{cc}
-F & A^{\top} \\
A & \delta I_{m}
\end{array}\right]\left[\begin{array}{cc}
\widetilde{F}^{-1 / 2} & 0_{n, m} \\
0_{m, n} & P_{N E}^{-1 / 2}
\end{array}\right]=\left[\begin{array}{cc}
-\hat{F} & R^{\top} \\
R & \delta P_{N E}^{-1}
\end{array}\right],
$$

where $\hat{F}$ is defined in (5.11) and $R:=P_{N E}^{-1 / 2} A \widetilde{F}^{-1 / 2}$. Any eigenvalue $\lambda$ of the preconditioned matrix $P_{A S}^{-1 / 2} K P_{A S, k}^{-1 / 2}$ must therefore satisfy

$$
\begin{align*}
-\hat{F} w_{1}+\quad R^{\top} w_{2} & =\lambda w_{1}  \tag{5.12}\\
R w_{1}+\delta_{k} P_{N E}^{-1} w_{2} & =\lambda w_{2} . \tag{5.13}
\end{align*}
$$

First note that

$$
\begin{equation*}
R R^{\top}=P_{N E}^{-1 / 2} A \widetilde{F}^{-1} A^{\top} P_{N E}^{-1 / 2}=P_{N E}^{-1 / 2}\left(\widetilde{M}-\delta I_{m}\right) P_{N E}^{-1 / 2}=\hat{M}_{N E, k}-\delta P_{N E}^{-1} . \tag{5.14}
\end{equation*}
$$

If $P_{N E}$ is chosen as in (5.2) or as in (5.7), the eigenvalues of $R R^{\top}$ are characterized by Theorem 5.2.1. If $\lambda \notin\left[-\beta_{F},-\alpha_{F}\right]$ then $\hat{F}+\lambda I_{n}$ is symmetric positive (or negative) definite; moreover $R^{\top} w_{2} \neq 0_{n}$. Then from (5.12) we obtain

$$
w_{1}=\left(\hat{F}+\lambda I_{n}\right)^{-1} R^{\top} w_{2},
$$

which, after substituting in (5.13) yields

$$
R\left(\hat{F}+\lambda I_{n}\right)^{-1} R^{\top} w_{2}+\delta P_{N E}^{-1} w_{2}=\lambda w_{2} .
$$

Premultiplying by $w_{2}^{\top}$ and dividing by $\left\|w_{2}\right\|^{2}$, we obtain the following equation where we have set $z=R^{\top} w_{2}$.

$$
\lambda=\frac{z^{\top}\left(\hat{F}+\lambda I_{n}\right)^{-1} z}{z^{\top} z} \frac{w_{2}^{\top} R R^{\top} w_{2}}{w_{2}^{\top} w_{2}}+\delta \frac{w_{2}^{\top} P_{N E}^{-1} w_{2}}{w_{2}^{\top} w_{2}}=\frac{1}{\gamma_{F}+\lambda}\left(\gamma_{N E}-\frac{\delta}{\gamma_{p}}\right)+\frac{\delta}{\gamma_{p}} .
$$

So $\lambda$ must satisfy the following second-order algebraic equation

$$
\lambda^{2}+\left(\gamma_{F}-\omega\right) \lambda-\left(\omega\left(\gamma_{F}-1\right)+\gamma_{N E}\right)=0 .
$$

where we have set $\omega=\frac{\delta}{\gamma_{p}}$ satisfying $\omega \leq 1$. Notice that $\left(\gamma_{N E}-\omega\right) \geq 0$ by construction.

We first consider the negative eigenvalue solution of the previous algebraic equation, that is:

$$
\begin{aligned}
\lambda_{-} & =\frac{1}{2}\left[\omega-\gamma_{F}-\sqrt{\left(\gamma_{F}-\omega\right)^{2}+4\left(\omega \gamma_{F}-\omega+\gamma_{N E}\right)}\right] \\
& =\frac{1}{2}\left[\omega-\gamma_{F}-\sqrt{\left(\gamma_{F}+\omega\right)^{2}+4\left(\gamma_{N E}-\omega\right)}\right] \\
& \leq \frac{1}{2}\left[\omega-\gamma_{F}-\sqrt{\left(\gamma_{F}+\omega\right)^{2}}\right]=-\gamma_{F} \leq-\alpha_{F}
\end{aligned}
$$

In order to derive a lower bound on $\lambda_{-}$we work similarly. That is:

$$
\begin{aligned}
\lambda_{-} & =\frac{1}{2}\left[\omega-\gamma_{F}-\sqrt{\left(\gamma_{F}+\omega\right)^{2}+4\left(\gamma_{N E}-\omega\right)}\right] \\
& \geq \frac{1}{2}\left[-\gamma_{F}-\sqrt{\gamma_{F}^{2}+4 \gamma_{N E}}\right] \\
& \geq \frac{1}{2}\left[-\beta_{F}-\sqrt{\beta_{F}^{2}+4 \beta_{N E}}\right] \geq-\beta_{F}-\sqrt{\beta_{N E}}
\end{aligned}
$$

where we used the fact that the $\lambda_{-}$is an increasing function with respect to $\omega$, and decreasing with respect to $\gamma_{N E}$. Combining all the previous yields:

$$
\lambda_{-}\left\{\begin{array}{l}
\geq-\beta_{F}-\sqrt{\beta_{N E}} \\
\leq-\alpha_{F}
\end{array}\right.
$$

Note that this interval for $\lambda_{-}$contains the interval $\left[-\beta_{F},-\alpha_{F}\right]$, which we have excluded in order to carry out the analysis.

Regarding the positive eigenvalues we have that:

$$
\begin{aligned}
\lambda_{+} & =\frac{1}{2}\left[\omega-\gamma_{F}+\sqrt{\left(\gamma_{F}-\omega\right)^{2}+4\left(\omega \gamma_{F}-\omega+\gamma_{N E}\right)}\right] \\
& =\frac{1}{2}\left[\omega-\gamma_{F}+\sqrt{\left(\gamma_{F}+\omega\right)^{2}+4\left(\gamma_{N E}-\omega\right)}\right]
\end{aligned}
$$

We proceed by finding a lower bound for $\lambda_{+}$. To that end, we notice that $\lambda_{+}$is a decreasing function with respect to the variable $\gamma_{F}$ and increasing with respect to $\gamma_{N E}$. Hence, we have that:

$$
\begin{aligned}
\lambda_{+} & \geq \frac{1}{2}\left[\omega-\beta_{F}+\sqrt{\left(\beta_{F}+\omega\right)^{2}+4\left(\alpha_{N E}-\omega\right)}\right] \\
& \geq \frac{1}{2}\left[-\beta_{F}+\sqrt{\beta_{F}^{2}+4 \alpha_{N E}}\right]
\end{aligned}
$$

where the last inequality follows because the penultimate expression is increasing with respect to $\omega$. Similarly, in order to derive an upper bound for $\lambda_{+}$, we observe that $\lambda_{+}$is an increasing function with respect to $\omega$, decreasing with respect to
$\gamma_{F}$ and increasing with respect to $\gamma_{N E}$. Combining all the previous yields:

$$
\lambda_{+} \leq \frac{1}{2}\left[1-\alpha_{F}+\sqrt{\left(\alpha_{F}+1\right)^{2}+4\left(\beta_{N E}-1\right)}\right] \leq 1+\sqrt{\beta_{N E}-1},
$$

where we used the fact that $\omega \leq 1$. Then, combining all the previous gives the desired bounds, that is:

$$
\lambda_{+}\left\{\begin{array}{l}
\geq \frac{1}{2}\left[-\beta_{F}+\sqrt{\beta_{F}^{2}+4 \alpha_{N E}}\right] \\
\leq 1+\sqrt{\beta_{N E}-1}
\end{array}\right.
$$

and completes the proof.

Remark 5.3.1. It is well known that a pessimistic bound on the convergence rate of MINRES can be obtained if the size of $I_{-}$and $I_{+}$are roughly the same. In our case, as usually $\beta_{F} \ll \beta_{N E}$, we can assume that the length of both intervals is roughly $\sqrt{\beta_{N E}}$. As a heuristic we may therefore use [67, Theorem 4.14], which predicts the reduction of the residual in the $P_{A S}^{-1}$-norm in the case where both intervals have exactly equal length. This then implies that

$$
\frac{\left\|r_{k}\right\|}{\left\|r_{0}\right\|} \leq 2\left(\frac{\kappa-1}{\kappa+1}\right)^{\lfloor k / 2\rfloor}
$$

where

$$
\kappa \approx \frac{1}{(1 / 2) \alpha_{F}\left(-\beta_{F}+\sqrt{\beta_{F}^{2}+4 \alpha_{N E}}\right)}\left(1+\sqrt{\beta_{N E}-1}\right)\left(\beta_{F}+\sqrt{\beta_{N E}}\right) .
$$

Remark 5.3.2. In the LP case $\widetilde{F}_{k}=F_{k}$ and therefore $\kappa_{F}=1$. It then turns out that $\kappa \approx 2 \kappa_{N E}$. The number of MINRES iterations is then driven by $2 \kappa_{N E}$ while the CG iterations depend on $\sqrt{\kappa_{N E}}$ [111]. We highlight that different norms are used to describe the reduction in the relative residual norm for MINRES and CG.

Remark 7. Notice that further approximations can be employed here. In particular, we could define a banded approximation of $Q$ and then employ the approximation proposed earlier. The implicit application of the Schur complement in Section 5.2.2 gives us complete freedom on how to approximate $Q$, and hence we no longer rely on diagonal approximations. We will return to this point in the numerical experiments.

### 5.3.2 Factorization-based preconditioners

Finally, given the regularized nature of the systems under consideration, we can construct factorization-based preconditioners for MINRES. In particular, we can compute $K=L D L^{\top}$ (with $K$ in (5.1)), where $D$ is a diagonal matrix (since $K$ is quasidefinite [171]) having $n$ negative and $m$ positive elements in its diagonal.

Then, by defining $P_{K}:=L|D|^{\frac{1}{2}}$, the preconditioned saddle point matrix reads:

$$
P_{K}^{-1} K P_{K}^{-\top}=|D|^{-1} D
$$

and hence contains only two distinct eigenvalues -1 and 1 [74, 134]. As before, let us assume that we have available a splitting of the columns of $A$ such that $A \mathscr{P}_{c}=\left[\begin{array}{ll}A^{\mathcal{B}} & A^{\mathcal{N}}\end{array}\right]$ where $\mathcal{B}$ contains indices corresponding to the smallest diagonal elements of $Q$. Then, we can precondition $K$, left and right, by $\widehat{P}_{K}:=\widehat{L}|\widehat{D}|^{\frac{1}{2}}$, where $\widehat{K}=\widehat{L} \widehat{D} \widehat{L}^{\top}$ and:

$$
\widehat{K}:=\left[\begin{array}{cc}
-\widetilde{Q} & \widetilde{A}^{\top}  \tag{5.15}\\
\widetilde{A} & \delta I_{m},
\end{array}\right]
$$

with $\widetilde{A}:=\left[\begin{array}{ll}A^{\mathcal{B}} & 0_{m, \mid \mathcal{N}]}\end{array}\right] \mathscr{P}_{c}^{\top}$, and $\widetilde{Q}$ defined as in (5.10).
Further limited-memory versions of this preconditioner can be employed, e.g. by using the methodologies presented in [134, 159]. Other approximations of the blocks of $\widehat{K}$, based on the structure of the problem at hand, could also be possible, as already mentioned in the previous subsection.

We should note, however, that this approach is less stable than the $L D L^{\top}$ based approach presented in Section 5.3.1. This is because we are required to use only diagonal pivots during the $L D L^{\top}$ decomposition for this methodology to work. If $\delta$ or $\rho$ have very small values, the stability of the factorization could be compromised, and we would have to heavily rely on stability introduced by means of uniform [154] or weighted regularization [1]. On the other hand, the $L D L^{\top}$-based methodology presented in Section 5.3.1 would not be affected by the occasional use of $2 \times 2$ pivots. Of course the latter is not the case if the "analyze" phase of the factorization is performed separately, however, the subset of columns in $\mathcal{B}$ may significantly change from one iteration to the next, making this strategy less attractive. Nevertheless, this factorization-based approach can be more efficient than the approach presented in Section 5.3.1, when solving certain non-separable convex programming problems. That is because the approach in Section 5.3.1 requires the computation of a $L D L^{\top}$ decomposition of $\widetilde{P}_{N E}$ (with potential $2 \times 2$ pivots) as well as a Cholesky decomposition of $\widetilde{Q}+\rho I_{n}$ (or some iterative scheme which could be application dependent, as in [140]).

### 5.4 Regularized IPMs: numerical results

Let us now focus on the case of the regularized saddle point systems (and their respective normal equations) arising from the application of regularized IPMs to convex programming. The MATLAB code, which is based on the IP-PMM presented in Section 3.5.1, can be found on GitHub ${ }^{1},{ }^{2}$. The reader is referred to Section 3.5.1 for the implementation details of the algorithm (such as termination criteria, the employed predictor-corrector scheme for the solution of the Newton system, as well as the tuning of the algorithmic regularization parameters). A Newton direction is accepted if it is at least 3-digit accurate, and the associated

[^1]iterative methods (i.e. PCG or MINRES) are terminated if the following accuracy is reached: $\frac{t_{0}}{\max \{1,\|\mathrm{rhs}\| \|\}}$, where tol is the tolerance requested by the user, and rhs is the right hand side of the system being solved. We note that this condition is rather strict, but it ensures convergence of IP-PMM for a very wide range of problems, hence it allows the implementation to be very general and robust. All the presented experiments were run on a PC with a 2.2 GHz Intel Core i7 processor (hexa-core), 16GB RAM, run under the Windows 10 operating system. The MATLAB version used was 2019a.

### 5.4.1 Linear programming

Let us initially focus on linear programming problems of the following form:

$$
\begin{equation*}
\min _{x \in \mathbb{R}^{n}} c^{\top} x, \text { s.t. } A x=b, x^{\mathcal{I}} \geq 0_{|\mathcal{I}|}, x^{\mathcal{F}} \text { free, } \tag{LP}
\end{equation*}
$$

where $A \in \mathbb{R}^{m \times n}, \mathcal{I} \cap \mathcal{F}=\emptyset$, and $\mathcal{I} \cup \mathcal{F}=\{1, \ldots, n\}$. Applying regularized IPMs to problems like (LP), one often solves a regularized normal equations system at every iteration. Such systems have a matrix of the following form:

$$
M=A G A^{\top}+\delta I_{m}, \quad(G)^{(i, i)}= \begin{cases}\frac{1}{\rho}, & \text { if } i \in \mathcal{F},  \tag{5.16}\\ \frac{1}{\rho+z^{i} / x^{i}} & \text { if } i \in \mathcal{I},\end{cases}
$$

where $\delta, \rho>0$ and $z \in \mathbb{R}^{n}$ (where $z^{\mathcal{I}} \geq 0_{|\mathcal{I}|}, z^{\mathcal{F}}=0_{|\mathcal{F}|}$ ) are the dual slack variables. Notice that the IPM penalty parameter $\mu$ is often tuned as $\mu=\frac{\left(x^{I}\right)^{\top} z^{I}}{n}$ and we expect that $\mu \rightarrow 0$. The variables are naturally split into "basic"- $\mathcal{B}$ (not in the simplex sense), "non-basic" $-\mathcal{N}$, and "undecided" $\mathcal{U}$. Hence, as IPMs progress towards optimality, we expect the following partition of the quotient $\frac{x^{I}}{z^{I}}$ :

$$
\begin{array}{ll}
\forall j \in \mathcal{N}: x^{j} \rightarrow 0, \quad z^{j} \rightarrow \hat{z}^{j}>0 & \Rightarrow \frac{x^{j}}{z^{j}}=\frac{x^{j} z^{j}}{\left(z^{j}\right)^{2}}=\boldsymbol{\Theta}(\mu), \\
\forall j \in \mathcal{B}: x^{j} \rightarrow \hat{x}^{j}>0, \quad z^{j} \rightarrow 0 & \Rightarrow \frac{x^{j}}{z^{j}}=\frac{\left(x^{j}\right)^{2}}{x^{j} z^{j}}=\boldsymbol{\Theta}\left(\mu^{-1}\right), \\
\forall j \in \mathcal{U}: x^{j}=\boldsymbol{\Theta}(1), \quad z^{j}=\boldsymbol{\Theta}(1) & \Rightarrow \frac{x^{j}}{z^{j}}=\boldsymbol{\Theta}(1),
\end{array}
$$

where $\mathcal{N}, \mathcal{B}$ and $\mathcal{U}$ are mutually disjoint, and $\mathcal{N} \cup \mathcal{B} \cup \mathcal{U}=\mathcal{I}$. For the rest of this section, we assume that $\delta=\boldsymbol{\Theta}(\rho)=\boldsymbol{\Theta}(\mu)$. This assumption is based on the developments in Chapters 3, 4. Following [23], we could precondition matrix $M$ using the following matrix:

$$
\begin{equation*}
\widehat{P}_{N E}=A^{\mathcal{R}} G^{\mathcal{R}}\left(A^{\mathcal{R}}\right)^{\top}+\delta I_{m}, \tag{5.17}
\end{equation*}
$$

where $\mathcal{R}:=\mathcal{F} \cup \mathcal{B} \cup \mathcal{U}$. Then, by applying Theorem 5.2.1, we obtain:

$$
\lambda_{\max }\left(\widehat{P}_{N E}^{-1} M\right) \leq 1+\frac{\max _{j \in \mathcal{N}}\left(G^{j}\right)}{\delta} \sigma_{\max }^{2}(A), \quad \lambda_{\min }\left(\widehat{P}_{N E}^{-1} M\right) \geq 1
$$

The preconditioner in (5.17) is a special case of the preconditioner defined in Section 5.2. Indeed, it is derived by setting $k_{r}=0$ and then by dropping all columns belonging to $\mathcal{N}$, i.e. we set $k_{c}=|\mathcal{N}|$ and we drop the $k_{c}$ columns of $A$ corresponding to the smallest diagonal elements of $G$. For the rest of this subsection, we will refer to this preconditioner as the base preconditioner.

From our previous remarks, we notice that $\max _{j \in \mathcal{N}}\left(G^{j}\right)=\Theta(\mu)=\boldsymbol{\Theta}(\delta)$ implies that the spectrum of the preconditioned matrix remains bounded and is asymptotically independent of $\mu$ (assuming that $\delta=\boldsymbol{\Theta}(\mu)$ ). This will be the main preconditioner used in the numerical results, since it is extremely effective. However, it can be expensive to compute in certain cases, as it needs to be inverted by means of a Cholesky decomposition. To that end, we also propose further approximating this matrix by a preconditioner of the form presented in Section 5.2. This idea is based on the fact that the preconditioned conjugate gradient method is expected to converge in a small number of iterations, if the preconditioned system matrix can be written as:

$$
P^{-1} M=I+U+V,
$$

where $P$ is the preconditioner, $M$ is the normal equations matrix, $U$ is a lowrank matrix, and $V$ is a matrix with small norm. In our case, dropping the part of the normal equations corresponding to $\mathcal{N}$ contributes the small-norm term, and furthermore dropping a few dense columns (or sparsifying certain rows) contributes the low-rank term.

To construct such a preconditioner, we first need to note that $\mathcal{R}$ will change at every IPM iteration. However, we can heuristically choose which columns to drop (and/or rows to sparsify) based on the sparsity pattern of $A$. To that end, at the beginning of the optimization procedure, we count the number of non-zeros of each column and row of $A$, respectively. These can then be used to sort the columns and rows of $A$ in descending order of their number of non-zero entries. These sorted columns and rows can easily be represented by means of two permutation matrices $\mathscr{P}_{c}$ and $\mathscr{P}_{r}$. We note that this is a heuristic, and it is not guaranteed to identify the most problematic columns or rows (which can be sources of difficulty for IPMs). For a discussion on such heuristics, and alternatives, the reader is referred to [2, Section 4], and the references therein.

## Numerical results

Initially, we present some results using the preconditioner in (5.17) (which is equivalent to the preconditioner given in (5.7)) over a set of small to large scale linear programming instances. For completeness, we compare PCG and MINRES on a small set of problems. We further demonstrate the effectiveness of using low-rank updates to tune the preconditioner (following the developments in Section 5.2.3). A comparison with the factorization-based approach developed in Section 3.5.1 is also given, in order to stress the importance of using inexact linear algebra within IPM solvers.

Then, we present some results to show the effect of dropping dense columns and then of sparsifying dense rows of the constraint matrix using the strategy
outlined in Section 5.2. Subsequently, we present a comparison between the preconditioner in (5.2) (that is, $P_{N E}$ ), the preconditioner given in (5.7) (denoted as $\widehat{P}_{N E}$ ), and the one in (5.8) (that is, $\widetilde{P}_{N E}$ ).

The base preconditioner Firstly, we run the method on the Netlib collection [130], using the preconditioner given in (5.17) (or equivalently given in (5.7)). The tolerance used in these experiments was $10^{-4}$. In Table 5.1 we collect statistics from the runs of the method over some medium scale instances of the Netlib test set (see [130]). For each problem, two runs are presented; in the first one, we solve the normal equations systems using CG, while in the second one, we solve the augmented systems using MINRES. We expect (see Remark 5.3.1) that MINRES can require more than twice as many iterations as CG to deliver an equally good direction. Hence, we set maxit ${ }_{\text {MINRES }}=3 \cdot$ maxit $_{\mathrm{CG}}=300$ (i.e. maxit $_{\text {CG }}=100$ ). However, as we already mentioned in Remark 5.3.1, it is not entirely clear how many more iterations MINRES requires to guarantee the same quality of solution as PCG, since the two algorithms optimize different residual norms. Hence, requiring three times more iterations for MINRES is based on the behavior we observed through numerical experimentation. It comes as no surprise that IP-PMM with MINRES is slower, however, it allows us to solve general convex quadratic problems for which the normal equations are too expensive to be formed, or applied to a vector (indeed, this would often require the inversion of the matrix $Q+\Theta^{-1}+\rho_{k} I_{n}$, which is possibly non-diagonal). More specifically, IP-PMM with CG solved the whole set successfully in 141.25 seconds, requiring 2,907 IP-PMM iterations and 101,382 CG iterations. Furthermore, IP-PMM with MINRES also solved the whole set successfully, requiring 341.23 seconds, 3,012 total IP-PMM iterations and 297,041 MINRES iterations.

Table 5.1: Medium scale linear programming problems ( $\mathrm{tol}=10^{-4}$ )

| Name | $\mathbf{n n z}(\boldsymbol{A})$ | IP-PMM: CG |  |  |  | IP-PMM: MINRES |  |  |
| :--- | ---: | ---: | ---: | ---: | ---: | ---: | ---: | ---: | ---: |
|  |  | Time (s) | IP-Iter. | CG-Iter. |  | Time (s) | IP-Iter. | MR-Iter. |
| 80BAU3B | 29,063 | 3.15 | 48 | 1,886 |  | 10.74 | 47 | 4,883 |
| D2Q06C | 35,674 | 2.16 | 42 | 1,562 |  | 7.48 | 46 | 5,080 |
| D6CUBE | 43,888 | 0.97 | 30 | 933 |  | 3.26 | 30 | 3,279 |
| DFL001 | 41,873 | 10.18 | 54 | 2,105 |  | 29.07 | 54 | 6,292 |
| FIT2D | 138,018 | 3.16 | 28 | 836 |  | 10.62 | 28 | 2,558 |
| FIT2P | 60,784 | 40.78 | 31 | 924 |  | 65.15 | 31 | 2,978 |
| PILOT87 | 73,804 | 7.29 | 40 | 1,260 |  | 18.36 | 42 | 3,543 |
| QAP12 | 44,244 | 4.38 | 14 | 495 |  | 8.62 | 14 | 1,465 |
| QAP15 | 110,700 | 22.83 | 18 | 575 |  | 47.45 | 18 | 1,808 |

While we previously presented the runs of IP-PMM using MINRES over the Netlib collection, we did so only to compare the two variants. In particular, for the rest of this section we employ the convention that IP-PMM uses CG whenever $Q=0$ or $Q$ is diagonal, and MINRES whenever this is not the case.

Most of the previous experiments were conducted on small to medium scale problems. In Table 5.2 we provide the statistics of the runs of the method over a small set of large scale problems.

Table 5.2: Large-scale linear programming problems ( $\mathrm{tol}=10^{-4}$ )

| Name | $\mathbf{n n z}(\boldsymbol{A})$ | IP-PMM: CG |  |  |
| :--- | ---: | ---: | ---: | ---: |
|  |  | Time (s) | IP-Iter. | CG-Iter. |
| CONT1-1 | $7,031,999$ | $*^{1}$ | $*$ | $*$ |
| FOME13 | 285,056 | 72.59 | 54 | 2,098 |
| FOME21 | 465,294 | 415.51 | 96 | 4,268 |
| LP-CRE-B | 260,785 | 14.25 | 51 | 2,177 |
| LP-CRE-D | 246,614 | 16.04 | 58 | 2,516 |
| LP-KEN-18 | 358,171 | 128.78 | 42 | 1,759 |
| LP-OSA-30 | 604,488 | 20.88 | 67 | 2,409 |
| LP-OSA-60 | $1,408,073$ | 56.65 | 65 | 2,403 |
| LP-NUG-20 | 304,800 | 132.41 | 17 | 785 |
| LP-NUG-30 | $1,567,800$ | $2,873.67$ | 22 | 1,141 |
| LP-PDS-30 | 340,635 | 363.89 | 81 | 3,362 |
| LP-PDS-100 | $1,096,002$ | $3,709.93$ | 100 | 6,094 |
| LP-STOCFOR3 | 43,888 | 8.96 | 60 | 1,777 |
| NEOS | $1,526,794$ | $\dagger^{2}$ | $\dagger$ | $\dagger$ |
| NUG08-3rd | 148,416 | 80.72 | 17 | 682 |
| RAIL2586 | $8,011,362$ | 294.12 | 51 | 1,691 |
| RAIL4284 | $11,284,032$ | 391.93 | 46 | 1,567 |
| WATSON-1 | $1,055,093$ | 181.63 | 73 | 2,588 |
| WATSON-2 | $1,846,391$ | 612.68 | 140 | 5,637 |

[^2]We notice that the proposed version of IP-PMM is able to solve larger problems, as compared to IP-PMM using factorization (see Section 3.5.2, and notice that the experiments there were conducted on the same PC, using the same version of MATLAB). To summarize the comparison of the two approaches, we include Figure 5.1. It contains the performance profiles of the two methods, over the 26 largest linear programming problems of the QAPLIB, Kennington, Mittelmann, and Netlib libraries, for which at least one of the two methods was terminated successfully. In particular, in Figure 5.1a we present the performance profiles with respect to time, while in Figure 5.1 b we show the performance profiles with respect to the number of IPM iterations. IP-PMM with factorization is represented by the green line (consisting of triangles), while IP-PMM with PCG is represented by the blue line (consisting of stars). As one can observe, IP-PMM with factorization was able to solve only $84.6 \%$ of these problems, due to excessive memory requirements (namely, problems LP-OSA-60, LP-PDS-100, RAIL4284, LP-NUG-30 were not solved due to insufficient memory). As expected, however, it converges in fewer iterations for most problems that are solved successfully by both methods. Moreover, IP-PMM with PCG is able to solve every problem that is successfully solved by IP-PMM with factorization. Furthermore, it manages to do so requiring significantly less time, which can be observed in Figure 5.1a.

Figure 5.1: Performance profiles for large-scale linear programming problems

(a) Performance profile in terms of CPU time

(b) Performance profile in terms of iterations

Notice that we restrict the comparison to only large-scale problems, since this is the case of interest. In particular, IP-PMM with factorization is expected to be more efficient for solving small to medium scale problems.

Finally, in order to clarify the use of the low-rank (LR) updates we conducted an analysis on two specific - yet representative - linear systems, at (predictor and corrector) IP step \#12 for problem nug20. In Table 5.3 we report the results in solving these linear systems with the low-rank strategy and different accuracy/number of eigenpairs ( $\operatorname{LR}(p$, tol $)$ meaning that we approximate $p$ eigenpairs with eigs with a tolerance tol). The best choice, using $p=10$ and 0.1 accuracy, improves the $\widehat{P}_{N E}$ preconditioner both in terms of linear iterations and total CPU time.

|  |  | predictor |  | corrector |  |  |
| :--- | ---: | ---: | ---: | ---: | ---: | ---: |
|  | CPU(eigs) $(\mathrm{s})$ | CG-Iter. | CPU (s) | CG-Iter. | CPU (s) | CPU tot. (s) |
| No tuning |  | 95 | 10.71 | 95 | 11.10 | 21.81 |
| LR $(5,0.1)$ | 2.39 | 79 | 9.51 | 78 | 9.59 | 21.49 |
| LR $(10,0.1)$ | 3.00 | 69 | 8.14 | 67 | 7.64 | 18.78 |
| LR $(20,0.1)$ | 5.98 | 64 | 7.79 | 63 | 7.85 | 22.62 |
| LR $\left(20,10^{-3}\right)$ | 9.59 | 64 | 7.79 | 63 | 7.85 | 26.23 |

Table 5.3: CPU times and number of linear iterations for the various tuned preconditioners at IP iteration \#12 for problem nug20.

Figure 5.2 accounts for the steepest convergence profile of the preconditioned-with-tuning normal equations matrix, when using the optimal parameters.

Dropping dense columns versus factorizing directly. We run IP-PMM on four problems from the Netlib collection that have some dense columns, where dense is defined in this case to be a column with at least $15 \%$ non-zero elements. We compare an IP-PMM using Cholesky factorization for the solution of the associated Newton system (Exact), with an IP-PMM that uses the preconditioner presented in Section 5.2 alongside PCG (Inexact). The latter method is only allowed to drop dense columns (at most 30) to create the preconditioner. The results are collected in Table 5.4 (noting that the presented dimensions include additional constraints or variables needed to transform the problem to the format


Figure 5.2: Convergence profiles of PCG accelerated with $\widehat{P}_{N E}$ and $\widehat{P}_{N E}$ updated with $\operatorname{LR}(10,0.1)$. Linear systems at IP iteration \#12 for problem nug20.
accepted by IP-PMM, while nnz denotes the number of non-zero elements present in the Cholesky factor).

Table 5.4: The effect of dropping dense ( $>15 \%$ ) columns ( $\mathrm{tol}=10^{-6}$ ).

| Name | $\mathbf{m}$ | $\mathbf{n}$ | $\mathbf{k}_{\mathbf{c}}$ | $\mathbf{n n z}$ |  |  | time (s) |  |
| :--- | :--- | :--- | :--- | :--- | ---: | ---: | ---: | ---: |
|  |  |  |  |  | Exact | Inexact |  | Exact |
|  |  | Inexact |  |  |  |  |  |
| FIT1P | 10,525 | 21,024 | 20 | 197,676 | 26,706 |  | 1.31 | 0.50 |
| FORPLAN | 183 | 584 | 8 | 3,810 | 2,918 |  | 0.11 | 0.11 |
| ISRAEL | 174 | 316 | 30 | 12,261 | 1,744 |  | 0.12 | 0.34 |
| SEBA | 1,030 | 1,551 | 14 | 55,937 | 2,238 |  | 0.32 | 0.22 |

From Table 5.4 we can immediately see that certain dense columns present in the constraint matrix $A$ can have a significant impact on the sparsity pattern of the Cholesky factors. This is a well-known fact (see for example the discussion in [2, Section 4]). Notice that the Netlib collection contains only small- to mediumscale instances. For such problems, memory is not an issue, and hence direct methods tend to be faster than their iterative alternatives (like PCG). Despite the small size of the presented problems, we can see tremendous memory savings (and even a decrease in CPU time) for problems FIT1P and SEBA, by eliminating only a small number of dense columns.

Sparsifying dense rows versus factorizing directly. Next, we consider the case where the inexact version of IP-PMM is only allowed to sparsify dense rows, where dense is defined in this case to be a row with at least $25 \%$ non-zero elements.

Before moving to the numerical results, let us note some differences between sparsifying rows and dropping columns. Firstly, as we have shown in Section 5.2 , sparsifying $k$ rows can potentially introduce twice as many outliers, while dropping $k$ columns introduces at most $k$ eigenvalue outliers. Furthermore, the potential density induced in the Cholesky factors by a single dense column is usually more significant than that introduced by a single dense row. However,
we cannot know in advance how effective the dropping of a column will be. On the other hand, sparsifying dense rows introduces a certain separability in the normal equations matrix, allowing us to estimate very well the memory savings.

In Table 5.5 we compare the direct IP-PMM, to its inexact version that is only allowed to sparsify at most 30 dense rows of the problem.

Table 5.5: The effect of dropping dense ( $>25 \%$ ) rows ( $\mathrm{tol}=10^{-6}$ ).

| Name | $\mathbf{m}$ | $\mathbf{n}$ | $\mathbf{k}_{\mathbf{r}}$ | $\mathbf{n n z}$ |  |  | time (s) |  |
| :--- | :--- | :--- | :--- | :--- | :--- | :--- | ---: | ---: |
|  |  |  |  |  | Exact | Inexact |  | Exact |
|  | Inexact |  |  |  |  |  |
| BEACONFD | 173 | 295 | 17 | 2,903 | 1,475 |  | 0.06 | 0.17 |
| FIT1D | 1,050 | 2,075 | 11 | 14,726 | 4,973 |  | 0.22 | 0.39 |
| FIT2D | 10,525 | 21,024 | 12 | 139,843 | 49,513 |  | 1.51 | 2.90 |
| WOOD1P | 244 | 2,595 | 27 | 19,088 | 13,879 |  | 0.34 | 0.49 |

From Table 5.5 we can observe that the required memory to form the Cholesky factors is consistently decreased but CPU time is increased by the row-dropping strategy. We should note that there is an increase in CPU time, which relates to the size of the problems under consideration, and CPU time as well as memory advantages can be observed if the problem is sufficiently large with sufficiently many dense rows. In particular, this row sparsifying strategy will be employed in Chapter 6 in order to tackle fMRI sparse approximation problems (in which the constraint matrix contains thousands of dense rows). In this special case, we show that memory requirements are significantly lowered, allowing this inexact version to outperform its exact counterpart, while being competitive with standard state-of-the-art first-order methods traditionally used to solve such problems (see Section 6.4).

The Cholesky versus the $\mathrm{LDL}^{\top}$ approach. Let us now provide some numerical evidence for the potential benefits of the approach presented in Section 5.2 .2 over that presented in Section 5.2.1. To that end, we run three inexact versions of IP-PMM. The first uses the preconditioner given in (5.2) (allowing at most 15 dense columns/rows to be dropped/sparsified), the second uses the preconditioner given in (5.7) (which is the same as the latter without employing the strategy of dropping/sparsifying dense columns/rows; i.e. the base preconditioner), while the third version uses the preconditioner in (5.8). In all three cases the set $\mathcal{B}$, used to decide which columns are dropped irrespectively of their density, is determined as indicated in the beginning of this section.

Table 5.6: Cholesky versus $L D L^{\top}$ preconditioners $\left(\right.$ tol $\left.=10^{-6}\right)$.

| Name | Krylov Its. |  |  | max nnz |  |  | time (s) |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | $P_{\text {NE }}$ | $\widehat{P}_{N E}$ | $\widetilde{P}_{N E}$ | $P_{\text {NE }}$ | $\widehat{P}_{N E}$ | $\widetilde{P}_{N E}$ | $P_{N E}$ | $\widehat{P}_{N E}$ | $\widetilde{P}_{N E}$ |
| AGG | 2,966 | 2,966 | 3,299 | $1.6 \cdot 10^{4}$ | $1.6 \cdot 10^{4}$ | $7.7 \cdot 10^{3}$ | 0.42 | 0.42 | 1.25 |
| FIT1P | 2,448 | 2,091 | 2,049 | $1.2 \cdot 10^{5}$ | $2.1 \cdot 10^{5}$ | $9.14 \cdot 10^{3}$ | 0.85 | 1.59 | 0.64 |
| FIT2P | 2,577 | 2,382 | 2,659 | $4.3 \cdot 10^{6}$ | $4.6 \cdot 10^{6}$ | $9.6 \cdot 10^{4}$ | 30.07 | 43.10 | 6.64 |
| SEBA | 2,629 | 2,340 | 2,380 | $2.24 \cdot 10^{3}$ | $5.6 \cdot 10^{4}$ | $9.1 \cdot 10^{3}$ | 0.32 | 0.58 | 0.69 |

From Table 5.6 we can observe that the $L D L^{\top}$-based preconditioner can provide substantial benefits for certain problems (e.g. see problem FIT1P, FIT2P). Nevertheless, we should note that this approach is usually slower (albeit more stable). In these specific runs we use a $10^{-9}$ threshold for the ldl function of MATLAB. The larger this threshold is, the slower, and more stable the factorization. In problem AGG, the two Cholesky-based variants are exactly the same, as no dense columns or rows were dropped.

There is a long-standing discussion on the comparison between the Cholesky and the $L D L^{\top}$ decompositions. The former tend to be faster and usually easier to implement, while the latter tend to be slower, more stable, and more general. For more on this subject, the reader is referred to [2, Section 4] and the references therein.

### 5.4.2 Convex quadratic programming

Next, we consider problems of the following form:

$$
\begin{equation*}
\min _{x \in \mathbb{R}^{n}} c^{\top} x+\frac{1}{2} x^{\top} Q x, \quad \text { s.t. } A x=b, x^{\mathcal{I}} \geq 0_{|\mathcal{I}|}, x^{\mathcal{F}} \text { free. } \tag{QP}
\end{equation*}
$$

Let us notice that a similar partitioning as that presented in Section 5.4.1 also holds in this case. Hence, the index set $\mathcal{N}$ guides us on which columns of $A$ to drop. In the case where $Q$ is either diagonal, or can be well-approximated by a diagonal, the discussion of Section 5.4.1 about dropping dense columns (or sparsifying dense rows) also applies here.

In what follows we make use of three different preconditioners. We compare the two block-diagonal preconditioners given in Section 5.3.1. The first is called $P_{A S, 1}$, and employs a diagonal approximation for $Q$, allowing one to drop dense columns and/or sparsify dense rows as shown in Section 5.2.1 (and will be called as the base preconditioner for the rest of this subsection), and the second is called $P_{A S, 2}$, and employs a block diagonal approximation of $Q$, using the implicit inversion of the Schur complement proposed in Section 5.2.2. The block-diagonal preconditioners are also compared against the factorization-based preconditioner presented in Section 5.3.2, termed as $\widehat{P}_{K}$.

## Numerical results

In the following experiments, MINRES is used to solve the associated Newton systems. Initially, we present the runs of IP-PMM using the base preconditioner (i.e. $P_{A S, 1}$ ) over some medium to large scale instances from the Maros-Mészáros collection (see [116]). Then, we compare the three preconditioning strategies over certain medium-scale non-separable convex quadratic programming problems from this collection. Subsequently, we provide the runs of each method over some $L^{1}$-regularized PDE-constrained optimization problems. Finally, we show the robustness of the approach using the base preconditioners, by running the method over the whole Netlib and Maros-Mészáros collections with increasing accuracy.

Maros-Mészáros collection. We present the runs of the method over the Maros-Mészáros test set [116]. In Table 5.7, we collect statistics from the runs of the method over some medium and large scale instances of the collection, requesting a 4 -digit accurate solution.

Table 5.7: Medium and large-scale QP problems ( $\mathrm{tol}=10^{-4}$ ).

| Name | $\mathbf{n n z}(\boldsymbol{A})$ | $\mathbf{n n z}(\boldsymbol{Q})$ | IP-PMM |  |  |
| :--- | ---: | ---: | ---: | ---: | ---: |
|  |  |  | Time (s) | IP-Iter. | Krylov-Iter. |
| AUG2DCQP | 20,200 | 80,400 | 4.46 | 41 | 1,188 |
| CONT-100 | 49,005 | 10,197 | 3.95 | 23 | 68 |
| CONT-101 | 49,599 | 2,700 | 8.83 | 85 | 282 |
| CONT-200 | 198,005 | 40,397 | 39.84 | 109 | 422 |
| CONT-300 | 448,799 | 23,100 | 134,76 | 126 | 405 |
| CVXQP1_L | 14,998 | 69,968 | 54.77 | 111 | 12,565 |
| CVXQP3_L | 22,497 | 69,968 | 80.18 | 122 | 14,343 |
| LISWET1 | 30,000 | 10,002 | 3.55 | 41 | 1,249 |
| POWELL20 | 20,000 | 10,000 | 2.71 | 31 | 937 |
| QSHIP12L | 16,170 | 122,433 | 2.99 | 26 | 3,312 |

In Table 5.8, we report on the runs of the method using each of the three preconditioners on instances having a non-diagonal Hessian $Q$. In order to stress the importance of using non-diagonal Hessian information in the formation of the preconditioner, we request for a 6 -digit accurate solution. This will allow us to see certain limitations of the base preconditioner.

Table 5.8: Comparison of QP preconditioners ( $\mathrm{tol}=10^{-6}$ ).

| Name | Krylov Its. |  |  | max nnz |  |  | time (s) |  |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  | $P_{A S, 1}$ | $P_{A S, 2}$ | $\widehat{P}_{K}$ | $P_{A S, 1}$ | $P_{A S, 2}$ | $\widehat{P}_{K}$ | $P_{A S, 1}$ | $P_{A S, 2}$ | $\widehat{P}_{K}$ |
| CVXQP2_L | $\dagger^{1}$ | 7,858 | 4,981 | $5.06 \cdot 10^{4}$ | $1.47 \cdot 10^{6}$ | $1.47 \cdot 10^{6}$ | $\dagger$ | 63.76 | 41.45 |
| CVXQP2_M | 6,164 | 4,288 | 4,176 | $5.04 \cdot 10^{3}$ | $6.18 \cdot 10^{4}$ | $6.19 \cdot 10^{4}$ | 2.19 | 1.85 | 1.96 |
| Q25FV47 | 10,048 | 10,589 | 11,700 | $2.39 \cdot 10^{4}$ | $8.95 \cdot 10^{4}$ | $1.12 \cdot 10^{5}$ | 3.10 | 4.87 | 5.77 |
| QSHIP12L | 6,654 | 6,126 | 6,040 | $1.10 \cdot 10^{4}$ | $2.04 \cdot 10^{5}$ | $2.07 \cdot 10^{5}$ | 3.22 | 5.12 | 5.28 |
| STCQP1 | 7,240 | 5,140 | 5,419 | $7.00 \cdot 10^{5}$ | $1.26 \cdot 10^{5}$ | $1.26 \cdot 10^{5}$ | 20.45 | 11.00 | 10.87 |
| STCQP2 | 5,292 | 3,660 | 3,665 | $5.44 \cdot 10^{4}$ | $2.13 \cdot 10^{5}$ | $2.13 \cdot 10^{5}$ | 7.43 | 7.85 | 7.36 |

${ }^{1} \dagger$ indicates that the solver did not reach the desired accuracy.
From Table 5.8, one can observe that most of the time $P_{A S, 1}$ is rather inexpensive, and naturally requires some additional Krylov iterations. For problem CVXQP2, this preconditioner was unable to yield a 6 -digit accurate solution (in fact, it could easily find a 5 -digit accurate solution, but numerical inaccuracy prevented the algorithm from converging with a tolerance of $10^{-6}$ ). On the other hand, both $P_{A S, 2}$ and $\widehat{P}_{K}$ allowed IP-PMM to converge to a 6 -digit accurate solution in this case. This specific instance is quite vulnerable to perturbations, and the use of additional Hessian information within the preconditioner is crucial to guarantee the convergence of IP-PMM. Similar observations hold for the remaining CVXQP problems of this collection (which are not included here). For the
remaining instances, we can observe that all three approaches are competitive and the efficiency of each approach depends on the problem under consideration. We should note that for both $P_{A S, 2}$ and $\widehat{P}_{K}$, a threshold of $10^{-9}$ was used within the MATLAB ldl function (thus in the former case we favor efficiency over stability).

PDE-constrained optimization instances. Next, we compare the preconditioning approaches on some PDE-constrained optimization problems. In particular, we consider the $L^{1} / L^{2}$-regularized Poisson control problem, as well as the $L^{1} / L^{2}$-regularized convection-diffusion control problem with control bounds. We should emphasize at this point that while bespoke preconditioners have been created for PDE problems of this form, here we treat the discretized problems as if we hardly know anything about their structure, to demonstrate the generality of the approaches presented in this chapter.

We consider problems of the following form:

$$
\begin{array}{rl}
\min _{\mathrm{y}, \mathrm{u}} & \mathrm{~J}(\mathrm{y}(\boldsymbol{x}), \mathrm{u}(\boldsymbol{x})):=\frac{1}{2}\|\mathrm{y}-\overline{\mathrm{y}}\|_{\mathrm{L}^{2}(\Omega)}^{2}+\frac{\alpha_{1}}{2}\|\mathrm{u}\|_{\mathrm{L}^{1}(\Omega)}^{2}+\frac{\alpha_{2}}{2}\|\mathrm{u}\|_{\mathrm{L}^{2}(\Omega)}^{2}, \\
\text { s.t. } & \mathrm{Dy}(\boldsymbol{x})+\mathrm{u}(\boldsymbol{x})=\mathrm{g}(\boldsymbol{x}),  \tag{5.18}\\
& \mathrm{u}_{\mathrm{a}}(\boldsymbol{x}) \leq \mathrm{u}(\boldsymbol{x}) \leq \mathrm{u}_{\mathrm{b}}(\boldsymbol{x}),
\end{array}
$$

where $(\mathrm{y}, \mathrm{u}) \in \mathrm{H}^{1}(\Omega) \times \mathrm{L}^{2}(\Omega)$, D denotes some linear differential operator associated with the differential equation, $\boldsymbol{x}$ is a 2 -dimensional spatial variable, and $\alpha_{1}, \alpha_{2} \geq 0$ denote the regularization parameters of the control variable. We note that other variants for $J(y, u)$ are possible, including measuring the state misfit and/or the control variable in other norms, as well as alternative weightings within the cost functionals. In particular, the methods tested here also work well for $L^{2}$ norm problems (e.g. see [138]). We consider problems of the form of (5.18) to create an extra level of difficulty for our solvers.

The problem is considered on a given compact spatial domain $\Omega$, where $\Omega \subset \mathbb{R}^{2}$ has boundary $\partial \Omega$, and is equipped with Dirichlet boundary conditions. The algebraic inequality constraints are assumed to hold a.e. on $\Omega$. We further note that $u_{a}$ and $u_{b}$ may take the form of constants, or functions in spatial variables, however we restrict our attention to the case where these represent constants.

Problems in the form of (5.18) are often solved numerically, by means of a discretization method. In the following experiments we employ the Q1 finite element discretization implemented in IFISS $^{3}$ (see [65, 66]). Applying the latter yields a sequence of non-smooth convex programming problems, which can be transformed to the smooth form of (QP), by introducing some auxiliary variables to deal with the $\ell_{1}$ terms appearing in the objective (see [140, Section 2]). In order to restrict the memory requirements of the approach, we consider an additional approximation of $Q$ in the preconditioner $P_{A S, 2}$. In the cases under consideration,

[^3]the resulting Hessian matrix (without the barrier terms) takes the following form:
\[

Q=\left[$$
\begin{array}{ccc}
J_{M} & 0_{d, d} & 0_{d, d} \\
0_{d, d} & \alpha_{2} J_{M} & -\alpha_{2} J_{M} \\
0_{d, d} & -\alpha_{2} J_{M} & \alpha_{2} J_{M}
\end{array}
$$\right],
\]

where $J_{M}$ is the mass matrix of size $d$. We approximate each block of $Q$ by its diagonal (i.e. $\widetilde{J_{M}}=\operatorname{Diag}\left(J_{M}\right)$; an approximation which is known to be optimal [176]). The resulting matrix is then further approximated as discussed in Section 5.3. From now on, the $L D L^{\top}$ preconditioner, which is based on an approximation of $P_{A S, 2}$, will be referred to as $P_{A S, 3}$. For these examples, the preconditioning strategy based on $\widehat{P}_{K}$ (given in Section 5.3.2) behaved significantly worse, and hence was not included in the numerical results. The preconditioner $P_{A S, 3}$ can be useful in that it allows us to employ block diagonal preconditioners of which the Schur complement takes into account non-diagonal information of the Hessian matrix $Q$. In certain cases, this might result in a faster convergence of IP-PMM, as compared to $P_{A S, 1}$ (which might not allow IP-PMM to converge to a 6 -digit accurate solution).

The first problem that we consider is the two-dimensional $\mathrm{L}^{1} / \mathrm{L}^{2}$-regularized Poisson optimal control problem, with bound constraints on the control and free state, posed on the domain $\Omega=(0,1)^{2}$. Following [140, Section 5.1], we consider the constant control bounds $\mathrm{u}_{\mathrm{a}}=-2, \mathrm{u}_{\mathrm{b}}=1.5$, and the desired state $\overline{\mathrm{y}}=\sin \left(\pi x_{1}\right) \sin \left(\pi x_{2}\right)$. In Table 5.9, we fix $\alpha_{2}=10^{-2}$ (which we find to be the most numerically interesting case), and we present the runs of the method using the different preconditioning approaches, with increasing grid size, and varying $\mathrm{L}^{1}$ regularization parameter (that is $\alpha_{1}$ ). To reflect the change in the grid size, we report the number of variables of the optimization problem after transforming it to the IP-PMM format. We also report the overall number of Krylov iterations required for IP-PMM to converge (and the number of IP-PMM iterations in brackets), the maximum number of nonzeros stored in order to invert the associated preconditioners, as well as the required CPU time.

We can draw several observations from the results in Table 5.9. Firstly, one can observe that in this case, a diagonal approximation of $Q$ is sufficiently good to deliver fast convergence of MINRES. The block-diagonal preconditioner using non-diagonal Hessian information (i.e. $P_{A S, 3}$ ) required consistently fewer MINRES iterations (and not necessarily more memory; see the three largest experiments), however, this did not result in a reduction in CPU time. There are several reasons for this. Firstly, the Hessian of the problem becomes more diagonally dominant as the grid size is increased. As a result, the diagonal approximation of it remains robust with respect to the grid size for the problem under consideration. On the other hand, the algorithm uses the built-in MATLAB function ldl to factorize the preconditioner $P_{A S, 3}$. While this implementation is very stable, it employs a dynamic permutation at each IP-PMM iteration, which slows down the algorithm. In this case, a specialized method using preconditioner $P_{A S, 3}$ should employ a separate symbolic factorization step, that could be used in subsequent IP-PMM iterations, thus significantly reducing the CPU time. This is not done here, however, as we treat these PDE-constrained optimization problems as black-

Table 5.9: Comparison of QP preconditioners (Poisson Control: grid size and varying regularization, tol $=10^{-6}$ ).

| $n$ | $\alpha_{1}$ | Krylov (IP) Its. |  | max nnz |  | time (s) |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | $P_{A S, 1}$ | $P_{A S, 3}$ | $P_{A S, 1}$ | $P_{A S, 3}$ | $P_{A S, 1}$ | $P_{A S, 3}$ |
| $2.11 \cdot 10^{4}$ | $10^{-2}$ | 1,568 (14) | 1,148 (14) | $3.98 \cdot 10^{5}$ | $4.74 \cdot 10^{5}$ | 6.49 | 7.05 |
|  | $10^{-4}$ | 2,130 (15) | 1,266 (15) | $3.98 \cdot 10^{5}$ | $4.75 \cdot 10^{5}$ | 8.45 | 7.75 |
|  | $10^{-6}$ | 1,726 (14) | 1,159 (14) | $3.98 \cdot 10^{5}$ | $4.75 \cdot 10^{5}$ | 6.97 | 7.10 |
| $8.32 \cdot 10^{4}$ | $\overline{10} \overline{0}^{-2^{-}}$ | 1, $\left.\overline{3} \overline{1} \overline{1}^{-1} \overline{1}\right)$ | ${ }^{1}, \overline{099}$ (1) $\overline{1}$ ) | ${ }^{2} .12 \cdot 10^{6}{ }^{-1}$ | $2 . \overline{2} \cdot{ }^{2} \cdot 10^{6}$ | 21.65 | $4 \overline{3} . \overline{1} 5$ |
|  | $10^{-4}$ | 1,723 (14) | 1,187 (14) | $2.12 \cdot 10^{6}$ | $2.22 \cdot 10^{6}$ | 26.31 | 45.00 |
|  | $10^{-6}$ | 1,728 (14) | 1,192 (14) | $2.12 \cdot 10^{6}$ | $2.22 \cdot 10^{6}$ | 26.23 | 44.99 |
| $3.30 \cdot 10^{5}$ | $1 \overline{0}^{=2^{-}}$ | 1, 474 - 1 (1) | 1,220 (14) | ${ }^{1.07} \overline{7}^{-1} 1 \overline{1}^{7^{-}}$ | $1 . \overline{10} \cdot 1 \overline{0}^{7}$ | 112.01 | $22 \overline{1} .9 \overline{0}$ |
|  | $10^{-4}$ | 1,685 (15) | 1,240 (15) | $1.07 \cdot 10^{7}$ | $1.10 \cdot 10^{7}$ | 128.20 | 224.29 |
|  | $10^{-6}$ | 1,687 (15) | 1,231 (15) | $1.07 \cdot 10^{7}$ | $1.10 \cdot 10^{7}$ | 127.47 | 224.51 |
| $1.32 \cdot 10^{6}$ | $\overline{1} \overline{0}=2^{-}$ | $1, \overline{4} 65{ }^{-}(\overline{1} \overline{4})$ | $\left.{ }^{1}, \overline{120} \overline{0}^{-1} \overline{1} 4\right)$ | $5.51 \cdot 10^{7^{-}}$ | $5 . \overline{3} \cdot{ }^{\text {c }} 1 \overline{0}^{7}$ | 523.25 | ${ }^{1}, \overline{0} \overline{2} \overline{8} . \overline{3} \overline{0}$ |
|  | $10^{-4}$ | 1,484 (14) | 1,120 (14) | $5.51 \cdot 10^{7}$ | $5.38 \cdot 10^{7}$ | 517.65 | 1,030.71 |
|  | $10^{-6}$ | 1,479 (14) | 1,113 (14) | $5.51 \cdot 10^{7}$ | $5.38 \cdot 10^{7}$ | 515.67 | 1,045.35 |

box (notice that the implementation allows the user to feed an approximation of the Hessian, but does not allow the user to use a different $L D L^{\top}$ decomposition). In all the previous runs, the reported Krylov iterations include both the predictor and the corrector steps of IP-PMM (i.e. on average, when $P_{A S, 1}$ was used, every linear system was successfully solved in about 50 iterations, while with $P_{A S, 2}$, each linear system was solved in about 40 iterations). We should note that for the problem under consideration employing a predictor-corrector scheme is not necessary, however, we wanted to keep the implementation as general and robust as possible, without tailoring it to specific applications. For this problem, we can also observe that IP-PMM was robust with respect to the grid size (i.e. IP-PMM convergence was not affected by the size of the problem). This is often observed when employing an IPM for the solution of PDE optimization problems (e.g. see [138]), however, in theory one should expect dependence of IPM on the problem size.

Next we consider the optimal control of the convection-diffusion equation, i.e. $\epsilon \Delta \mathrm{y}+\mathrm{w} \nabla \mathrm{y}=\mathrm{u}$, on the domain $\Omega=(0,1)^{2}$, where w is the wind vector given by $\mathrm{w}=\left[2 \mathrm{x}_{2}\left(1-\mathrm{x}_{1}\right)^{2},-2 \mathrm{x}_{1}\left(1-\mathrm{x}_{2}^{2}\right)\right]^{\top}$, with control bounds $\mathrm{u}_{\mathrm{a}}=-2$, $\mathrm{u}_{\mathrm{b}}=$ 1.5 and free state (e.g. see [140, Section 5.2]). Once again, the problem is discretized using Q1 finite elements, employing the streamline upwind PetrovGalerkin (SUPG) upwinding scheme implemented in [33]. We define the desired state as $\overline{\mathrm{y}}=\exp \left(-64\left(\left(\mathrm{x}_{1}-0.5\right)^{2}+\left(\mathrm{x}_{2}-0.5\right)^{2}\right)\right)$ with zero boundary conditions. The diffusion coefficient $\epsilon$ is set as $\epsilon=0.02$. The $\mathrm{L}^{2}$ regularization parameter $\alpha_{2}$ is set as $\alpha_{2}=10^{-2}$. We run IP-PMM with the two different preconditioning approaches on the aforementioned problem, with different $\mathrm{L}^{1}$ regularization values (i.e. $\alpha_{1}$ ) and with increasing grid size. The results are collected in Table 5.10.

In Table 5.10 we can observe that the IP-PMM convergence can be slightly affected by the problem size, as well as by the $\mathrm{L}^{1}$ regularization parameter $\alpha_{1}$. When $P_{A S, 3}$ was employed, this behaviour did not relate to the inexact solution of the associated linear systems, but to the algorithmic scheme itself. However,

Table 5.10: Comparison of QP preconditioners (Convection-Diffusion Control: grid size and varying regularization, tol $\left.=10^{-6}\right)$.

| $n$ | $\alpha_{1}$ | Krylov (IP) Its. |  | max nnz |  | time (s) |  |
| :---: | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
|  |  | $P_{A S, 1}$ | $P_{A S, 3}$ | $P_{A S, 1}$ | $P_{A S, 3}$ | $P_{A S, 1}$ | $P_{A S, 3}$ |
| $2.11 \cdot 10^{4}$ | $10^{-2}$ | 5,601 (24) | 3,058 (24) | $3.88 \cdot 10^{5}$ | $4.65 \cdot 10^{5}$ | 22.41 | 16.28 |
|  | $10^{-4}$ | , | 2,816 (21) | $3.88 \cdot 10^{5}$ | $4.65 \cdot 10^{5}$ | $\dagger$ | 15.09 |
|  | $10^{-6}$ | 7,510 (24) | 3,525 (23) | $3.88 \cdot 10^{5}$ | $4.65 \cdot 10^{5}$ | 30.10 | 18.49 |
| $8.32 \cdot 10^{4}$ | $\overline{1} \overline{0}^{-2^{-}}$ | 5, $\overline{371}$ - $(\overline{25})$ | $\overline{3}, \overline{12} \overline{4} \overline{(24)}$ | $\overline{2.12} \cdot \overline{10} \overline{0}^{6}$ | -2.21' $\overline{1} \overline{0}^{6}$ | 77.94 | $9 \overline{9} \cdot \overline{1} \overline{8}$ |
|  | $10^{-4}$ | , | 3,358 (23) | $2.12 \cdot 10^{6}$ | $2.21 \cdot 10^{6}$ | $\dagger$ | 101.80 |
|  | $10^{-6}$ | 9,272 (30) | 4,059 (25) | $2.12 \cdot 10^{6}$ | $2.21 \cdot 10^{6}$ | 130.13 | 119.99 |
| $3.30 \cdot 10^{5}$ | $\overline{1} \overline{0}^{-2^{-}}$ | - $\overline{4}, \overline{6} 7 \overline{0}^{-}(\overline{2} \overline{4})$ | $\overline{3}, \overline{185} \overline{5}^{-}(\overline{4})$ | $1.06 \cdot \overline{1} 0^{-7^{7}}$ | $1.09 \cdot 10^{7}$ | $\overline{3} 24.70$ | $4 \overline{7} \overline{0} .9 \overline{9}$ |
|  | $10^{-4}$ |  | 4,569 (26) | $1.06 \cdot 10^{7}$ | $1.09 \cdot 10^{7}$ | $\dagger$ | 615.22 |
|  | $10^{-6}$ | 9,318 (30) | 3,185 (24) | $1.06 \cdot 10^{7}$ | $1.09 \cdot 10^{7}$ | 636.53 | 474.40 |
| $1.32 \cdot 10^{6}$ | $\overline{1} \overline{0}^{-2}$ | - $\overline{3}, \overline{6} 4 \overline{5}(\overline{2} \overline{0})$ | 2,996 (20) | $5.51 \cdot 10^{7}$ | ${ }^{-} 5.388^{-1} \overline{10}^{7}$ | 1,185.78 | 1,930. $\overline{0} 2$ |
|  | $10^{-4}$ | 8,867 (33) | 5,266 (28) | $5.51 \cdot 10^{7}$ | $5.38 \cdot 10^{7}$ | 2,807.26 | 3,082.31 |
|  | $10^{-6}$ | 8,119 (31) | 4,594 (26) | $5.51 \cdot 10^{7}$ | $5.38 \cdot 10^{7}$ | 2,555.12 | 2,765.34 |

for this problem one can observe that the preconditioner $P_{A S, 1}$ was not good enough to deliver fast convergence of IP-PMM consistently. In certain cases, convergence to a 6 -digit solution was not possible, while in others, numerical inaccuracy resulted in an increase in the IP-PMM iterations. On the other hand, when $P_{A S, 3}$ was employed IP-PMM did not face any numerical issues. A possible increase in the overall number of Krylov iterations in this case was connected to an increase in the number of outer IP-PMM iterations. We should note that for all these instances a 4 - or a 5 -digit accurate solution was found very quickly by both approaches. Unlike in the case of the Poisson optimal control problem, we can observe that in this problem using non-diagonal Hessian information within the preconditioner is significantly more important. As before, we should mention that the reported number of Krylov iterations includes the solution of both the predictor and the corrector steps for each IP-PMM iteration.

Robustness of inexact IP-PMM Finally, in Table 5.11 we collect the statistics of the runs of the method using the base preconditioners (i.e. $\widehat{P}_{N E}, P_{A S, 1}$ ) over the entire Netlib and Maros-Mészáros test sets, in order to showcase the robustness of the solver. In particular, we solve each set with increasing accuracy and report the overall success rate of the method, the total time, as well as the total IP-PMM and Krylov iterations. All previous experiments demonstrate that IP-PMM with the proposed preconditioning strategy inherits the reliability of IP-PMM with a direct approach (see Section 3.5.1), while allowing one to control the memory and processing requirements of the method (which is not the case when employing a factorization to solve the resulting Newton systems).

Overall, we observe that each of the presented preconditioning approaches can be very successful on a wide range of problems, including those of very large scale. Although we have treated every problem as if we know nothing about its structure for these numerical tests, our a priori knowledge of the preconditioners and of the problem's structure could in principle aid us in selecting a preconditioner without compromising their "general purpose" nature.

Table 5.11: Robustness of inexact IP-PMM

| Collection | Tol | Solved (\%) | IP-PMM |  |  |  |
| :--- | :---: | ---: | ---: | ---: | ---: | ---: |
|  |  |  | Time (s) | IP-Iter. | Krylov-Iter. |  |
| Netlib | $10^{-4}$ | 100 | $\%$ | 141.25 | 2,907 | 101,482 |
| Netlib | $10^{-6}$ | 100 | $\%$ | 183.31 | 3,083 | 107,911 |
| Netlib | $10^{-8}$ | 96.87 | $\%$ | 337.21 | 3,670 | 119,465 |
| Maros-Mészáros | $10^{-4}$ | $99.21 \%$ | 422.75 | 3,429 | 247,724 |  |
| Maros-Mészáros | $10^{-6}$ | $97.64 \%$ | 545.26 | 4,856 | 291,286 |  |
| Maros-Mészáros | $10^{-8}$ | $92.91 \%$ | 637.35 | 5,469 | 321,636 |  |

### 5.5 Conclusions

In this chapter we have presented several general-purpose preconditioning methodologies suitable for primal-dual regularized interior point methods, applied to convex optimization problems. All presented preconditioners are positive definite and hence can be used within symmetric solvers such as PCG or MINRES. After analyzing and discussing the different preconditioning approaches, we have presented extensive numerical results, showcasing their use and potential benefits for different types of practical applications of convex optimization. A robust and general IP-PMM implementation, using the proposed preconditioners, has been provided for the solution of convex quadratic programming problems, and one can readily observe its ability of reliably and efficiently solving general large-scale problems, with minimal input from the user.

As a future research direction, we would like to construct certain matrix-free preconditioning methodologies that could be used as alternatives for huge-scale instances that cannot be solved by means of factorization-based preconditioners, due to memory requirements.

## Chapter 6

## IP-PMMs for Sparse Approximation Problems

### 6.1 Introduction

Each of the previous chapters contained numerical results on standard test sets. To that end, generic implementations of IP-PMM were employed and used as black-box solvers. In practice, however, it is often the case that the problem under consideration has certain properties that a solver like IP-PMM can exploit, leading to more robust and efficient implementations; albeit less general.

In this chapter, we are concerned with the efficient solution of a wide class of problems which are very large and are expected to yield sparse solutions. In practice, the sparsity is often induced by the presence of $\ell_{1}$ norm terms in the objective. We assume that a general problem of the following form

$$
\begin{array}{rl}
\min _{x} & f(x)+\tau_{1}\|x\|_{1}+\tau_{2}\|L x\|_{1},  \tag{6.1}\\
\text { s.t. } & A x=b,
\end{array}
$$

needs to be solved, where $f: \mathbb{R}^{n} \mapsto \mathbb{R}$ is a twice continuously differentiable convex function, $L \in \mathbb{R}^{l \times n}, A \in \mathbb{R}^{m \times n}, b \in \mathbb{R}^{m}, m \leq n$, and $\tau_{1}, \tau_{2}>0$. The terms $\|x\|_{1}$ and $\|L x\|_{1}$ induce sparsity in the vector $x$ and/or in some (possibly redundant) dictionary $L x$. Numerous real-life problems can be recast into the form (6.1). Among the various application areas, one can find portfolio optimization [114], signal and image processing [42, 150], classification in statistics [166] and machine learning [172], inverse problems [168], PDE-constrained optimization [163], and compressed sensing [35], to name just a few.

Optimization problems arising in these applications are usually solved by different specialized variants of first-order methods. Indeed, highly specialized and tuned to a narrow class of problems, first-order methods often outperform standard off-the-shelf second-order techniques; the latter might be too expensive or might struggle with excessive memory requirements. Such comparisons are not fair though. With this chapter we hope to change the incorrect opinion on the second-order methods.

Various second-order approaches have been proposed in the literature for problems of the form of (6.1). In particular, one might employ proximal (projected)

Newton-type methods (see [106, 157]) in which a proximal term is added to deal with the non-smooth part of the objective function (unless its minimum-norm subgradient has a closed form solution, in which case the proximal term can be excluded). Alternatively, such problems can be solved by means of standard semi-smooth Newton methods (see [101, 103] and the references therein) or semi-smooth Newton methods combined with the augmented Lagrangian method (see, e.g., [108]). The aforementioned approaches employ line-search schemes that allow one to show linear or local superlinear convergence, given certain assumptions. For methods involving proximal terms, superlinear convergence is only guaranteed when the associated penalty parameters increase to infinity.

Here we consider interior point methods, which exhibit better convergence (in practice and in theory), at the expense of worse conditioning of the associated linear systems that have to be solved at every IPM iteration. When efficiently implemented and specialized to a particular application, interior point methods offer an attractive alternative. They can be equally (or more) efficient than the best first-order methods available, and they deliver unmatched robustness and reliability.

The specializations of interior point methods proposed in this chapter do not go beyond what has been commonly exploited by first-order methods. Namely we propose:

- to exploit special features of the problems in the linear algebra of IPMs,
- and to take advantage of the expected sparsity of the optimal solution.

In order to achieve our goals we propose to convert sparse approximation problems to standard smooth nonlinear convex programming ones by replacing the $\ell_{1}$ norm terms with a usual modeling trick in which an absolute value $|a|$ is substituted with the sum of two non-negative parts, $|a|=a^{+}+a^{-}$, where $a^{+}=\max \{a, 0\}$ and $a^{-}=\max \{-a, 0\}$. By introducing the auxiliary variable $d=L x \in \mathbb{R}^{l}$, problem (6.1) is then transformed into the following one:

$$
\begin{array}{cl}
\min _{x^{+}, x^{-}, d^{+}, d^{-}} & f\left(x^{+}-x^{-}\right)+\tau_{1}\left(e_{n}^{\top} x^{+}+e_{n}^{\top} x^{-}\right)+\tau_{2}\left(e_{l}^{\top} d^{+}+e_{l}^{\top} d^{-}\right), \\
\text {s.t. } & A\left(x^{+}-x^{-}\right)=b,  \tag{6.2}\\
& L\left(x^{+}-x^{-}\right)=d^{+}-d^{-}, \\
& x^{+}, x^{-} \geq 0_{n}, \quad d^{+}, d^{-} \geq 0_{l},
\end{array}
$$

where $x^{+}, x^{-} \in \mathbb{R}^{n}$ are such that $x=x^{+}-x^{-}$, and $d^{+}, d^{-} \in \mathbb{R}^{l}$ are such that $d=d^{+}-d^{-}$. It is worth observing that (6.1) and (6.2) are equivalent; indeed the presence of linear terms which penalize for the sum of positive and negative parts of vectors $x$ and $d$ guarantees that at optimality only one of the split variables can take a nonzero value. We also note that the number of variables is greater than or equal to the number of equality constraints in (6.2). Although (6.2) is larger than (6.1) because the variables have been replicated and new constraints have been added, it is in a form eligible to a straightforward application of an interior point method. We expect that the well-known ability of IPMs to handle large sets of linear equality and inequality constraints will compensate for this increase of the problem dimension.

IPMs employ Newton method to solve a sequence of logarithmic barrier subproblems (see Section 1.1.1 for more details). In standard implementations of IPMs this requires many involved linear algebra operations (building and inverting the Hessian matrix), and for large problems it might become prohibitively expensive. In this chapter we demonstrate that the use of inexact Newton method $[16,34,80]$ combined with a knowledgeable choice and appropriate tuning of linear algebra solvers (see $[51,56,78,83]$ and the references therein) is the key to success when developing an IPM specialized to a particular class of problems. We also demonstrate an attractive ability of IPMs to select important variables and prematurely drop the irrelevant ones, a feature which is very well suited to solving sparse approximation problems in which the majority of variables are expected to be zero at optimality. It is worth mentioning at this point that our understanding of the features of IPMs applied to sparse approximation problems benefited from the earlier studies which focused on compressed sensing problems [69, 70].

Ultimately, we provide computational evidence that IPMs can be more efficient than methods which are routinely used for the solution of sparse approximation problems by exploiting only first-order information.

## Structure of the chapter

The rest of this chapter is organized as follows. In Section 6.2 we briefly describe an IP-PMM for non-linear convex programming. In Sections 6.3 to 6.6 we present four applications formulated as optimization problems with sparsity sought in the solutions, and recast them in the form (6.2). In detail, in Section 6.3 we focus on a multi-period portfolio selection strategy, in Section 6.4 on the classification of data coming from functional magnetic resonance imaging (fMRI), in Section 6.5 on the restoration of images corrupted by Poisson noise, and in Section 6.6 on linear classification through regularized logistic regression. The first two applications yield convex quadratic programming problems, while the remaining ones yield general nonlinear convex programming problems. For each application, we provide a brief description of its mathematical model and explain how IP-PMM is specialized for that case in terms of linear algebra solvers, including variable dropping strategies to help sparsification; we also show the results of computational experiments, including comparisons with state-of-the art methods widely used by the scientific community on the selected problems.

### 6.2 An IP-PMM for convex programming

In this section, we derive an IP-PMM suitable for solving convex programming problems. The method is based on the developments in Chapter 3. We consider the following primal problem (which can be equivalently formulated as (CP), by adding some additional constraints):

$$
\begin{equation*}
\min _{x} f(x), \quad \text { s.t. } \quad A x=b, \quad x^{\mathcal{I}} \geq 0_{|\mathcal{I}|}, \quad x^{\mathcal{F}} \text { free, } \tag{6.3}
\end{equation*}
$$

where $f: \mathbb{R}^{n} \mapsto \mathbb{R}$ is a twice continuously-differentiable convex function, $A \in$ $\mathbb{R}^{m \times n}(m \leq n), b \in \mathbb{R}^{m}, x \in \mathbb{R}^{n}, \mathcal{I} \subseteq\{1, \ldots, n\}$, and $\mathcal{F}=\{1, \ldots, n\} \backslash \mathcal{I}$. Generalizing the IP-PMM of Chapter 3, assume that, at some iteration $k$ of the method, we have available an estimate $\eta_{k}$ for an optimal Lagrange multiplier vector $y^{*}$ associated to the equality constraints of (6.3). Similarly, we denote by $\zeta_{k}$ an estimate of a primal solution $x^{*}$. Now, we define the proximal penalty function that has to be minimized at the $k$-th iteration of the PMM, for solving (6.3), given the estimates $\eta_{k}, \zeta_{k}$ :

$$
\mathcal{L}_{\rho_{k}, \delta_{k}}^{P M M}\left(x ; \zeta_{k}, \eta_{k}\right)=f(x)-\eta_{k}^{\top}(A x-b)+\frac{1}{2 \delta_{k}}\|A x-b\|_{2}^{2}+\frac{\rho_{k}}{2}\left\|x-\zeta_{k}\right\|_{2}^{2},
$$

with $\left\{\delta_{k}\right\},\left\{\rho_{k}\right\}$ two positive non-increasing sequences of penalty parameters. Following Chapter 3, we require that these parameters decrease at the same rate as $\mu_{k}$; however, in practice we never allow these values to be reduced below a certain appropriately chosen threshold. We alter the previous penalty function, by including logarithmic barriers, that is

$$
\begin{equation*}
\mathcal{L}_{\rho_{k}, \delta_{k}}^{I P-P M M}\left(x ; \zeta_{k}, \eta_{k}\right)=\mathcal{L}_{\rho_{k}, \delta_{k}}^{P M M}\left(x ; \zeta_{k}, \eta_{k}\right)-\mu_{k} \sum_{j \in \mathcal{I}} \ln x^{j}, \tag{6.4}
\end{equation*}
$$

where $\mu_{k}>0$ is the barrier parameter. We form the optimality conditions of the latter as follows

$$
\nabla f(x)-A^{\top} \eta_{k}+\frac{1}{\delta_{k}} A^{\top}(A x-b)+\rho_{k}\left(x-\zeta_{k}\right)-\mathscr{P}^{\top}\left[\begin{array}{c}
0_{|\mathcal{F}|} \\
\mu_{k}\left(X^{\mathcal{I}}\right)^{-1} e_{|\mathcal{I}|}
\end{array}\right]=0_{n}
$$

where $\mathscr{P}$ is an appropriate permutation matrix, such that $\mathscr{P} x_{k}=\left[\left(x_{k}^{\mathcal{F}}\right)^{\top},\left(x_{k}^{\mathcal{I}}\right)^{\top}\right]^{\top}$. Next, we define the variables $y=\eta_{k}-\frac{1}{\delta_{k}}(A x-b)$ and $z \in \mathbb{R}^{n}$, such that $z^{\mathcal{I}}=\mu_{k}\left(X^{\mathcal{I}}\right)^{-1} e_{|\mathcal{I}|}, z^{\mathcal{F}}=0$, to obtain:

$$
\left[\begin{array}{c}
\nabla f(x)-A^{\top} y-z+\rho_{k}\left(x-\zeta_{k}\right) \\
A x+\delta_{k}\left(y-\eta_{k}\right)-b \\
X^{\mathcal{I}} z^{\mathcal{I}}-\mu_{k} e_{|\mathcal{I}|}
\end{array}\right]=\left[\begin{array}{c}
0_{n} \\
0_{m} \\
0_{|\mathcal{I}|}
\end{array}\right]
$$

To approximately solve the previous mildly nonlinear system of equations, we employ a damped perturbed Newton method (that is, we alter its right-hand side using a centering parameter $\tau_{k} \in(0,1)$ ). Thus, at every iteration of IP-PMM we want to solve the following system of equations:

$$
\begin{gather*}
{\left[\begin{array}{ccc}
-\left(\nabla^{2} f\left(x_{k}\right)+\rho_{k} I_{n}\right) & A^{\top} & I_{n} \\
A & \delta_{k} I_{m} & 0_{m, n} \\
Z_{k} & 0_{n, m} & X_{k}
\end{array}\right]\left[\begin{array}{c}
\Delta x \\
\Delta y \\
\mathscr{P}^{\top}\left[\begin{array}{l}
0_{|\mathcal{F}|} \\
\Delta z^{\mathcal{I}}
\end{array}\right]
\end{array}\right]} \\
=\left[\begin{array}{c}
\nabla f\left(x_{k}\right)-A^{\top} y_{k}+\tau_{k} \rho_{k}\left(x_{k}-\zeta_{k}\right)-z_{k} \\
b-A x_{k}-\tau_{k} \delta_{k}\left(y_{k}-\eta_{k}\right) \\
\mathscr{P}^{\top}\left[\begin{array}{c}
0_{|\mathcal{F}|} \\
\tau_{k} \mu_{k} e_{|\mathcal{I}|}-X_{k}^{\mathcal{I}} z_{k}^{\mathcal{I}}
\end{array}\right]
\end{array}\right] \tag{6.5}
\end{gather*}
$$

From the third block-equation of (6.5) we have $\Delta z^{\mathcal{F}}=0_{|\mathcal{F}|}$ and

$$
\Delta z^{\mathcal{I}}=\left(X_{k}^{\mathcal{I}}\right)^{-1}\left(-Z_{k}^{\mathcal{I}} \Delta x^{\mathcal{I}}+\tau_{k} \mu_{k} e_{|\mathcal{I}|}-X_{k}^{\mathcal{I}} z_{k}^{\mathcal{I}}\right) .
$$

In light of the previous computations, (6.5) reduces to:

$$
\left[\begin{array}{cc}
-\left(\nabla^{2} f\left(x_{k}\right)+\Xi_{k}+\rho_{k} I_{n}\right) & A^{\top}  \tag{6.6}\\
A & \delta_{k} I_{m}
\end{array}\right]\left[\begin{array}{c}
\Delta x \\
\Delta y
\end{array}\right]=\left[\begin{array}{c}
r_{1, k} \\
r_{2, k}
\end{array}\right],
$$

where
and

$$
\Xi_{k}:=\mathscr{P}^{\top}\left[\begin{array}{cc}
0_{|\mathcal{F}|,|\mathcal{F}|} & 0_{|\mathcal{I}|,|\mathcal{F}|} \\
0_{|\mathcal{F}|,|\mathcal{I}|} & \left(X_{k}^{\mathcal{I}}\right)^{-1}\left(Z_{k}^{\mathcal{I}}\right)
\end{array}\right] \mathscr{P} .
$$

For the rest of this chapter, we will make use of the notation $\Xi_{k}$ in cases where only a subset of the primal variables $x$ are constrained to be non-negative. In the case where all the entries of $x$ must satisfy this constraint, we will employ the standard IPM notation $\Theta_{k} \equiv \Xi_{k}^{-1}$, since in this case $\mathcal{F}=\emptyset$. In the special case where $\nabla^{2} f\left(x_{k}\right)$ is a diagonal (or zero) matrix, it could be beneficial to further reduce system (6.6), by eliminating variables $\Delta x$. The resulting normal equations yield a positive definite system of equations that reads as follows:

$$
\begin{align*}
& \left(A\left(\nabla^{2} f\left(x_{k}\right)+\Xi_{k}+\rho_{k} I_{n}\right)^{-1} A^{\top}+\delta_{k} I_{m}\right) \Delta y \\
& \quad=r_{2, k}+A\left(\nabla^{2} f\left(x_{k}\right)+\Theta_{k}^{-1}+\rho_{k} I_{n}\right)^{-1}\left(r_{1, k}\right) \tag{6.8}
\end{align*}
$$

The parameters $\eta_{k}, \zeta_{k}$ are tuned as in Chapter 3. In particular, we set $\eta_{0}=y_{0}$ and $\zeta_{0}=x_{0}$, where $\left(x_{0}, y_{0}, z_{0}\right)$ is the starting point of IP-PMM. Then, at the end of every iteration $k$, we set $\left(\zeta_{k+1}, \eta_{k+1}\right)=\left(x_{k+1}, y_{k+1}\right)$ only if the primal and dual residuals are decreased sufficiently. If the latter is not the case, we set $\left(\zeta_{k+1}, \eta_{k+1}\right)=\left(\zeta_{k}, \eta_{k}\right)$.

It has been demonstrated (see Chapter 3) that IP-PMM using a single Newton step per iteration converges to an $\epsilon$-optimal solution in a number of iterations that is polynomial with respect to the problem size $n$, if $f$ is a convex quadratic function. Furthermore, the latter holds for linear positive semi-definite programming problems, even if one solves the Newton system inexactly, i.e., requiring only the residual norm to be bounded by a suitable multiple of the barrier parameter $\mu_{k}$ (see Chapter 4). Nevertheless, the previous is not proven to hold for the general convex (nonlinear) case. In the latter case, one would have to employ Newton method combined with a line-search or a trust-region strategy (see, e.g., [7, 173]), in order to guarantee the convergence of the method. In all the cases analyzed in this chapter we make use of a simple Mehrotra-type [119] predictor-corrector scheme, which in general is sufficient to produce good directions that allow the method to converge quickly to the optimal solution. In the corrector stage, the right-hand side is approximated by a linearization of the function $f(\cdot)$ that is
being minimized (see [165]).

### 6.2.1 Testing environment

The various specializations of IP-PMM discussed in the following sections have been implemented in MATLAB and compared with MATLAB implementations of state-of-the-art methods for each specific problem. All the tests were run with MATLAB R2019b on an Intel Xeon Platinum 8168 CPU with 192 GB of RAM, available from the magicbox server at the Department of Mathematics and Physics of the University of Campania "L. Vanvitelli".

### 6.3 Portfolio selection problem

Portfolio selection is one of the most central topics in modern financial economics. It deals with the decision problem of how to allocate resources among several competing assets in accordance with the investment objectives. For mediumand long-time horizons, the multi-period strategy is suitable, because it allows the change of the allocation of the capital across the assets, taking into account the evolution of the available information. In a multi-period setting, the investment period is partitioned into $m$ sub-periods, delimited by $m+1$ re-balancing dates $t^{j}$. The decisions are taken at the beginning of each sub-period $\left[t^{j}, t^{j+1}\right)$, $j=1, \ldots, m$, and kept within it. The optimal portfolio is defined by the vector

$$
w=\left[w_{1}^{\top}, w_{2}^{\top}, \ldots, w_{m}^{\top}\right]^{\top},
$$

where $w_{j} \in \mathbb{R}^{s}$ is the portfolio of holdings at time $t^{j}$ and $s$ is the number of assets.
The mean-variance formulation proposed by Markowitz [114] was extended to a multi-period portfolio selection by Li and Ng [107], and in recent years there has been a significant advancement of both theory and methodologies. In a multi-period mean-variance framework, we fix a final target expected return and adopt as risk measure the function obtained by summing the single-period variance terms [43]:

$$
\rho(w)=\sum_{j=1}^{m} w_{j}^{\top} C_{j} w_{j},
$$

where $C_{j} \in \mathbb{R}^{s \times s}$ is the covariance matrix, assumed to be positive definite, estimated at $t^{j}$. A common strategy to estimate Markowitz model parameters is to use historical data as predictive of the future behavior of asset returns. Different regularization techniques have been proposed to deal with ill-conditioning due to asset correlation; in the last years the $\ell_{1}$-regularization has been used to promote sparsity in the solution [48]. It allows investors to reduce the number of positions to be monitored and held and the overall transaction costs. Another useful interpretation of the $\ell_{1}$ norm is related to the amount of short positions (i.e., negative components in the solution), which indicate an investment strategy where an investor is selling borrowed stocks in the open market, expecting that the market will drop, in order to realize a profit. A suitable tuning of the
regularization parameter permits short controlling in both the single- and the multi-period case [45, 48]. However, in the multi-period case, the sparsity in the solution does not guarantee the control of the transaction costs, especially if the pattern of the active positions (i.e., positive components in the solution) completely changes across periods. In this case, sparsity must be introduced in the variation, e.g., by adding an $\ell_{1}$ term involving the differences of the wealth values allocated on the assets between two contiguous re-balancing times. This acts as a penalty on the portfolio turnover, which has the effect of reducing the number of transactions and hence the transaction costs [46, 53].

Thus, we consider the following fused lasso optimization problem for multiperiod portfolio selection [53]:

$$
\begin{align*}
\min _{w} & \frac{1}{2} w^{\top} C w+\tau_{1}\|w\|_{1}+\tau_{2}\|L w\|_{1}, \\
\text { s.t. } & w_{1}^{\top} e_{s}=\xi_{\text {init }},  \tag{6.9}\\
& w_{j}^{\top} e_{s}=\left(e_{s}+r_{j-1}\right)^{\top} w_{j-1}, \quad j=2, \ldots, m, \\
& \left(e_{s}+r_{m}\right)^{\top} w_{m}=\xi_{\text {term }},
\end{align*}
$$

where $n=m s, C=\operatorname{Diag}\left(C_{1}, C_{2}, \ldots, C_{m}\right) \in \mathbb{R}^{n \times n}$ is a block diagonal symmetric positive definite matrix, $\tau_{1}, \tau_{2}>0, L \in \mathbb{R}^{(n-s) \times n}$ is the discrete difference operator representing the fused lasso regularizer, $r_{j} \in \mathbb{R}^{s}$ is the expected return vector at time $t^{j}, \xi_{\text {init }}$ is the initial wealth, and $\xi_{\text {term }}$ is the target expected wealth resulting from the overall investment. The first constraint is the initial budget constraint. The strategy is assumed to be self-financing, as constraints from 2 to $m$ establish; this means that the value of the portfolio changes only because the asset prices change. The $(m+1)$-st constraint defines the expected final wealth. To deal with the non-separability of the objective function in (6.9), we introduce an auxiliary variable $d$, which is constrained to be equal to $L w$, and we equivalently formulate problem (6.9) as follows:

$$
\begin{align*}
\min _{w, d} & \frac{1}{2} w^{\top} C w+\tau_{1}\|w\|_{1}+\tau_{2}\|d\|_{1}, \\
\text { s.t. } & \bar{A} w=\bar{b},  \tag{6.10}\\
& L w=d,
\end{align*}
$$

where the constraint matrix $\bar{A} \in \mathbb{R}^{(m+1) \times n}$ can be interpreted as an $(m+1) \times m$ lower bi-diagonal block matrix, with blocks of dimension $1 \times s$ defined by

$$
\bar{A}^{i, j}=\left\{\begin{array}{cl}
e_{s}^{\top} & \text { if } i=j, \\
-\left(e_{s}+r_{i-1}\right)^{\top} & \text { if } j=i+1, \\
0_{s}^{\top} & \text { otherwise }
\end{array}\right.
$$

and $\bar{b}=\left(\xi_{\text {init }}, 0,0, \ldots, \xi_{\text {term }}\right)^{\top} \in \mathbb{R}^{m+1}$.

### 6.3.1 Specialized IP-PMM for quadratic portfolio optimization problems

Using the standard trick described in Section 6.1, we split $w$ and $d$ into two vectors of the same size, representing the non-negative and non-positive parts of the entries of $w$ and $d$ respectively, i.e., $w=w^{+}-w^{-}$and $d=d^{+}-d^{-}$. Then, problem (6.10) is reformulated as the QP problem given in (CQP), where we set $l=n-s, \bar{n}=2(n+l)=2 s(2 m-1), \bar{m}=m+1+l=(m+1)+s(m-1)$,

$$
\begin{gather*}
x=\left[\left(w^{+}\right)^{\top},\left(w^{-}\right)^{\top},\left(d^{+}\right)^{\top},\left(d^{-}\right)^{\top}\right]^{\top} \in \mathbb{R}^{\bar{n}}, \\
\left.Q=\left[\begin{array}{rr}
C & -C \\
-C & C
\end{array}\right] \begin{array}{c}
0_{2 n, 2 l} \\
0_{2 l, 2 n}
\end{array}\right] \in \mathbb{R}_{2 l, 2 l}^{\bar{n} \times \bar{n}}, \quad A=\left[\begin{array}{ccc}
\bar{A} & -\bar{A} & 0_{(m+1), 2 l} \\
L & -L & {\left[\begin{array}{ll}
-I_{l} & I_{l}
\end{array}\right]}
\end{array}\right] \in \mathbb{R}^{\bar{m} \times \bar{n}}, \\
c=\left[\tau_{1}, \ldots, \tau_{1}, \tau_{2}, \ldots, \tau_{2}\right]^{\top} \in \mathbb{R}^{\bar{n}}, \quad b=\left[\bar{b}^{1}, \ldots, \bar{b}^{m+1}, 0, \ldots, 0\right]^{\top} \in \mathbb{R}^{\bar{m}} . \tag{6.11}
\end{gather*}
$$

## Dropping primal variables

The optimal solution of problem (6.10) is expected to be sparse. On the other hand, in light of the reformulation in the form of (CQP), we anticipate (and verify in practice) that most of the primal variables $w$ attain a zero value close to optimality. Such variables may significantly contribute to the ill conditioning of the matrix in (6.6) (see, e.g., [51, 78] and the references therein). In order to take advantage of this special property displayed by this problem, we employ the following heuristic method, which aims at dropping variables $x^{j}$ which are sufficiently close zero, their seemingly optimal value. This results in better conditioning of the augmented system, whose dimension is also significantly reduced close to optimality, thus decreasing the computational cost of each IPM iteration. In other words, as IP-PMM progresses, we project the problem onto a smaller space. After this reduced problem is solved, its optimal solution is expanded back to the original space by filling all earlier eliminated variables with zeros. This delivers an optimal solution to the original problem.

In particular, we set a threshold value $\epsilon_{\text {drop }}>0$, and a large constant $\xi>0$. At iteration $k=0$, we define a set $\mathcal{V}=\emptyset$. Then, at every iteration $k$ of IP-PMM, we check the following condition, for every $j \in \mathcal{I} \backslash \mathcal{V}$ :

$$
\begin{equation*}
x_{k}^{j} \leq \epsilon_{\text {drop }} \quad \text { and } \quad z_{k}^{j} \geq \xi \cdot \epsilon_{\text {drop }} \quad \text { and } \quad\left(r_{d}\right)_{k}^{j} \leq \epsilon_{\text {drop }}, \tag{6.12}
\end{equation*}
$$

where $\left(r_{d}\right)_{k}^{j}=\left(c-A^{\top} y_{k}+Q x_{k}-z_{k}\right)^{j}$ represents the dual infeasibility corresponding to the $j$-th variable. Any variable that satisfies the latter condition is dropped, that is, we set $x_{k}^{j}=0, \mathcal{V}=\mathcal{V} \cup\{j\}, \mathcal{G}=\mathcal{F} \cup(\mathcal{I} \backslash \mathcal{V})$, we drop $z_{k}^{j}$ and solve

$$
\left[\begin{array}{cc}
-\left(Q^{\mathcal{G}, \mathcal{G}}+\Xi_{k}^{\mathcal{G}, \mathcal{G}}+\rho_{k} I_{|\mathcal{G}|}\right) & \left(A^{\mathcal{H}, \mathcal{G}}\right)^{\top}  \tag{6.13}\\
A^{\mathcal{H}, \mathcal{G}} & \delta_{k} I_{\bar{m}}
\end{array}\right]\left[\begin{array}{c}
\Delta x^{\mathcal{G}} \\
\Delta y
\end{array}\right]=\left[\begin{array}{c}
r_{1, k}^{\mathcal{G}} \\
r_{2, k}
\end{array}\right],
$$

where, $\Xi_{k}$ is defined as in Section 6.2, $r_{1, k}, r_{2, k}$ are defined in (6.7) (by substituting $A^{\mathcal{H}, \mathcal{G}}$ as the constraint matrix), and $\mathcal{H}=\{1, \ldots, \bar{m}\}$. We should note that this
is a heuristic, since once a variable is dropped, it is not considered again until the method converges. Hence, one has to make sure that none of the nonzero variables $x_{k}^{j}$ is dropped. Nevertheless, at the end of the optimization process we can test whether the variables in $\mathcal{V}$ are indeed nonzero. More specifically, once an optimal solution $\left(x^{*}, y^{*}, z^{*}\right)$ is found, we compute:

$$
z^{\mathcal{V}}=c^{\mathcal{V}}-\left(A^{\mathcal{H}, \mathcal{V}}\right)^{\top} y^{*}+Q^{\mathcal{V}, \mathcal{G}}\left(x^{*}\right)^{\mathcal{G}} .
$$

If there exists $j$ such that $\left(z^{\mathcal{V}}\right)^{j} \leq 0$, then we would identify that a variable $x^{j}$ was incorrectly dropped. Otherwise, the optimal solution of the reduced problem coincides with the nonzero part of the optimal solution of the original problem. We notice that this methodology is not new. In particular, a similar strategy was employed in [81], where a special class of linear programming problems was solved using a primal-dual logarithmic barrier method.

### 6.3.2 Computational experience

We test the effectiveness of the IP-PMM applied to the fused lasso model on the following real-world data sets:

1. FF48-FF100 (Fama \& French 48-100 Industry portfolios, USA), containing 48-100 portfolios considered as assets, from July 1926 to December 2015.
2. ES50 (EURO STOXX 50), containing 50 stocks from 9 Eurozone countries (Belgium, Finland, France, Germany, Ireland, Italy, Luxembourg, the Netherlands and Spain), from January 2008 to December 2013.
3. FTSE100 (Financial Times Stock Exchange, UK), containing 100 assets, from July 2002 to April 2016.
4. SP500 (Standard \& Poors, USA), containing 500 assets, from January 2008 to December 2016.
5. NASDAQC (National Association of Securities Dealers Automated Quotation Composite, USA), containing almost all stocks listed on the Nasdaq stock market, from February 2003 to April 2016.

Following [46, 47], we generate 10 problems with annual or quarterly rebalancing, after a preprocessing procedure that eliminates the elements with the highest volatilities. A rolling window (RW) for setting up the model parameters is considered. For each dataset, the length of the RWs is fixed in order to build positive definite covariance matrices and ensure statistical significance. Different datasets require different lengths for the RWs. In Table 6.1 we summarize the information on the test problems.

We introduce some measures to evaluate the goodness of the optimal portfolios versus the benchmark one, in terms of risk, sparsity and transaction costs. We consider as benchmark the multi-period naive portfolio, based on the strategy for which at each rebalancing date the total wealth is equally divided among the assets. We assume that the investor has one unit of wealth at the beginning of

Table 6.1: Characteristics of the portfolio test problems ( $\mathrm{y}=$ years, $\mathrm{m}=$ months)

| Problem | Assets | RW | Sub-periods | $\bar{n}$ |
| :--- | ---: | ---: | ---: | ---: |
| FF48-10 | 48 | 5 y | 10 y | 1632 |
| FF48-20 | 48 | 5 y | 20 y | 3552 |
| FF48-30 | 48 | 5 y | 30 y | 5472 |
| FF100-10 | 96 | 10 y | 10 y | 3264 |
| FF100-20 | 96 | 10 y | 20 y | 7104 |
| FF100-30 | 96 | 10 y | 30 y | 10,944 |
| ES50 | 50 | 1 y | 22 m | 4300 |
| FTSE100 | 83 | 1 y | 10 y | 3154 |
| SP500 | 431 | 2 y | 8 y | 11,206 |
| NASDAQC | 1203 | 1 y | 10 y | 45,714 |

the planning horizon, i.e., $\xi_{\text {init }}=1$, and we set as expected final wealth the one provided by the benchmark. As in [46], we define:

$$
\begin{equation*}
\text { ratio }=\frac{w_{\text {naive }}^{\top} C w_{\text {naive }}}{w_{\text {opt }}^{\top} C w_{\text {opt }}}, \tag{6.14}
\end{equation*}
$$

where $w_{\text {naive }}$ and $w_{\text {opt }}$ are respectively the naive portfolio and the optimal one. This value measures the risk reduction factor with respect to the benchmark. We consider the number of active positions as a measure of holding costs; then the value

$$
\begin{equation*}
\text { ratio }_{h}=\frac{\# \text { active positions of } w_{\text {naive }}}{\# \text { active positions of } w_{o p t}} \tag{6.15}
\end{equation*}
$$

measures the reduction factor of the holding costs with respect to the benchmark. Finally, we consider the number of variations in the weights as a measure of transaction costs. More precisely, if $w_{j}^{i} \neq w_{j+1}^{i}$ we assume that security $i$ has been bought or sold in the period $\left[t^{j}, t^{j+1}\right)$. Then we estimate the number of transactions as:

$$
\mathcal{T}=\operatorname{trace}\left(V^{\top} V\right)
$$

where $V \in \mathbb{R}^{s \times(m-1)}$, with

$$
v^{i j}= \begin{cases}1 & \text { if }\left|w_{j}^{i}-w_{j+1}^{i}\right| \geq \epsilon \\ 0 & \text { otherwise. }\end{cases}
$$

and $\epsilon>0$, in order to make sense in financial terms. A measure of the transaction reduction factor with respect to the benchmark is given by

$$
\begin{equation*}
\text { ratio }_{t}=\frac{\mathcal{T}_{\text {naive }}}{\mathcal{T}_{\text {opt }}} \tag{6.16}
\end{equation*}
$$

We consider a version of the presented IP-PMM algorithm in which the solution of problem (6.13) is computed by means of factorization, the parameter $\epsilon_{\text {drop }}$ controlling the heuristic described in Section 6.3.1 is set to $10^{-4}$, and the constant $\xi$, which is used to ensure that the respective dual slack variable is bounded away
from zero, is set to $10^{2}$. We compare IP-PMM with the split Bregman method, which is known to be very efficient for this kind of problems. In detail, we consider the alternating split Bregman algorithm used in [47], based on a further reformulation of problem (6.10) as

$$
\begin{aligned}
\min _{w, u, d} & \frac{1}{2} w^{\top} C w+\tau_{1}\|u\|_{1}+\tau_{2}\|d\|_{1}, \\
\text { s.t. } & \bar{A} w=\bar{b}, \\
& L w=d \\
& w=u
\end{aligned}
$$

This algorithm splits the minimization in three parts. Given $w_{k}, u_{k}, d_{k}$, the ( $k+1$ )st iteration consists in the minimization of a quadratic function to determine $w_{k+1}$ and the application of the soft-thresholding operator

$$
[\mathcal{S}(v, \gamma)]^{i}=\operatorname{sign}\left(v^{i}\right) \cdot \max \left(\left|v^{i}\right|-\gamma, 0\right),
$$

where $v$ is a real vector and $\gamma>0$, to determine $u_{k+1}$ and $d_{k+1}$. The optimal value $w_{k+1}$ can be obtained by solving the system $H w=p_{k+1}$, with

$$
\begin{equation*}
H=C+\lambda_{1} \bar{A}^{\top} \bar{A}+\lambda_{2} L^{\top} L+\lambda_{3} I_{n}, \tag{6.17}
\end{equation*}
$$

where $\lambda_{1}, \lambda_{2}, \lambda_{3}>0$ are fixed and $p_{k+1}$ depends on the iteration. Since $H$ is independent of the iteration and is symmetric positive definite, sparse, and banded, in [47] the authors compute its sparse Cholesky factorization only once and solve two triangular systems at each iteration. We refer to this algorithm as ASB-Chol.

The values of $\tau_{1}$ and $\tau_{2}$ in (6.9) are selected to guarantee reasonable portfolios in terms of short positions. We recall that from the financial point of view, negative solutions correspond to transactions in which an investor sells borrowed securities in anticipation of a price decline. In our runs we consider the smallest values of $\tau_{1}$ and $\tau_{2}$ that produce at most $2 \%$ of short positions. We set $\tau_{1}=\tau_{2}=$ $10^{-2}$ for the FF48 and FF100 data sets, $\tau_{1}=\tau_{2}=10^{-3}$ for ES50 and SP500, $\tau_{1}=10^{-2}$ and $\tau_{2}=10^{-3}$ for FTSE, and $\tau_{1}=10^{-2}$ and $\tau_{2}=10^{-4}$ for NASDAQC.

In Table 6.2 we present the results obtained with IP-PMM and ASB-Chol on the test problems. The termination criteria of IP-PMM are the same as in Section 3.5.1. The stopping criterion for ASB-Chol is based only on the relative reduction of the primal feasibility $\|\bar{A} x-b\|$, which is a standard choice in literature. The relative tolerance for the two algorithms is $t o l=10^{-6}$, which guarantees that the values of ratio differ by at most $10 \%$, so that both algorithms produce comparable portfolios in terms of risk. We note that the solution computed by ASB-Chol is thresholded by setting to zero all the entries with absolute value not exceeding the same value of $\epsilon_{\text {drop }}$ used in the IP-PMM dropping strategy. The results show that the optimal portfolios computed by IP-PMM and ASB-Chol outperform the benchmark ones in terms of all the metrics. Concerning ratio ${ }_{h}$ and ratio ${ }_{t}$, IPPMM is generally able to produce greater values than ASB-Chol, which indicates a higher sparsity in the solution found by IP-PMM. IP-PMM generally performs comparably or better than ASB-Chol in terms of elapsed time. Although ASB-

Table 6.2: IP-PMM vs ASB-Chol

|  | IP-PMM |  |  |  |  |
| :--- | ---: | ---: | ---: | ---: | ---: |
| Problem | Time (s) | Iters | ratio | ratio | ratio |
| FF48-10 | $1.37 \mathrm{e}-1$ | 12 | $2.32 \mathrm{e}+0$ | $6.67 \mathrm{e}+0$ | $1.66 \mathrm{e}+1$ |
| FF48-20 | $3.77 \mathrm{e}-1$ | 16 | $2.28 \mathrm{e}+0$ | $6.58 \mathrm{e}+0$ | $2.13 \mathrm{e}+1$ |
| FF48-30 | $8.43 \mathrm{e}-1$ | 21 | $4.64 \mathrm{e}+0$ | $6.15 \mathrm{e}+0$ | $1.69 \mathrm{e}+1$ |
| FF100-10 | $4.92 \mathrm{e}-1$ | 12 | $1.58 \mathrm{e}+0$ | $1.78 \mathrm{e}+1$ | $4.36 \mathrm{e}+1$ |
| FF100-20 | $1.63 \mathrm{e}+0$ | 15 | $1.81 \mathrm{e}+0$ | $2.04 \mathrm{e}+1$ | $4.92 \mathrm{e}+1$ |
| FF100-30 | $3.93 \mathrm{e}+0$ | 21 | $5.82 \mathrm{e}+0$ | $1.34 \mathrm{e}+1$ | $3.60 \mathrm{e}+1$ |
| ES50 | $4.59 \mathrm{e}-1$ | 14 | $2.12 \mathrm{e}+0$ | $4.42 \mathrm{e}+0$ | $5.75 \mathrm{e}+1$ |
| FTSE100 | $4.64 \mathrm{e}-1$ | 14 | $1.85 \mathrm{e}+0$ | $5.37 \mathrm{e}+1$ | $6.09 \mathrm{e}+1$ |
| SP500 | $3.43 \mathrm{e}+1$ | 16 | $1.57 \mathrm{e}+0$ | $8.62 \mathrm{e}+1$ | $1.50 \mathrm{e}+2$ |
| NASDAQC | $7.05 \mathrm{e}+2$ | 20 | $3.15 \mathrm{e}+0$ | $2.73 \mathrm{e}+0$ | $3.89 \mathrm{e}+2$ |
|  |  |  | ASB-Chol |  |  |
| Problem | Time (s) | Iters | ratio | ratio | ratio |
| FF48-10 | $1.67 \mathrm{e}-1$ | 1431 | $2.33 \mathrm{e}+0$ | $6.67 \mathrm{e}+0$ | $1.66 \mathrm{e}+1$ |
| FF48-20 | $3.72 \mathrm{e}-1$ | 1985 | $2.31 \mathrm{e}+0$ | $7.93 \mathrm{e}+0$ | $2.09 \mathrm{e}+1$ |
| FF48-30 | $1.12 \mathrm{e}+0$ | 4125 | $4.64 \mathrm{e}+0$ | $6.08 \mathrm{e}+0$ | $1.66 \mathrm{e}+1$ |
| FF100-10 | $8.49 \mathrm{e}-1$ | 3087 | $1.58 \mathrm{e}+0$ | $1.78 \mathrm{e}+1$ | $4.36 \mathrm{e}+1$ |
| FF100-20 | $2.09 \mathrm{e}+0$ | 3635 | $1.80 \mathrm{e}+0$ | $1.78 \mathrm{e}+1$ | $4.27 \mathrm{e}+1$ |
| FF100-30 | $8.54 \mathrm{e}+0$ | 9043 | $5.83 \mathrm{e}+0$ | $1.12 \mathrm{e}+1$ | $2.97 \mathrm{e}+1$ |
| ES50 | $9.70 \mathrm{e}-1$ | 4297 | $2.05 \mathrm{e}+0$ | $2.94 \mathrm{e}+0$ | $4.26 \mathrm{e}+1$ |
| FTSE100 | $4.29 \mathrm{e}-1$ | 1749 | $1.80 \mathrm{e}+0$ | $5.07 \mathrm{e}+1$ | $5.71 \mathrm{e}+1$ |
| SP500 | $1.98 \mathrm{e}+1$ | 3728 | $1.74 \mathrm{e}+0$ | $6.16 \mathrm{e}+1$ | $1.01 \mathrm{e}+2$ |
| NASDAQC | $8.84 \mathrm{e}+2$ | 14264 | $3.15 \mathrm{e}+0$ | $2.73 \mathrm{e}+0$ | $3.89 \mathrm{e}+2$ |

Chol is faster than IP-PMM on SP500 by 14.5 seconds (42\%), IP-PMM is able to reach a better solution in terms of sparsity. When applied to FF100-30 IPPMM produces a portfolio associated with lower transaction costs and takes less than half of the time required by ABS-Chol. When applied to NASDAQC, which is the largest problem under consideration, the two methods reach comparable solutions in terms of all the evaluation metrics, but IP-PMM needs about $20 \%$ less time ( 179 seconds) than ASB-Chol. This suggests that the use of IP-PMM can be beneficial especially when solving high-dimensional problems.

### 6.4 Classification models for functional magnetic resonance imaging data

The functional magnetic resonance imaging (fMRI) technique measures brain spatio-temporal activity via blood-oxygen-level-dependent (BOLD) signals. Starting from the assumption that neuronal activity is coupled with cerebral blood flow, fMRI signals have been used to identify regions associated with functions such as speaking, vision, movement, etc.. By analyzing the different oxigenation
levels in specific areas of the brain of healthy and ill patients, in the last decades fMRI signals have been used to investigate the effect on the brain functionality of tumors, strokes, head and brain injuries and of cognitive disorders such as schizophrenia or Alzheimer's (see [57, 99, 120] and the references therein).

In an fMRI scan, voxels representing regions of the brain of a patient are recorded at different time instances. The temporal resolution is usually in the order of a few seconds, while the spatial resolution generally ranges from $4-5 \mathrm{~mm}$ (for some full brain analyses) to 1 mm (for analyses on specific brain regions), which may amount up to about a million voxels. Since fMRI experiments are conducted over groups of patients, the dimensionality of the data is further increased. Therefore, the interpretation of fMRI results requires the analysis of huge quantities of data. To this aim, machine learning techniques are being increasingly used in recent years, because of their capability of dealing with massive amounts of data, incorporating also a-priori information about the problems they are targeted to $[12,13,57,61,84,120,149]$.

Here we focus on the problem of training a binary linear classifier to distinguish between different classes of patients (e.g., ill/healthy) or different kinds of stimuli (e.g., pleasant/unpleasant), and to get information about the most significant brain areas associated with the related neural activity. The two classes are identified by the labels -1 and 1 . We assume that the training set consists of $s_{-1}$ 3 -dimensional (3d) scans in class -1 and $s_{1} 3 \mathrm{~d}$ scans in class 1 , where each 3 d scan is reshaped as a row vector of size $q=q_{1} \times q_{2} \times q_{3}$, and $q_{i}$ is the number of voxels along the $i$-th coordinate direction of the domain covering the brain. All the scans are stored as rows of a matrix $D \in \mathbb{R}^{s \times q}$, where $s=s_{-1}+s_{1}$.

We use a square loss function with the aim of determining an unbiased hyperplane in $\mathbb{R}^{q}$ that can separate the patients in the two classes. This leads to a minimization problem of the form:

$$
\begin{equation*}
\min \frac{1}{2 s}\|D w-\hat{y}\|^{2}, \tag{6.18}
\end{equation*}
$$

where $\hat{y}$ is a vector containing the labels associated with each scan. Notice that the use of the Euclidean loss is a standard practice in the literature for the classification of fMRI data (e.g. see [13, 84, 87, 98, 109]). Nevertheless, it should be noted that different loss functions could be employed as well (e.g. see [152, 182]), potentially leading to better classification accuracy in certain cases.

Since the number of patients is usually much smaller than the size of a scan, i.e., $s \ll q$, problem (6.18) is strongly ill posed and thus requires regularization. Recently, significant attention has been given to regularization terms encouraging the presence of structured sparsity, where smoothly varying nonzero coefficients of the solution are associated with small contiguous regions of the brain. This is motivated by the possibility of obtaining more interpretable solutions than those corresponding to other regularizers that do not promote sparsity or lead to sparse solutions without any structure (see [13, 84, 109] and the references therein).

Structured sparsity can be promoted, e.g., by using a combination of $\ell_{1}$ and anisotropic total variation (TV) terms [3], which can be regarded as a fused lasso
regularizer [166]. The regularized problem reads

$$
\begin{equation*}
\min _{w} \frac{1}{2 s}\|D w-\hat{y}\|^{2}+\tau_{1}\|w\|_{1}+\tau_{2}\|L w\|_{1} \tag{6.19}
\end{equation*}
$$

where $\|L w\|_{1}$ is the discrete anisotropic TV of $w$, i.e., $L=\left[\begin{array}{lll}L_{x}^{\top} & L_{y}^{\top} & L_{z}^{\top}\end{array}\right]^{\top} \in \mathbb{R}^{l \times q}$ is the matrix representing first-order forward finite differences in the $x, y, z$ directions at each voxel. By penalizing the difference between each voxel and its neighbors in each direction, one enforces the weights of the classification hyperplane (which share the 3d structure of the scans) to assume similar values for contiguous regions of the brain, thus leading to identify whole regions of the brain involved in the decision process.

The previous problem can be reformulated by introducing the variables $u=$ $D w$ and $d=L w$ and applying the splitting

$$
w=w^{+}-w^{-}, d=d^{+}-d^{-}, \quad\left(w^{+}, w^{-}, d^{+}, d^{-}\right) \geq\left(0_{q}, 0_{q}, 0_{l}, 0_{l}\right)
$$

Let $m=l+s$ and $n=s+2 q+2 l$. Using the previous variables, (6.19) can be equivalently written as:

$$
\begin{align*}
\min _{x} & \frac{1}{2} x^{\top} Q x+c^{\top} x \\
\text { s.t. } & A x=b,  \tag{6.20}\\
& x_{\mathcal{I}} \geq 0_{|\mathcal{I}|}, x_{\mathcal{F}} \text { free, } \mathcal{I}=\{s+1, \ldots, n\}, \mathcal{F}=\{1, \ldots, s\},
\end{align*}
$$

where $b=0_{s+l} \in \mathbb{R}^{m}$,
$x=\left[u^{\top},\left(w^{+}\right)^{\top},\left(w^{-}\right)^{\top},\left(d^{+}\right)^{\top},\left(d^{-}\right)^{\top}\right]^{\top}, \quad c=\left[-\frac{\hat{y}^{\top}}{s}, \tau_{1} e_{w}^{\top}, \tau_{1} e_{w}^{\top}, \tau_{2} e_{d}^{\top}, \tau_{2} e_{d}^{\top}\right]^{\top} \in \mathbb{R}^{n}$,
and

$$
Q=\left[\begin{array}{cc}
\frac{1}{s} I_{s} & 0_{s,(n-s)}  \tag{6.21}\\
0_{(n-s), s} & 0_{(n-s),(n-s)}
\end{array}\right] \in \mathbb{R}^{n \times n}, \quad A=\left[\begin{array}{ccccc}
-I_{s} & D & -D & 0_{s, l} & 0_{s, l} \\
0_{l, s} & L & -L & -I_{l} & I_{l}
\end{array}\right] \in \mathbb{R}^{m \times n} .
$$

### 6.4.1 Specialized IP-PMM for fused lasso least squares

Notice that problem (6.20) is in the same form as (6.3). In what follows, we present a specialized inexact IP-PMM, suitable for solving unconstrained fused lasso least squares problems. The proposed specialized IP-PMM is characterized by the two following implementation details. Firstly, instead of factorizing system (6.6), we employ an iterative method (namely, the preconditioned conjugate gradient method [93]) to solve system (6.8). Secondly, as suggested in Section 6.3.1, we take advantage of the fact that the optimal solution of problem (6.20) is expected to be sparse, and use the heuristic approach that allows us to drop many of the variables of the problem, when the method is close to the optimal solution.

## Solving the Newton system

We focus on solving the normal equations in (6.8). Let $k$ denote an arbitrary iteration of IP-PMM. We re-write the matrix in (6.8) without using the succinct notation introduced earlier:

$$
M_{k}=\left[\begin{array}{ll}
M_{1, k} & M_{2, k}^{\top}  \tag{6.22}\\
M_{2, k} & M_{3, k}
\end{array}\right]
$$

where:

$$
\begin{align*}
M_{1, k}= & \left(\left(\frac{1}{s}+\rho_{k}\right)^{-1}+\delta_{k}\right) I_{s}+D\left(\left(\Xi_{w^{+}, k}+\rho_{k} I_{q}\right)^{-1}+\left(\Xi_{w^{-}, k}+\rho_{k} I_{q}\right)^{-1}\right) D^{\top}, \\
M_{2, k}= & L\left(\Xi_{w^{+}}+\rho_{k} I_{q}\right)^{-1} D^{\top}+L\left(\Xi_{w^{-}, k}+\rho_{k} I_{q}\right)^{-1} D^{\top}, \\
M_{3, k}= & L\left(\left(\Xi_{w^{+}, k}+\rho_{k} I_{q}\right)^{-1}+\left(\Xi_{w^{-}, k}+\rho_{k} I_{q}\right)^{-1}\right) L^{\top} \\
& +\left(\left(\Xi_{d^{+}, k}+\rho_{k} I_{l}\right)^{-1}+\left(\Xi_{d^{-}, k}+\rho_{k} I_{l}\right)^{-1}+\delta_{k} I_{l}\right), \tag{6.23}
\end{align*}
$$

while

$$
\Xi_{k}^{\mathcal{I}}=\left[\begin{array}{cccc}
\Xi_{w^{+}, k} & 0_{q, q} & 0_{q, l} & 0_{q, l} \\
0_{q, q} & \Xi_{w^{-}, k} & 0_{q, l} & 0_{q, l} \\
0_{l, q} & 0_{l, q} & \Xi_{d^{+}, k} & 0_{l, l} \\
0_{l, q} & 0_{l, q} & 0_{l, l} & \Xi_{d^{-}, k}
\end{array}\right],
$$

and $\Xi_{k}$ is defined as in Section 6.2.
Notice that the matrix $D$ in (6.18) is dense, and hence we expect $M_{1, k}$ and $M_{2, k}$ in (6.23) to also be dense. On the other hand, $M_{3, k}$ remains sparse, and we know that $l \gg s$. As a consequence, the Cholesky factors of the matrix in (6.22) would inevitably contain dense blocks. Hence, it might be prohibitively expensive to compute such a decomposition. Instead, we solve the previous system using a PCG method. In order to do so efficiently, we must find an approximation for the coefficient matrix in (6.22). Given the fact that $M_{3, k}$ is sparse, while $M_{1, k}$ and $M_{2, k}$ are dense, we would like to find an approximation for the dense blocks. Based on the assumption $l \gg s$, we can approximate $M_{k}$ by the following block-diagonal preconditioner:

$$
P_{k}=\left[\begin{array}{cc}
M_{1, k} & 0_{s, l}  \tag{6.24}\\
0_{l, s} & M_{3, k}
\end{array}\right] \text {, where, } P_{k}^{-1}=\left[\begin{array}{cc}
M_{1, k}^{-1} & 0_{s, l} \\
0_{l, s} & M_{3, k}^{-1}
\end{array}\right] \text {. }
$$

We notice that $M_{3, k}$ does have a sparse Cholesky factor, due to the sparsity displayed in the discrete anisotropic TV matrix $L$. On the other hand, the Cholesky factors of $M_{1, k}$ are dense. However, computation and storage of these dense factors is possible, as we only need to perform $O\left(s^{3}\right)$ operations, and store $O\left(s^{2}\right)$ elements. Let us observe that the preconditioner in (6.24) is exactly the row-sparsifying preconditioner proposed in Section 5.2. Indeed, this preconditioner employs the strategy of sparsifying certain dense rows of the normal equations matrix, and thus the spectral analysis of the preconditioned system follows directly by Theorem $5 \cdot 2.1$ (by setting $I_{2}$ as $I_{2}=[1,2]$ ).

## Dropping primal variables

The preconditioner (6.24) may be computed (and applied) very efficiently as we expect the Cholesky factor of $M_{3, k}$ to preserve sparsity and $M_{1, k} \in \mathbb{R}^{s \times s}$ to be relatively small (recall that $s \ll l$ ). We deduce from Theorem 5.2.1 that the preconditioner defined in (6.24) remains effective as long as the regularization parameters $\rho_{k}$ and $\delta_{k}$ are not too small. However, to attain convergence of IPPMM $\rho_{k}$ and $\delta_{k}$ have to be reduced and then, due to the nature of IPMs, the matrix in (6.8) becomes increasingly ill conditioned as the method approaches the optimal solution. This implies that the preconditioner defined in (6.24) has only a limited applicability. In particular, this means that there is a limited scope for refining it and we may not be able to prevent degrading behaviour of PCG when IPM gets very close to the optimal solution.

However, we notice that the optimal solution of problem (6.19) is expected to be sparse. Like in the portfolio optimization problem, in light of the reformulation (6.20), we know that most of the primal variables $x$ converge to zero. Close to optimality the presence of such variables would adversely affect the conditioning of the matrix in (6.8). To prevent that, we employ a heuristic similar to the one introduced in Section 6.3.1 which consists of eliminating variables which approach zero and have an associated Lagrange multiplier bounded away from zero.

Given $\epsilon_{\text {drop }}>0, \xi>0$ and $\mathcal{V}=\emptyset$, at every iteration $k$ of IP-PMM, we add to $\mathcal{V}$ each variable $j \in \mathcal{I} \backslash \mathcal{V}$ satisfying condition (6.12), and replace (6.8) with the reduced system

$$
\begin{align*}
&\left(A^{\mathcal{H}, \mathcal{G}}\left(Q^{\mathcal{G}, \mathcal{G}}+\Xi_{k}^{\mathcal{G}, \mathcal{G}}+\rho_{k} I_{|\mathcal{G}|}\right)^{-1}\left(A^{\mathcal{H}, \mathcal{G}}\right)^{\top}+\delta_{k} I_{m}\right) \Delta y \\
&=r_{2, k}+A^{\mathcal{H}, \mathcal{G}}\left(Q^{\mathcal{G}, \mathcal{G}}+\Xi_{k}^{\mathcal{G}, \mathcal{G}}+\rho_{k} I_{|\mathcal{G}|}\right)^{-1} r_{1, k}^{\mathcal{G}}, \tag{6.25}
\end{align*}
$$

where $\mathcal{H}=\{1, \ldots, m\}, \mathcal{G}=\mathcal{F} \cup(\mathcal{I} \backslash \mathcal{V})$, and $r_{1, k}, r_{2, k}$ are defined in (6.7) (with constraint matrix $\left.A^{\mathcal{H}, \mathcal{G}}\right)$.

### 6.4.2 Computational experience

We consider a dataset consisting of fMRI scans for 16 male healthy US college students (age 20 to 25), with the aim of analyzing two active conditions: viewing unpleasant and pleasant images [125]. The preprocessed and registered data ${ }^{1}$ consist of 1344 scans of size 122,128 voxels (only voxels with probability greater than 0.5 of being in the gray matter are considered), with 42 scans considered per subject and active condition (i.e., 84 scans per subject in total).

In order to assess the performance of the IP-PMM on this type of problems, we carry out a comparison with two state-of-the-art algorithms for the solution of problem (6.19):

[^4]- FISTA. As done for the tests in [13], problem (6.19) is reformulated as

$$
\min _{w} \frac{1}{2 s}\|D w-\hat{y}\|^{2}+\|\hat{L} w\|_{1},
$$

where $\hat{L}=\left[\begin{array}{ll}\tau_{1} I_{q} & \tau_{2} L^{\top}\end{array}\right]^{\top}$, and solved by a version of FISTA [15] in which the proximal operator associated with $\|\hat{L} w\|_{1}$ is approximated by 10 steps of an inner FISTA cycle.

- ADMM. We consider the ADMM method [32] applied to the problem

$$
\begin{aligned}
\min _{w, u, d} & \frac{1}{2 s}\|D w-\hat{y}\|^{2}+\tau_{1}\|u\|_{1}+\tau_{2}\|d\|_{1}, \\
\text { s.t. } & w-u=0_{q}, \\
& L w-d=0_{l},
\end{aligned}
$$

in which the minimization of the quadratic function associated with the update of $w$ is approximated by 10 steps of the CG algorithm.

In Table 6.3 we show the results obtained by applying the algorithms to the solution of the fMRI data classification problem. For each choice of the pair of regularization parameters $\left(\tau_{1}, \tau_{2}\right)$, we report the average results obtained in a leave-one-subject-out (LOSO) cross-validation test over the full dataset of patients. This consists in using the data concerning 1 patient as the validation set and the data concerning the remaining patients as the training set. Because of this setting, for each problem the size of $w$ is $q=122,128$, the number of rows of $D$ is $s=1260$, and the dimension of $d=L w$ is $l=339,553$.

By preliminary experiments the choice $\tau_{1}=\tau_{2}$ appeared the most appropriate. Furthermore, for the IP-PMM, the parameters $\epsilon_{\text {drop }}$ and $\xi$ controlling the heuristic described in Section 6.4.1 are set to $10^{-6}$ and $10^{2}$, respectively. To perform a fair comparison between the three algorithms, we consider a stopping criterion based on the execution time, which, after some preliminary tests, is fixed to 30 minutes. The solution of the normal equations system (6.25) is computed by the MATLAB pcg function, for which we set the maximum number of iterations to 600 and the tolerance as

$$
t o l= \begin{cases}10^{-4} & \text { if }\left\|r_{y, k}\right\|<1, \\ \max \left\{10^{-8}, \frac{10^{-4}}{\left\|r_{y, k}\right\|}\right\} & \text { otherwise },\end{cases}
$$

where $r_{y, k}$ is the right-hand side of equation (6.25). If the maximum number of pcg iterations is reached, we switch to the low-rank update strategy proposed in Section 5.2.1 (in order to guarantee efficiency of the method).

For each algorithm tested, we report the mean and the standard deviation for three quality measures of the solution: classification accuracy (ACC), solution density (DEN) and corrected pairwise overlap (CORR OVR) (see [13, Section 2.3.3]). Let $N_{f}$ be the number of folders in the cross validation setting and let $w_{i}$ be a given approximate solution to the problem associated with the $i$-th folder. For each $w_{i}$ we define the accuracy (ACC) as the percentage of test vectors correctly classified by the linear model identified by $w_{i}$. Given a vector $v \in \mathbb{R}^{q}$, we define $\mathcal{Z}(v)$ as the set of indices corresponding to the nonzero components in
$v$ and $\mathcal{D}(v)=|\mathcal{Z}(v)| / q$ as the density of $v$. Hence, for each $w_{i}$ the density (DEN) is computed as $\mathcal{D}\left(w_{i}\right)$. Finally, given any pair of indices $i, j \in\left\{1, \ldots, N_{f}\right\}$, the corrected pairwise overlap is defined as

$$
\mathcal{O}_{i, j}^{c}=\frac{\left|\mathcal{Z}\left(w_{i}\right) \cap \mathcal{Z}\left(w_{j}\right)\right|-E}{\max \left\{\left|\mathcal{Z}\left(w_{i}\right)\right|,\left|\mathcal{Z}\left(w_{j}\right)\right|\right\}},
$$

where $E$ is the expected overlap between the support of two random vectors with density equal to $\mathcal{D}\left(w_{i}\right)$ and $\mathcal{D}\left(w_{j}\right)$, respectively, which is given by $E=$ ${ }_{q} \mathcal{D}\left(w_{i}\right) \mathcal{D}\left(w_{j}\right)$. We observe that the corrected pairwise overlap, which may be the less common in the field of machine learning, is meant to measure the "stability" of the voxel selection. The three metrics are computed after thresholding the solution, as in [13]: after sorting the entries by their increasing magnitude, we set to zero the entries contributing at most to $0.01 \%$ of the $\ell_{1}$-norm of the solution.

Table 6.3: Comparison of IP-PMM, FISTA and ADMM in terms of the LOSO cross-validation scores

| Algorithm | $\tau_{1}=\tau_{2}$ | ACC | DEN | CORR OVR |
| :--- | ---: | ---: | ---: | ---: |
| IP-PMM | $10^{-2}$ | $86.16 \pm 7.11$ | $20.56 \pm 6.63$ | $43.47 \pm 9.09$ |
|  | $5 \cdot 10^{-2}$ | $84.90 \pm 4.80$ | $3.77 \pm 0.84$ | $62.70 \pm 10.39$ |
|  | $10^{-1}$ | $82.29 \pm 6.22$ | $2.49 \pm 0.34$ | $82.60 \pm 9.24$ |
| FISTA | $10^{-2}$ | $86.90 \pm 5.01$ | $88.97 \pm 0.71$ | $5.43 \pm 0.43$ |
|  | $5 \cdot 10^{-2}$ | $84.15 \pm 5.92$ | $19.36 \pm 0.86$ | $65.50 \pm 2.68$ |
|  | $10^{-1}$ | $81.62 \pm 7.58$ | $5.14 \pm 0.44$ | $80.44 \pm 5.72$ |
| ADMM | $10^{-2}$ | $86.46 \pm 6.91$ | $98.70 \pm 0.03$ | $0.03 \pm 0.01$ |
|  | $5 \cdot 10^{-2}$ | $85.57 \pm 5.37$ | $97.97 \pm 0.05$ | $0.15 \pm 0.04$ |
|  | $10^{-1}$ | $82.07 \pm 6.51$ | $97.50 \pm 0.19$ | $0.26 \pm 0.13$ |

By looking at Table 6.3, one can see that IP-PMM appears to be generally better than the other algorithms in enforcing the structured sparsity of the solution, presenting a good level of sparsity and overlap. It is worth noting that, because of its definition, the corrected pairwise overlap tends to zero as the density goes towards $100 \%$. Hence, for ADMM, which seems to be unable to enforce sparsity in the solution, the overlap is close to zero. As suggested in [13], one can evaluate the results in terms of the distance of the pair (ACC, CORR OVR) from the pair $(100,100)$ (the smaller the distance, the better the results). For the tests reported in the table, we can see that the best scores are obtained by IP-PMM with regularization parameters $\tau_{1}=\tau_{2}=10^{-1}$, for which the average accuracy is $82.3 \%$ and the corrected overlap is $82.6 \%$ with an average solution density of $2.5 \%$. To further evaluate the efficiency of IP-PMM in the solution of this class of problems, we compare its performance in terms of elapsed time against the performance of FISTA on the problem where the two methods reach the best scores, i.e., with $\tau_{1}=\tau_{2}=10^{-1}$. For all the 16 instances of the LOSO cross validation, we store the current solution of each algorithm after every minute and, at the end of the execution, we compute the three quality measures for such intermediate solutions. The results are shown in Figure 6.1 in terms of history of the


Figure 6.1: History of classification accuracy, solution density and corrected pairwise overlap for IP-PMM (left) and FISTA (right), in the case $\tau_{1}=\tau_{2}=10^{-1}$. For the three quantities we report average measures with $95 \%$ confidence intervals.
mean values (lines) together with their $95 \%$ confidence intervals (shaded regions). From the plots we can see that while FISTA reaches the measures reported in Table 6.3 at the end of the 30-minute run, the performance of IP-PMM stabilizes after about 20 minutes. At the 20 minutes mark we observe that for IP-PMM the value of each of the three metrics is the same as the one reported in Table 6.3. For FISTA, while accuracy (81.32\%) and overlap (80.54\%) have similar values as those reported in the table, we observe a larger density ( $6.83 \%$ ).

### 6.5 TV-based Poisson image restoration

Next we consider the restoration of images corrupted by Poisson noise, which arises in many applications, such as fluorescence microscopy, computed tomography (CT) and astronomical imaging (see, e.g., [55] and the references therein). In the discrete formulation of the restoration problem, the object to be restored is represented by a vector $w \in \mathbb{R}^{n}$ and the measured data are assumed to be a vector $g \in \mathbb{N}_{0}^{m}$, whose entries $g^{j}$ are samples from $m$ independent Poisson random variables $G^{j}$ with probability

$$
\mathbb{P}\left(G^{j}=g^{j}\right)=\frac{e^{-(D w+a)^{j}}\left[(D w+a)^{j}\right]^{g^{j}}}{g^{j}!}
$$

where $a \in \mathbb{R}_{+}^{m}$ models the background radiation detected by the sensors. The matrix $D=\left(d^{i j}\right) \in \mathbb{R}^{m \times n}$ models the functioning of the imaging system and satisfies

$$
d^{i j} \geq 0 \text { for all } i, j, \quad \sum_{i=1}^{m} d^{i j}=1 \text { for all } j .
$$

Here we assume that $D$ represents a convolution operator with periodic boundary conditions, which implies that $D$ has a block-circulant structure with circulant blocks (BCCB). Hence, $D w$ is computed expeditiously using the 2-dimensional fast Fourier transform (FFT). The maximum-likelihood approach [26] for the
estimation of $u$ leads to the minimization of the Kullback-Leibler (KL) divergence of $D w+a$ from $g$ :

$$
\begin{equation*}
D_{K L}(w) \equiv D_{K L}(D w+a, g)=\sum_{j=1}^{m}\left(g^{j} \ln \frac{g^{j}}{(D w+a)^{j}}+(D w+a)^{j}-g^{j}\right) \tag{6.26}
\end{equation*}
$$

where we set $g^{j} \ln \left(g^{j} /(D w+a)^{j}\right)=0$ if $g^{j}=0$ (we implicitly assume that $g$ has been converted into a real vector with entries ranging in the same interval as the entries of $w$ ). Since the estimation problem is highly ill conditioned, a regularization term is added to (6.26). We consider the total variation [150], which has received considerable attention because of its ability of preserving edges and smoothing flat areas of the images. Notice that, while it may introduce staircase artifacts, TV is still applied in many medical and biological applications (see, e.g., $[14,124,187]$ and J. Huang's webpage ${ }^{2}$ ). The feasible set of the problem is defined by non-negativity constraints on the image intensity and the linear constraint $\sum_{i=1}^{n} w^{i}=\sum_{j=1}^{m}\left(g^{j}-a^{j}\right) \equiv r$ which guarantees preservation of the total intensity of the image.

The resulting model is

$$
\begin{array}{cl}
\min _{w} & D_{K L}(w)+\lambda\|L w\|_{1} \\
\text { s.t. } & e_{n}^{\top} w=r,  \tag{6.27}\\
& w \geq 0_{n},
\end{array}
$$

where $L \in \mathbb{R}^{l \times n}$ is the matrix arising from the discretization of the TV functional (as in [41]).

### 6.5.1 Specialized IP-PMM for image restoration problems

By employing the splitting strategy used in the previous sections, we can transform problem (6.27) to the following equivalent form:

$$
\begin{array}{cl}
\min _{x} & f(x) \equiv D_{K L}(w)+c^{\top} u, \\
\text { s.t. } & A x=b  \tag{6.28}\\
& x \geq 0_{\bar{n}}
\end{array}
$$

where, after introducing the additional constraint $d=L w$, and letting $\bar{m}=l+1$, $\bar{n}=n+2 l$, we set $x=\left[w^{\top}, u^{\top}\right]^{\top} \in \mathbb{R}^{\bar{n}}, u=\left[\left(d^{+}\right)^{\top},\left(d^{-}\right)^{\top}\right]^{\top} \in \mathbb{R}^{2 l}, c=\lambda e_{2 l}$, $b=\left[r, 0_{l}^{\top}\right]^{\top} \in \mathbb{R}^{\bar{m}}$, and

$$
A=\left[\begin{array}{ccc}
e_{n}^{\top} & 0_{l}^{\top} & 0_{l}^{\top} \\
L & -I_{l} & I_{l}
\end{array}\right] \in \mathbb{R}^{\bar{m} \times \bar{n}}
$$

We solve problem (6.28) by using IP-PMM combined with a perturbed composite Newton method [165]. Following the presentation in Section 6.2, we know that at the $k$-th iteration of the method we have to solve two linear systems of the

[^5]form of (6.6). In order to avoid factorizations, every such system is solved using the preconditioned minimal residual (MINRES) method [135]. In order to accelerate the convergence of MINRES, we employ a block-diagonal preconditioner, which uses a diagonal approximation of $\nabla^{2} f(x)$. More specifically, at iteration $k$ of IP-PMM, we have the following coefficient matrix:
\[

M_{k}=\left[$$
\begin{array}{cc}
-H_{k} & A^{\top} \\
A & \delta_{k} I_{\bar{m}}
\end{array}
$$\right],
\]

where $H_{k}=\left(\nabla^{2} f\left(x_{k}\right)+\Theta_{k}^{-1}+\rho_{k} I_{\bar{n}}\right)$, and we precondition it using the matrix

$$
\widetilde{M}_{k}=\left[\begin{array}{cc}
\widetilde{H}_{k} & 0_{\bar{n}, \bar{m}}  \tag{6.29}\\
0_{\bar{m}, \bar{n}} & A \widetilde{H}_{k}^{-1} A^{\top}+\delta_{k} I_{\bar{m}}
\end{array}\right],
$$

where $\widetilde{H}_{k}$ is a diagonal approximation of $H_{k}$. In order to analyze the spectral properties of the preconditioned matrix, we follow the developments in Section 5.3.1. More specifically, we define $\widehat{H}_{k}:=\widetilde{H}_{k}^{-\frac{1}{2}} H_{k} \widetilde{H}_{k}^{\frac{1}{2}}$, and let:

$$
\alpha_{H}=\lambda_{\min }\left(\widehat{H}_{k}\right), \quad \beta_{H}=\lambda_{\max }\left(\widehat{H}_{k}\right), \quad \kappa_{H}=\frac{\beta_{H}}{\alpha_{H}} .
$$

Using this notation, we know that an arbitrary element of the numerical range of this matrix is represented as $\gamma_{H} \in W\left(\widehat{H}_{k}\right)=\left[\alpha_{H}, \beta_{H}\right]$. Furthermore, we observe that in the special case where $\widetilde{H}_{k}=\operatorname{Diag}\left(H_{k}\right)$, we have $\alpha_{H} \leq 1 \leq \beta_{H}$ since

$$
\frac{1}{n+2 l} \sum_{i=1}^{n+2 l} \lambda_{i}\left(\widehat{H}_{k}^{-1} H_{k, j}\right)=\frac{1}{n+2 l} \operatorname{Tr}\left(\hat{H}_{k}^{-1} H_{k}\right)=1 .
$$

Theorem 6.5.1. Let $k$ be an arbitrary IP-PMM iteration. Then, the eigenvalues of $\widetilde{M}_{k}^{-1} M_{k}$ lie in the union of the following intervals:

$$
I_{-}=\left[-\beta_{H}-1,-\alpha_{H}\right], \quad I_{+}=\left[\frac{1}{1+\beta_{H}}, 1\right] .
$$

Proof. The proof follows exactly the developments in Theorem 5.3.1.
In problem (6.28), $f(x)=D_{K L}(w)+c^{\top} u$ and hence

$$
\nabla f(x)=\left[\begin{array}{c}
\nabla D_{K L}(w) \\
c
\end{array}\right], \quad \nabla^{2} f(x)=\left[\begin{array}{cc}
\nabla^{2} D_{K L}(w) & 0_{n, 2 l} \\
0_{2 l, n} & 0_{2 l, 2 l}
\end{array}\right],
$$

where

$$
\nabla D_{K L}(w)=D^{\top}\left(e_{m}-\frac{g}{D w+a}\right), \quad \nabla^{2} D_{K L}(w)=D^{\top} U(w)^{2} D
$$

with $U(w)=\operatorname{Diag}\left(\frac{\sqrt{g}}{D w+a}\right)$. Here the ratios and the square root are assumed to be component-wise. Notice that $D$ might be dense; however, as previously
noted, its action can be computed via the FFT. Unfortunately, $D^{\top} U(w)^{2} D$ is not expected to be close to multilevel circulant. Even if it could be well-approximated by a multilevel circulant matrix, the scaling matrix of IP-PMM would destroy this structure. In other words, we use the structure of $D$ only when applying it to a vector. As a result, we only store the first column of $D$ and we use the FFT to apply this matrix to a vector. This allows us to compute the action of the Hessian easily.
Remark 8. The obvious choice would be to employ the approximation $\widetilde{H}_{k}=$ $\operatorname{Diag}\left(H_{k}\right)$, but the structure of the problem makes this choice rather expensive. A more efficient alternative is to use $\widetilde{H}_{k}=U\left(w_{k}\right)^{2}$, which is easier to compute and, as we will see in the following section, leads to good reconstruction results in practice.

### 6.5.2 Computational experience

To evaluate the performance of the IP-PMM on this class of problems, we consider a set of three $256 \times 256$ grayscale images, which are presented in Figure 6.2. For each of the three images we set up three restoration tests, where


Figure 6.2: The three $256 \times 256$ grayscale images of the image restoration tests.
the images are corrupted by Poisson noise and $D$ represents one of the following blurs: Gaussian blur (GB), motion blur (MB), and out-of-focus blur (OF) (see, e.g., [91] for further details).

We compare the proposed method with the state-of-the-art primal-dual algorithm with linesearch (PDAL) proposed in [113]. By following the example of [178, Algorithm 2], problem (6.27) is reformulated as

$$
\begin{equation*}
\min _{w} \max _{p, y} g^{\top} \ln (1+y)-y^{\top}(D w+a)-\lambda w^{\top} L^{\top} p+\chi_{\infty}(p)+\chi_{\mathcal{C}}(w) \tag{6.30}
\end{equation*}
$$

where $\chi_{\infty}$ is the characteristic function of the $\infty$-norm unit ball and $\chi_{\mathcal{C}}$ the characteristic function of the feasible set $\mathcal{C}$ of problem (6.27). It is worth noting that the PDAL algorithm for the solution of problem (6.30) requires at each step a projection on the feasible set $\mathcal{C}$, which is performed here by using the secant algorithm proposed by Dai and Fletcher in [49]. Concerning the parameters of PDAL, we use the same notation and tuning as in [113]. Following Section 6
of that paper, we set $\mu=0.7, \delta=0.99$ and $\beta=25$. The initial steplength is $\tau=\sqrt{1 / \omega}$, where $\omega$ is an estimate of $\left\|M^{\top} M\right\|$ and $M=\left[D^{\top} L^{\top}\right]^{\top}$ is the matrix linking the primal and dual variables. In the IP-PMM, we use the MINRES code by Michael Saunders and co-workers ${ }^{3}$ for which we set the relative tolerance $t o l=10^{-4}$ and the maximum number of iterations at each call equal to 20 . The regularization parameter $\lambda$ is determined by trial and error to minimize the root mean square error (RMSE) obtained by IP-PMM. We recall that, denoting the original image as $\bar{w} \in \mathbb{R}^{n}$, for any given approximate solution $w \in \mathbb{R}^{n}$ we have that

$$
\operatorname{RMSE}(w)=\frac{1}{\sqrt{n}}\|w-\bar{w}\|_{2} .
$$

For all the problems, the starting point is chosen to be the noisy and blurry image, i.e., $g$.


Figure 6.3: Comparison between IP-PMM and PDAL in terms of root mean square error (RMSE) vs execution time in the solution of the 9 image restoration problems. From top to bottom, the rows refer to the cameraman instances, the house instances and the peppers instances, respectively. From left to right, the columns refer to the GB, MB and OF, respectively.

For all 9 tests we run 20 iterations of the IP-PMM method and let PDAL run for the same amount of time. In Figure 6.3 we report a comparison between

[^6]the two algorithms in terms of elapsed time versus root mean square error in the solution of the 9 instances described above. As can be seen from the plots, the IP-PMM clearly outperforms PDAL on the instances with GB and OF (columns 1 and 3 , respectively, of Figure 6.3), while on the instances characterized by MB the two algorithms perform comparably.

Table 6.4: Comparison between IP-PMM and PDAL in terms of RMSE, PSNR and MSSIM computed at the solutions provided by the two algorithms.

|  | IP-PMM |  |  | PDAL |  |  |
| :--- | :---: | :---: | :---: | :---: | :---: | :---: | :---: |
| Problem | RMSE | PSNR | MSSIM | RMSE | PSNR | MSSIM |
| cameraman - GB | $4.85 \mathrm{e}-2$ | $2.63 \mathrm{e}+1$ | $8.33 \mathrm{e}-1$ | $5.02 \mathrm{e}-2$ | $2.60 \mathrm{e}+1$ | $8.22 \mathrm{e}-1$ |
| cameraman - MB | $5.52 \mathrm{e}-2$ | $2.52 \mathrm{e}+1$ | $8.11 \mathrm{e}-1$ | $5.59 \mathrm{e}-2$ | $2.51 \mathrm{e}+1$ | $7.77 \mathrm{e}-1$ |
| cameraman - OF | $5.14 \mathrm{e}-2$ | $2.58 \mathrm{e}+1$ | $7.98 \mathrm{e}-1$ | $5.26 \mathrm{e}-2$ | $2.56 \mathrm{e}+1$ | $7.62 \mathrm{e}-1$ |
| house - GB | $9.71 \mathrm{e}-2$ | $2.03 \mathrm{e}+1$ | $7.51 \mathrm{e}-1$ | $9.88 \mathrm{e}-2$ | $2.01 \mathrm{e}+1$ | $6.92 \mathrm{e}-1$ |
| house - MB | $2.70 \mathrm{e}-2$ | $3.14 \mathrm{e}+1$ | $8.67 \mathrm{e}-1$ | $2.77 \mathrm{e}-2$ | $3.11 \mathrm{e}+1$ | $8.43 \mathrm{e}-1$ |
| house - OF | $3.80 \mathrm{e}-2$ | $2.84 \mathrm{e}+1$ | $8.33 \mathrm{e}-1$ | $4.09 \mathrm{e}-2$ | $2.78 \mathrm{e}+1$ | $7.70 \mathrm{e}-1$ |
| peppers - GB | $1.23 \mathrm{e}-1$ | $1.82 \mathrm{e}+1$ | $7.46 \mathrm{e}-1$ | $1.25 \mathrm{e}-1$ | $1.81 \mathrm{e}+1$ | $6.57 \mathrm{e}-1$ |
| peppers - MB | $8.76 \mathrm{e}-2$ | $2.12 \mathrm{e}+1$ | $8.90 \mathrm{e}-1$ | $8.78 \mathrm{e}-2$ | $2.11 \mathrm{e}+1$ | $8.72 \mathrm{e}-1$ |
| peppers - OF | $9.47 \mathrm{e}-2$ | $2.05 \mathrm{e}+1$ | $8.01 \mathrm{e}-1$ | $9.70 \mathrm{e}-2$ | $2.03 \mathrm{e}+1$ | $6.60 \mathrm{e}-1$ |

To better analyze the difference between the solutions provided by the two algorithms, one can look at Table 6.4, where we report the value of three scores: RMSE, peak signal-to-noise ratio (PSNR), which is defined as

$$
\operatorname{PSNR}(w)=20 \log _{10} \frac{\max _{i} \bar{w}^{i}}{\operatorname{RMSE}(w)}
$$

and mean structural similarity (MSSIM), which is a structural similarity measure related to the perceived visual quality of the image (see [175] for a detailed definition). It is worth noting that for RMSE smaller values are better, while for PSNR and MSSIM, higher values indicate better noise removal and perceived similarity between the restored and original image, respectively. From the table it is clear that in all the considered cases IP-PMM is able to produce a better restored image than PDAL, having always a larger MSSIM, also when the RMSE and PSNR values are comparable.

For the sake of space, we now restrict the comparison to the cases where the two algorithms seem to have reached equivalent solutions in terms of RMSE, to understand the differences in the restored images. We focus on the three instances in which $D$ represents MB (second column of Figure 6.3). In Figure 6.4 we report the results for cameraman, house and peppers with MB. By looking at the images one can see that those reconstructed by IP-PMM appear to be smoother (look, for example, at the sky in cameraman and house), which somehow indicates that the IP-PMM is better than PDAL in enforcing the TV regularization. Observe that this "visual" difference is reflected by the higher values of MSSIM reported for IP-PMM in Table 6.4.


Figure 6.4: Results on cameraman, house and peppers with MB: noisy and blurry images (left), images restored by IP-PMM (center), images restored by PDAL (right).

### 6.6 Linear classification via regularized logistic regression

Finally, we deal with the problem of training a linear binary classifier. Let us consider a matrix $D \in \mathbb{R}^{n \times s}$ whose rows $\left(d^{i}\right)^{\top}$, with $i \in\{1, \ldots, n\}$, represent the training points, and a vector of labels $g \in\{-1,1\}^{n}$. In other words, we have a training set with $n$ binary-labeled samples and $s$ features. According to the logistic model, the conditional probability of having the label $g^{i}$ given the point $d_{i}$ has the form

$$
p_{l o g}(w)_{i}=P\left(g^{i} \mid d^{i}\right)=\frac{1}{1+e^{-g^{i} w^{\top} d^{i}}},
$$

where $w \in \mathbb{R}^{s}$ is the vector of weights determining the unbiased linear model under consideration. By following the maximum-likelihood approach, the weight vector
$w$ can be obtained by maximizing the log-likelihood function or, equivalently, by minimizing the logistic loss function, i.e., by solving

$$
\min _{w} \phi(w) \equiv \frac{1}{n} \sum_{i=1}^{n} \phi_{i}(w), \quad \phi_{i}(w)=\log \left(1+e^{-g^{i} w^{\top} d^{i}}\right)
$$

To cope with the inherent ill-posedness of the estimation process, a regularization term is usually added to the previous model. For large-scale instances, where the features tend to be redundant, an $\ell^{1}$-regularization term is usually introduced to enforce sparsity in the solution, thus embedding feature selection in the training process. This results in the well-studied $\ell^{1}$-regularized logistic regression model:

$$
\begin{equation*}
\min _{w} \phi(w)+\tau\|w\|_{1}, \tag{6.31}
\end{equation*}
$$

where $\tau>0$.
As done in the previous sections, we can replace the nonsmooth model (6.31) with an equivalent smooth convex programming problem, i.e.,

$$
\begin{gather*}
\min _{x} f(x) \equiv \phi(w)+c^{\top} u \\
\text { s.t. } A x=b  \tag{6.32}\\
\quad u \geq 0_{2 s}
\end{gather*}
$$

where, after introducing the additional constraint $u=w$, with $u=\left[\left(d^{+}\right)^{\top},\left(d^{-}\right)^{\top}\right]^{\top}$, and letting $\bar{m}=s, \bar{n}=3 s$, we set $x=\left[w^{\top}, u^{\top}\right]^{\top} \in \mathbb{R}^{\bar{n}}, c=\tau e_{2 s}, b=0_{\bar{m}}$, and $A \in \mathbb{R}^{\bar{m} \times \bar{n}}$ defined as $A=\left[\begin{array}{lll}I_{s} & -I_{s} & I_{s}\end{array}\right]$. The version of IP-PMM solving problem (6.32) is very similar to the one used to solve (6.28). The only difference here lies in the preconditioner. In particular, when solving problems of the form (6.32), we use the preconditioner defined in (6.29) (and subsequently analyzed in Theorem 6.5.1), but we set $\widetilde{H}_{k}=\operatorname{Diag}\left(H_{k}\right)$.

### 6.6.1 Computational experience

To illustrate the performance of the IP-PMM on this class of problems, we consider a set of three linear classification problems from the LIBSVM dataset for binary classification ${ }^{4}$. The names of the datasets, together with their number of features, training points and testing points are summarized in Table 6.5. For real-sim there is no predetermined separation of data between train and test, hence we apply a hold-out strategy keeping $30 \%$ of the data for testing.

To overcome the absence of the hyperplane bias in model (6.31), we add to the data matrices a further column with all ones, hence the resulting size of the problems is equal to $s+1$. For all the problems we set $\tau=\frac{1}{n}$, which is a standard choice in the literature.

To assess the effectiveness and efficiency of the proposed method we compare it with two state-of-the-art methods:

[^7]Table 6.5: Characteristics of the $\ell^{1}$-regularized logistic regression problems

| Problem | Features | Train pts | Test pts |
| :--- | ---: | ---: | ---: |
| gisette | 5000 | 6000 | 1000 |
| rcv1 | 47,236 | 20,242 | 677,399 |
| real-sim | 20,958 | 50,617 | 21,692 |

- an ADMM [32] ${ }^{5}$;
- a MATLAB implementation of the newGLMNET method [184] used in LIBSVM, developed by the authors of $[185]^{6}$.

As in the tests presented in Section 6.5.2, the solution of the augmented system in IP-PMM is performed by means of the MINRES implementation by Michael Saunders' team, with maximum number of iterations equal to 20 and tolerance tol $=10^{-4}$.

We compare the three algorithms in terms of objective function value and classification error versus execution time, on runs lasting 15 seconds. The plots are reported in Figure 6.5. The IP-PMM is comparable with newGLMNET on the gisette instance, characterized by a very dense ( $>99 \%$ ) training data matrix, and both IP-PMM and newGLMNET clearly outperform ADMM. On the rcv1 and real-sim instances the IP-PMM method sightly outperforms newGLMNET in terms of classification error, and it is noticeably better in terms of the objective function value.

### 6.7 Conclusions

We have presented specialized IP-PMMs for quadratic and general nonlinear convex optimization problems that model various sparse approximation instances. We have shown that by a proper choice of linear algebra solvers, which are a key issue in IPMs, we are able to efficiently solve the larger but smooth optimization problems coming from a standard reformulation of the original ones. This confirms the ability of IPMs to handle large sets of linear equality and inequality constraints. Computational experiments have been performed on diverse applications: multi-period portfolio selection, classification of fMRI data, restoration of blurry images corrupted by Poisson noise, and linear binary classification via regularized logistic regression. Comparisons with state-of-the-art first-order methods, which are widely used to tackle sparse approximation problems, have provided evidence that the presented IP-IPM approaches can offer a noticeable advantage over those methods, especially when dealing with not-so-well conditioned problems.

We also believe that the results presented in this chapter may provide a basis for an in-depth analysis of the application of IPMs to many sparse approximation

[^8]

Figure 6.5: Results on the three $\ell^{1}$-regularized logistic regression problems (objective function value and classification error versus execution time).
problems, and we plan to work in that direction in the future.

## Chapter 7

## Conclusions and Future Directions

### 7.1 Conclusions

In this thesis we studied the effects of algorithmic regularization in the context of interior point methods for convex optimization. We designed, analyzed and implemented several regularized interior point algorithms, suitable for a wide range of convex optimization problems. We analyzed the convergence properties of standard primal-dual regularized interior point methods, as well as the computational and numerical benefits of regularization.

In Chapter 2, we developed a non-diagonal regularization technique for interior point methods, with applications to linear and convex quadratic programming. We discussed how non-diagonal regularization can be employed as a means of sparsifying the linear systems that have to be solved at each IPM iteration. We performed a perturbation analysis to understand the effects of regularization in the eigenvalues of the associated linear systems, and proposed a tuning of the regularization parameters, so as to avoid perturbing the linear systems significantly. Then, we analyzed the spectral properties of the resulting regularized systems and demonstrated the effectiveness of the implemented approach by extensive experimentation.

In Chapter 3 we analyzed a primal-dual regularized IPM for linear and convex quadratic programming, which we called an interior point-proximal method of multipliers. We showed that under standard assumptions, the method converges to an $\epsilon$-optimal solution in a polynomial number of steps, and we provided a dynamical tuning for the regularization parameters, independent of the problem under consideration. As a byproduct of the theory, we designed an infeasibility detection mechanism. To demonstrate the practical effectiveness of the approach, we implemented it and tested it over a large set of real-life problems, showcasing the efficiency and most importantly the robustness of the approach.

In Chapter 4 we extended IP-PMM to the class of linear positive semi-definite programming problems, showing that polynomial complexity also holds in this case, even if one solves the associated linear systems inexactly. At the end of this chapter, we discussed the capability of the algorithm to detect pathological
semi-definite programming instances.
In Chapter 5 we discussed the efficient solution of large-scale linear systems arising from the application of a regularized interior point method to an arbitrary convex program. In particular, we derived several general-purpose preconditioning strategies usable within symmetric Krylov subspace solvers like the preconditioned conjugate gradient and the minimal residual method. A spectral analysis of each approach was given, and the effectiveness of the preconditioners was numerically verified after extensive experimentation on standard test sets, as well as on $L^{1}$-regularized optimal control problems.

Finally, in Chapter 6 we used the methodologies presented in the previous chapters to design specialized IP-PMM schemes suitable for the solution of various sparse approximation convex programs. In particular, we specialized IPPMM to problems arising from multi-period portfolio optimization, classification of functional magnetic resonance imaging data, restoration of images corrupted by Poisson noise, as well as classification via logistic regression. For each application, we tested the specialized IP-PMMs against state-of-the-art first-order approaches which are traditionally employed in the literature, and we were able to show that IP-PMM can be equally (or more) efficient, and at the same time more robust and reliable.

### 7.2 Future directions

There are several open questions as well as possible future research directions in the field of regularized interior point methods for convex programming.

Firstly, we know that the algorithm presented in Chapter 2 is globally convergent under certain assumptions. A possible extension would be to study a variation of this algorithm, based on the developments in Chapter 3, and potentially show polynomial convergence. Furthermore, it would be interesting to extend this non-diagonal regularization strategy to the case of nonlinear convex programming, and potentially analyze the local convergence of the resulting algorithm.

On the other hand, the algorithms developed in Chapters 3, 4, can be easily extended to other classes of conic optimization problems, and we conjecture that polynomial complexity should hold for any symmetric cone. Specifically, a case of interest would be to extend the analysis of Chapter 4 to more general symmetrization strategies, such as those appearing within the Monteiro-Zhang family (see [167] for a detailed description of symmetrization strategies for semi-definite programming). From a more practical perspective, a potential research direction would be to derive efficient implementations of IP-PMM for the second-order cone as well as the semi-definite cone, utilizing the preconditioning approaches developed in Chapter 5. Furthermore, one could even design an IP-PMM for other non-symmetric cones, such as the exponential or the power cone. Additionally, it would be interesting to analyze the local convergence of IP-PMM for the class of general nonlinear convex programming problems. Finally, an open question is to show that polynomial complexity of the algorithms presented in Chapters 3, 4 , holds even in the rank-deficient case. We were not able to bypass the full-rank
assumption, and thus we are not aware if this is possible.
An important extension to the preconditioning strategies in Chapter 5 would be to derive matrix-free preconditioners that could be computed by means of matrix-vector products, and thus would not need to employ a factorization scheme. Such preconditioners would allow regularized IPMs to tackle huge-scale instances, or problems which are too large to be stored in the available memory.

Finally, based on the success of IP-PMM in solving sparse approximation instances, one could perform an in-depth analysis of the application of regularized IPMs to a wide range of sparse approximation problems. This could shed light on the effectiveness of such methods for this class of problems, and could pave the way for the creation of robust general implementations for their efficient solution.

## Bibliography

[1] A. Altman and J. Gondzio, Regularized symmetric indefinite systems in interior point methods for linear and quadratic optimization, Optimization Methods and Software, 11 (1999), pp. 275-302, https://doi.org/10. 1080/10556789908805754.
[2] E. D. Andersen, J. Gondzio, C. Mészáros, and X. Xu, Implementation of interior point methods for large scale linear programming, in Interior Point Methods in Mathematical Programming, T. Terlaky, ed., Kluwer Academic Publishers, 1996, pp. 189-252, https://doi.org/10. 1007/978-1-4613-3449-1_6.
[3] A. Argyriou, L. Baldassarre, C. A. Micchelli, and M. Pontil, On sparsity inducing regularization methods for machine learning, in Empirical Inference: Festschrift in Honor of Vladimir N. Vapnik, B. Schölkopf, Z. Luo, and V. Vovk, eds., Berlin, Heidelberg, 2013, Springer, pp. 205-216, https://doi.org/10.1007/978-3-642-41136-6_18.
[4] P. Armand and J. Benoist, Uniform boundedness of the inverse of a Jacobian matrix arising in regularized interior-point methods, Mathematical Programming, 137 (2013), pp. 587-592, https://doi.org/10.1007/ s10107-011-0498-3.
[5] P. Armand and J. Benoist, A mixed logarithmic barrier-augmented Lagrangian method for nonlinear optimization, Journal of Optimization Theory and Applications, 173 (2017), pp. 523-547, https://doi. org/10. 1007/s10957-017-1071-x.
[6] P. Armand, J. Benoist, and D. Orban, From global to local convergence of interior methods for nonlinear optimization, Optimization Methods and Software, 28 (2013), pp. 1051-1080, https://doi.org/10.1080/ 10556788.2012.668905.
[7] P. Armand and R. Omheni, A globally and quadratically convergent primal-dual augmented Lagrangian algorithm for equality constrained optimization, Optimization Methods and Software, 32 (2017), pp. 1-21, https://doi.org/10.1080/10556788.2015.1025401.
[8] P. Armand and N. N. Tran, Rapid infeasibility detection in a mixed logarithmic barrier-augmented Lagrangian method for nonlinear optimiza-
tion, Optimization Methods and Software, 34 (2017), https://doi.org/ 10.1080/10556788.2018.1528250.
[9] P. Armand and N. N. Tran, An augmented Lagrangian method for equality constrained optimization with rapid infeasibility detection capabilities, Journal of Optimization Theory and Applications, 181 (2019), pp. 197215, https://doi.org/10.1007/s10957-018-1401-7.
[10] S. Arreckx and D. Orban, A regularized factorization-free method for equality-constrained optimization, SIAM Journal on Optimization, 28 (2018), pp. 1613-1639, https://doi.org/doi.org/10.1137/16M1088570.
[11] V. Balakrishnan and F. Wang, Handbook of Semidefinite Programming, International Series in Operational Research \& Managment Science, Vol. 27. Springer, Boston, MA., 2000.
[12] L. Baldassarre, J. Mourão-Miranda, and M. Pontil, Structured sparsity models for brain decoding from fMRI data, in 2012 Second International Workshop on Pattern Recognition in NeuroImaging, July 2012, pp. 5-8, https://doi.org/10.1109/PRNI.2012.31.
[13] L. Baldassarre, M. Pontil, and J. Mourão-Miranda, Sparsity is better with stability: Combining accuracy and stability for model selection in brain decoding, Frontiers in Neuroscience, 11 (2017), https://doi.org/ 10.3389/fnins. 2017.00062 .
[14] R. C. Barnard, H. Bilheux, T. Toops, E. Nafziger, C. Finney, D. Splitter, and R. Archibald, Total variation-based neutron computed tomography, Review of Scientific Instruments, 89 (2018), p. 053704, https://doi.org/10.1063/1.5037341.
[15] A. Beck and M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM Journal on Imaging Sciences, 2 (2009), pp. 183-202, https://doi.org/10.1137/080716542.
[16] S. Bellavia, Inexact interior-point method, Journal of Optimization Theory and Applications, 96 (1998), pp. 109-121, https://doi.org/10.1023/ A: 1022663100715 .
[17] S. Bellavia, J. Gondzio, and B. Morini, A matrix-free preconditioner for sparse symmetric positive definite systems and least-squares problems, SIAM Journal on Scientific Computing, 35 (2013), https://doi.org/10. 1137/110840819.
[18] S. Bellavia, J. Gondzio, and M. Porcelli, An inexact dual logarithmic barrier method for solving sparse semidefinite programs, Mathematical Programming, 178 (2019), pp. 109-143, https://doi.org/10.1007/ s10107-018-1281-5.
[19] S. Bellavia, J. Gondzio, and M. Porcelli, A relaxed interior point method for low-rank semidefinite programming problems, Journal of Scientific Computing, 89 (2021), p. 46, https://doi.org/10.1007/ s10915-021-01654-1.
[20] A. Ben-Tal and A. Nemirovski, Robust convex optimization, Mathematics of Operations Research, 23 (1998), pp. 769-805, https://doi.org/ 10.1287/moor.23.4.769.
[21] M. Benzi, G. H. Golub, and J. Liesen, Numerical solution of saddle point systems, Acta Numerica, 14 (2005), pp. 1-137, https://doi.org/ 10.1017/S0962492904000212.
[22] L. Bergamaschi, R. Bru, and A. Martínez, A low-rank update of preconditioners for the inexact Newton method with SPD Jacobian, Mathematical and Computer Modelling, 54 (2011), pp. 1863-1873, https: //doi.org/10.1016/j.mcm.2010.11.064.
[23] L. Bergamaschi, J. Gondzio, A. Martínez, J. W. Pearson, and S. Pougkakiotis, A new preconditioning approach for an interior pointproximal method of multipliers for linear and convex quadratic programming, Numerical Linear Algebra with Applications, 28 (2020), p. e2361, https://doi.org/10.1002/nla. 2361.
[24] L. Bergamaschi, J. Gondzio, M. Venturin, and G. Zilli, Inexact constraint preconditioners for linear systems arising in interior point methods, Computational Optimization and Applications, 36 (2007), pp. 137147, https://doi.org/10.1007/s10589-006-9001-0. see also Erratum, COAP 49 (2011) 401-406.
[25] L. Bergamaschi, J. Gondzio, and G. Zilli, Preconditioning indefinite systems in interior point methods for optimization, Computational Optimization and Applications, 28 (2004), pp. 149-171, https://doi. org/10. 1023/B:COAP.0000026882.34332.1b.
[26] M. Bertero, P. Boccacci, G. Desiderà, and G. Vicidomini, Image deblurring with Poisson data: from cells to galaxies, Inverse Problems, 25 (2009), p. 123006, https://doi.org/10.1088/0266-5611/25/12/123006.
[27] D. Bertsekas, Constrained Optimization and Lagrange Multiplier Methods, Athena Scientific, 1996.
[28] D. Bertsekas, A. Nedic, and E. Ozdaglar, Convex Analysis and Optimization, Athena Scientific, 2003.
[29] D. Bertsekas and J. N. Tsitsiklis, Parallel and Distributed Computation: Numerical Methods, Athena Scientific, 1997.
[30] S. Bocanegra, F. Campos, and A. Oliveira, Using a hybrid preconditioner for solving large-scale linear systems arising from interior point methods, Computational Optimization and Applications, 36 (2007), pp. 149-164, https://doi.org/10.1007/s10589-006-9009-5.
[31] P. T. Boggs and J. W. Tolle, Sequential quadratic programming for large-scale nonlinear optimization, Journal of Computational and Applied Mathematics, 124 (2000), pp. 123-137, https://doi.org/10.1016/ S0377-0427(00)00429-5.
[32] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Foundations and Trends in Machine Learning, 3 (2011), pp. 1-122, https://doi.org/10.1561/2200000016.
[33] A. N. Brooks and T. J. R. Hughes, Streamline upwind/PetrovGalerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier-Stokes equations, Computer Methods in Applied Mechanics and Engineering, 32 (1982), pp. 199-259, https: //doi.org/10.1016/0045-7825(82)90071-8.
[34] S. Cafieri, M. D'Apuzzo, V. De Simone, D. di Serafino, and G. Toraldo, Convergence analysis of an inexact potential reduction method for convex quadratic programming, Journal of Optimization Theory and Applications, 135 (2007), pp. 355-366, https://doi.org/10.1007/ s10957-007-9264-3.
[35] E. J. Candés, J. Romberg, and T. Tao, Stable signal recovery from incomplete and inaccurate measurements, Communications in Pure Applied Mathematics, 59 (2006), pp. 1207-1223, https://doi.org/doi.org/10. 1002/cpa. 20124.
[36] Y. Cao, C. D. Laird, and V. M. Zavala, Clustering-based preconditioning for stochastic programs, Computational Optimization and Applications, 64 (2016), pp. 379-406, https://doi.org/10.1007/ s10589-015-9813-x.
[37] J. Castro, A specialized interior-point algorithm for multicommodity network flows, SIAM Journal on Optimization, 10 (2000), pp. 852-877, https://doi.org/10.1137/S1052623498341879.
[38] J. Castro and J. Cuesta, Quadratic regularizations in an interior-point method for primal block-angular problems, Mathematical Programming, 130 (2011), pp. 415-445, https://doi.org/10.1007/s10107-010-0341-2.
[39] Y. Censor and A. Zenios, Proximal minimization algorithm with Dfunctions, Journal of Optimization Theory and Applications, 73 (1992), pp. 451-464, https://doi.org/10.1007/BF00940051.
[40] J.-S. Chai and K.-C. Toh, Preconditioning and iterative solution of symmetric indefinite linear systems arising from interior point methods for linear programming, Computational Optimization and Applications, 36 (2007), pp. 221-247, https://doi.org/10.1007/s10589-006-9006-8.
[41] A. Chambolle, An algorithm for total variation minimization and applications, Journal of Mathematical Imaging and Vision, 20 (2004), pp. 89-97, https://doi.org/10.1023/B: JMIV.0000011325.36760.1e.
[42] S. S. Chen, D. L. Donoho, and M. A. Saunders, Atomic decomposition by basis pursuit, SIAM Review, 43 (2001), pp. 129-159, https: //doi.org/10.1137/S003614450037906X.
[43] Z.-P. Chen, G. Li, and J.-E. Guo, Optimal investment policy in the time consistent mean-variance formulation, Insurance: Mathematics \& Economics, 52 (2013), pp. 145-156, https://doi.org/10.1016/j. insmatheco.2012.11.007.
[44] G. H. Cheng, Q. Tan, and Z. D. Wang, A note on eigenvalues of perturbed 2x2 block Hermitian matrices, Linear and Multilinear Algebra, 63 (2014), pp. 820-825, https://doi.org/10.1080/03081087.2014.903252.
[45] S. Corsaro and V. De Simone, Adaptive $l_{1}$-regularization for short-selling control in portfolio selection, Computational Optimization and Applications, 72 (2019), pp. 457-478, https://doi.org/10.1007/ s10589-018-0049-4.
[46] S. Corsaro, V. De Simone, and Z. Marino, Fused lasso approach in portfolio selection, Annals of Operations Research, 299 (2021), pp. 47-59, https://doi.org/10.1007/s10479-019-03289-w.
[47] S. Corsaro, V. De Simone, and Z. Marino, Split Bregman iteration for multi-period mean variance portfolio optimization, Applied Mathematics and Computation, 392 (2021), pp. 125715, 10, https://doi.org/10.1016/ j.amc. 2020.125715.
[48] S. Corsaro, V. De Simone, Z. Marino, and F. Perla, $l_{1}-$ regularization for multi-period portfolio selection, Annals of Operations Research, 294 (2020), pp. 75-86, https://doi.org/10.1007/ s10479-019-03308-w.
[49] Y.-H. Dai and R. Fletcher, New algorithms for singly linearly constrained quadratic programs subject to lower and upper bounds, Mathematical Programming, 106 (2006), pp. 403-421, https://doi.org/10.1007/ s10107-005-0595-2.
[50] M. D'Apuzzo, V. De Simone, and D. di Serafino, On mutual impact of numerical linear algebra and large-scale optimization with focus on interior point methods, Computational Optimization and Applications, 45 (2010), pp. 283-310, https://doi.org/10.1007/s10589-008-9226-1.
[51] M. D'Apuzzo, V. De Simone, and D. di Serafino, On mutual impact of numerical linear algebra and large-scale optimization with focus on interior point methods, Computational Optimization and Applications, 45 (2010), pp. 283-310, https://doi.org/10.1007/s10589-008-9226-1.
[52] V. De Simone, D. di Serafino, J. Gondzio, S. Pougkakiotis, and M. Viola, Sparse approximations with interior point methods, arXiv:2102.13608, (2021).
[53] V. De Simone, D. di Serafino, and M. Viola, A subspace-accelerated split Bregman method for sparse data recovery with joint $\ell_{1}$-type regularizers, Electronic Transactions on Numerical Analysis, 53 (2020), pp. 406-425, https://doi.org/10.1553/etna_vol53s406.
[54] A. Dehghani, J. L. Goffin, and D. Orban, A primal-dual regularized interior-point method for semidefinite programming, Optimization Methods and Software, 32 (2017), pp. 193-219, https://doi.org/10.1080/ 10556788.2016.1235708.
[55] D. di Serafino, G. Landi, and M. Viola, ACQUIRE: An inexact iteratively reweighted norm approach for TV-based Poisson image restoration, Applied Mathematics and Computation, 364 (2020), pp. 124678, 23, https://doi.org/10.1016/j.amc.2019.124678.
[56] D. di Serafino and D. Orban, Constraint-preconditioned Krylov solvers for regularized saddle-point systems, SIAM Journal on Scientific Computing, 43 (2021), pp. A1001-A1026, https://doi.org/10.1137/ 19M1291753.
[57] E. D. Dohmatob, A. Gramfort, B. Thirion, and G. Varoquaux, Benchmarking solvers for TV-l1 least-squares and logistic regression in brain imaging, in 2014 International Workshop on Pattern Recognition in Neuroimaging, June 2014, pp. 1-4, https://doi.org/10.1109/PRNI. 2014.6858516.
[58] D. E. Dolan and J. J. Moré, Benchmarking optimization software with performance profiles, Mathematical Programming, 91 (2002), pp. 201-213, https://doi.org/10.1007/s101070100263.
[59] H. S. Dollar, Constraint-style preconditioners for regularized saddle point problems, SIAM Journal on Matrix Analysis and Applications, 29 (2007), pp. 672-684, https://doi.org/10.1137/050626168.
[60] H. S. Dollar, N. I. M. Gould, W. H. A. Schilders, and A. J. WaTHEN, Using constraint preconditioners with regularized saddle-point problems, Computational Optimization and Applications, 36 (2007), pp. 249270, https://doi.org/10.1007/s10589-006-9004-x.
[61] M. Dubois, F. Hadj-Selem, T. Löfstedt, M. Perrot, C. Fischer, V. Frouin, and E. Duchesnay, Predictive support recovery with

TV-Elastic Net penalty and logistic regression: An application to structural MRI, in 2014 International Workshop on Pattern Recognition in Neuroimaging, June 2014, pp. 1-4, https://doi.org/10.1109/PRNI. 2014. 6858517.
[62] I. S. Duff, A. M. Erisman, and J. K. Reid, Direct Methods for Saprse Matrices, 2nd ed., Oxford University Press, Oxford, UK, 2017, https:// doi.org/10.1093/acprof:oso/9780198508380.001.0001.
[63] C. Durazzi and V. Ruggiero, Indefinitely preconditioned conjugate gradient method for large sparse equality and inequality constrained quadratic problems, Numerical Linear Algebra with Applications, 10 (2003), pp. 673688, https://doi.org/10.1002/nla. 308.
[64] J. Eckstein, Nonlinear proximal point algorithms using Bregman functions, with applications to convex programming, Mathematics of Operations Research, 18 (1993), pp. 202-226.
[65] H. C. Elman, A. Ramage, and D. J. Silvester, Algorithm 866: IFISS, a Matlab toolbox for modelling incompressible flow, ACM Transactions on Mathematical Software, 33 (2007), p. 14, https://doi.org/ 10.1145/1236463.1236469.
[66] H. C. Elman, A. Ramage, and D. J. Silvester, IFISS: A computational laboratory for investigating incompressible flow problems, SIAM Review, 52 (2014), pp. 261-273, https://doi.org/10.1137/120891393.
[67] H. C. Elman, D. J. Silvester, and A. J. Wathen, Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics, Numerical Mathematics and Scientific Computation, Oxford University Press, 2nd ed., 2014.
[68] D. C.-L. Fong and M. A. Saunders, LSMR: An iterative algorithm for sparse least-squares problems, SIAM Journal on Scientific Computing, 33 (2011), pp. 2950-2971, https://doi.org/10.1137/10079687X.
[69] K. Fountoulakis and J. Gondzio, A second-order method for strongly convex $\ell_{1}$-regularization problems, Mathematical Programming, 156 (2016), pp. 189-219, https://doi.org/10.1007/s10107-015-0875-4.
[70] K. Fountoulakis, J. Gondzio, and P. Zhlobich, Matrix-free interior point method for compressed sensing problems, Mathematical Programming Computation, 6 (2014), pp. 1-31, https://doi.org/10.1007/ s12532-013-0063-6.
[71] M. Friedlander and P. Tseng, Exact regularization of convex programs, SIAM Journal on Optimization, 18 (2007), pp. 1326-1350, https: //doi.org/10.1137/060675320.
[72] M. P. Friedlander and D. Orban, A primal-dual regularized interiorpoint method for convex quadratic programs, Mathematical Programming Computation, 4 (2012), pp. 71-107, https://doi.org/10.1007/ s12532-012-0035-2.
[73] C. F. Gauss, Letter to Gerling, In Werke, 9 (1903), pp. 278-281.
[74] P. E. Gill, W. Murray, D. B. Ponceleón, and M. A. SaunDERS, Preconditioners for indefinite systems arising in optimization, SIAM Journal on Matrix Analysis and Applications, 13 (1992), pp. 292-311, https://doi.org/10.1137/0613022.
[75] O. GÜLER, New proximal point algorithms for convex minimization, SIAM Journal on Optimization, 2 (1992), pp. 649-664, https://doi.org/10. 1137/0802032.
[76] GLPK. https://www.gnu.org/software/glpk/, 2018.
[77] J. GonDzio, Multiple centrality corrections in a primal-dual method for linear programming, Computational Optimization and Applications, 6 (1996), pp. 137-156, https://doi.org/10.1007/BF00249643.
[78] J. Gondzio, Interior point methods 25 years later, European Journal of Operational Research, 218 (2012), pp. 587-601, https://doi.org/10. 1016/j.ejor.2011.09.017.
[79] J. Gondzio, Matrix-free interior point method, Computational Optimization and Applications, 51 (2012), pp. 457-480, https://doi. org/10.1007/ s10589-010-9361-3.
[80] J. Gondzio, Convergence analysis of an inexact feasible interior point method for convex quadratic programming, SIAM Journal on Optimization, 23 (2013), pp. 1510-1527, https://doi.org/10.1137/120886017.
[81] J. Gondzio and M. Makowski, Solving a class of LP problems with a primal-dual logarithmic barrier method, European Journal of Operational Research, 80 (1995), pp. 184-192, https://doi.org/https://doi.org/ 10.1016/0377-2217 (93)E0323-P.
[82] J. Gondzio, S. Pougkakiotis, and J. W. Pearson, General-purpose preconditioning for regularized interior point methods, 2021, https:// arxiv.org/abs/arXiv:2107.06822.
[83] J. Gondzio and G. Toraldo (Eds.), Linear algebra issues arising in interior point methods, Special issue of Computational Optimization and Applications, 36 (2007), pp. 137-341.
[84] A. Gramfort, B. Thirion, and G. Varoquaux, Identifying predictive regions from fMRI with TV-L1 prior, in 2013 International Workshop on Pattern Recognition in Neuroimaging, June 2013, pp. 17-20, https://doi. org/10.1109/PRNI.2013.14.
[85] A. Greenbaum, Iterative Methods for Solving Linear Systems, Frontiers in Applied Mathematics, Society for Industrial and Applied Mathematics, 1997, https://doi.org/10.1137/1.9781611970937.
[86] A. Greenbaum, V. Pták, and Z. Strakoš, A nonincreasing convergence curve is possible for GMRES, SIAM Journal on Matrix Analysis and Applications, 17 (1996), pp. 465-469, https://doi.org/10.1137/ S0895479894275030.
[87] L. Grosenick, B. Klingenberg, K. Katovich, B. Knutson, and J. E. Taylor, Interpretable whole-brain prediction analysis with GraphNet, NeuroImage, 72 (2013), pp. 304-321, https://doi.org/10.1016/j. neuroimage.2012.12.062.
[88] M. Gu, Primal-dual interior-point methods for semidefinite progamming in finite precision, SIAM Journal on Optimization, 10 (2000), pp. 462-502, https://doi.org/10.1137/S105262349731950X.
[89] O. GüLER and Y. Ye, Convergence behaviour of interior point algorithms, Mathematical Programming, 60 (1993), pp. 215-228, https://doi.org/ 10.1007/BF01580610.
[90] G. Haeser, O. Hinder, and Y. Ye, On the behaviour of Lagrange multipliers in convex and non-convex ifeasible interior point methods, Mathematical Programming, 186 (2021), pp. 257-288, https://doi.org/10. 1007/s10107-019-01454-4.
[91] P. C. Hansen, J. G. Nagy, and D. P. O'Leary, Deblurring Images, Society for Industrial and Applied Mathematics, 2006, https://doi.org/ 10.1137/1.9780898718874.
[92] M. R. Hestenes, Multiplier and gradient methods, Journal of Optimization Theory and Applications, 4 (1969), pp. 303-320, https://doi.org/ 10.1007/BF00927673.
[93] M. R. Hestenes and E. Stiefel, Method of conjugate gradients for solving linear systems, Journal of Research of the National Bureau of Standards, 49 (1952), pp. 409-436.
[94] I. C. F. IPSEn, A note on preconditioning non-symmetric matrices, SIAM Journal on Scientific Computing, 23 (2001), pp. 1050-1051, https://doi. org/10.1137/S1064827500377435.
[95] A. N. Iusem, Some properties of generalized proximal point methods for quadratic and linear programming, Journal of Optimization Theory and Applications, 85 (1995), pp. 593-612, https://doi.org/10.1007/ BF02193058.
[96] C. G. J. Jacobi, Ueber eine neue auflösungsart der bei der methode der Kleinsten quadrate vorkommenden lineären gleichungen, Astronomische Nachrichten, 22 (1845), pp. 297-306.
[97] X. Jiang and L. Vandenberghe, Bregman primal-dual first-order method and application to sparse semidefinite programming, http://www. optimization-online.org/DB_HTML/2020/03/7702.html, (2020).
[98] B. Jie, C.-Y. Wee, D. Shen, and D. Zhang, Hyper-connectivity of functional networks for brain disease diagnosis, Medical Image Analysis, 32 (2016), pp. 84-100, https://doi.org/10.1016/j.media.2016.03.003.
[99] Y. Kamitani and F. Tong, Decoding the visual and subjective contents of the human brain, Nature Neuroscience, 8 (2005), pp. 679-685, https: //doi.org/10.1038/nn1444.
[100] C. Keller, N. I. Gould, and A. J. Wathen, Constraint preconditioning for symmetric indefinite linear systems, SIAM Journal on Matrix Analysis and Applications, 21 (2000), pp. 1300-1317, https://doi.org/ 10.1137/S0895479899351805.
[101] D. Klatte and B. Kummer, Nonsmooth Equations in Optimization, Regularity, Calculus, Methods and Applications, vol. 60 of Nonconvex Optimization and its Applications, Kluwer Academic Publishers, Dordrecht, Springer, Boston, MA, 2002, https://doi.org/10.1007/b130810.
[102] M. Kojima, N. Megiddo, and S. Mizuno, A primal-dual infeasible-interior-point algorithm for linear programming, Mathematical Programming, 61 (1993), pp. 263-280, https://doi.org/10.1007/BF01582151.
[103] M. Kojima and S. Shindo, Extension of Newton and quasi-Newton methods to systems of PC ${ }^{1}$ equations, Journal of the Operational Research Society of Japan, 29 (1986), pp. 352-375, https://doi.org/10.15807/jorsj. 29.352.
[104] G. Lan and R. D. C. Monteiro, Iteration-complexity of first-order augmented Lagrangian methods for convex programming, Mathematical Programming, 155 (2016), pp. 511-547, https://doi.org/10.1007/ s10107-015-0861-x.
[105] J. Lavaei and S. H. Low, Zero duality gap in optimal power flow problem, IEEE Transactions on Power Systems, 27 (2012), pp. 92-107, https:// doi.org/10.1109/TPWRS. 2011.2160974.
[106] J. D. Lee, Y. Sun, and M. A. Saunders, Proximal Newton-type methods for minimizing composite functions, SIAM Journal on Optimization, 24 (2014), pp. 1420-1443, https://doi.org/10.1137/130921428.
[107] D. Li and W. Ng, Optimal dynamic portfolio selection: Multiperiod mean-variance formulation, Mathematical Finance, 10 (2000), pp. 387-406, https://doi.org/10.1111/1467-9965.00100.
[108] X. Li, D. Sun, and K.-C. Toh, A highly efficient semismooth Newton augmented Lagrangian method for solving lasso problems, SIAM Journal on Optimization, 28 (2017), pp. 433-458, https://doi.org/10.1137/ 16M1097572.
[109] Y. Li, C. Sun, P. Li, Y. Zhao, G. K. Mensah, Y. Xu, H. Guo, and J. Chen, Hypernetwork construction and feature fusion analysis based on sparse group lasso method on fMRI dataset, Frontiers in Neuroscience, 14 (2020), https://doi.org/10.3389/fnins. 2020.00060 .
[110] J. Liesen and Z. Strakos̆, Krylov Subspace Methods: Principles and Analysis, Numerical Mathematics and Sicentific Computation, Oxford University Press, 2012, https://doi.org/10.1093/acprof:oso/ 9780199655410.001.0001.
[111] J. Liesen and P. Tichý, Convergence analysis of Krylov subspace methods, GAMM-Mitteilungen, 27 (2005), pp. 153-173, https://doi.org/10. 1002/gamm. 201490008.
[112] M. Liu and G. Pataki, Exact duals and short certificates of infeasibility and weak infeasibility in conic linear programming, Mathematical Programming, 167 (2018), pp. 435-480, https://doi.org/10.1007/ s10107-017-1136-5.
[113] Y. Malitsky and T. Pock, A first-order primal-dual algorithm with linesearch, SIAM Journal on Optimization, 28 (2018), pp. 411-432, https: //doi.org/10.1137/16M1092015.
[114] H. M. Markowitz, Portfolio Selection: Efficient Diversification of Investments, Cowles Foundation for Research in Economics at Yale University, Monograph 16, John Wiley \& Sons, Inc., New York; Chapman \& Hall, Ltd., London, 1959.
[115] I. Maros and C. Mészáros, The role of the augmented system in interior point methods, European Journal of Operational Research, 107 (1998), pp. 720-736, https://doi.org/10.1016/S0377-2217(97)00074-X.
[116] I. Maros and C. Mészáros, A Repository of convex quadratic programming problems, Optimization Methods and Software, 11 \& 12 (1999), pp. 671-681, https://doi.org/10.1080/10556789908805768.
[117] A. Martínez, Tuned preconditioners for the eigensolution of large SPD matrices arising in engineering problems, Numerical Linear Algebra with Applications, 23 (2016), pp. 427-443, https://doi.org/10.1002/nla. 2032.
[118] N. Megiddo, Pathways to the optimal set in linear programming, in Progress in Mathematical Programming, N. Megiddo, ed., Springer-Verlag, 1989, https://doi.org/10.1007/978-1-4613-9617-8_8.
[119] S. Mehrotra, On the implementation of a primal-dual interior-point method, SIAM Journal on Optimization, 2 (1992), pp. 575-601, https: //doi.org/10.1137/0802028.
[120] V. Michel, A. Gramfort, G. Varoquaux, E. Eger, and B. Thirion, Total variation regularization for fMRI-based prediction of behavior, IEEE Transactions on Medical Imaging, 30 (2011), pp. 1328-1340, https://doi.org/10.1109/TMI.2011.2113378.
[121] S. Mizuno and F. Jarre, Global and polynomial-time convergence of an infeasible-interior-point algorithm using inexact computation, Mathematical Programming, 84 (1999), pp. 105-122, https://doi.org/10.1007/ s10107980020a.
[122] S. Mizuno, M. J. Todd, and Y. Ye, On adaptive-step primal-dual interior-point algorithms for linear programming, Mathematics of Operational Research, 18 (1993), pp. 964-981, https://doi.org/10.1287/moor. 18.4.964.
[123] MOSEK, The MOSEK optimization software, 2020.
[124] A. M. Mota, N. Oliveira, P. Almeida, and N. Matela, 3D total variation minimization filter for breast tomosynthesis imaging, in Breast Imaging, A. Tingberg, K. Lång, and P. Timberg, eds., Cham, 2016, Springer, pp. 501-509, https://doi.org/10.1007/978-3-319-41546-8_ 63.
[125] J. Mourão-Miranda, E. Reynaud, F. McGlone, G. Calvert, and M. Brammer, The impact of temporal compression and space selection on SVM analysis of single-subject and multi-subject fMRI data, NeuroImage, 33 (2006), pp. 1055-1065, https://doi.org/10.1016/j. neuroimage.2006.08.016.
[126] M. F. Murphy, G. H. Golub, and A. J. Wathen, A note on preconditioning for indefinite linear systems, SIAM Journal on Scientific Computing, 21 (2000), pp. 1969-1972, https://doi.org/10.1137/ S1064827599355153.
[127] Y. Nakatsukasa, Eigenvalue perturbation bounds for Hermitian block tridiagonal matrices, Applied Numerical Mathematics, 62 (2012), pp. 6778, https://doi.org/10.1016/j.apnum.2011.09.010.
[128] Y. Nesterov and A. S. NemirovskiI, Interior Point Polynomial Methods in Convex Programming: Theory and Algorithms, SIAM Publications, SIAM, Philadelphia, USA, 1994, https://doi.org/10.1137/1. 9781611970791.
[129] Y. Nesterov, M. J. Todd, and Y. Ye, Infeasible-start primal-dual methods and infeasibility detectors for nonlinear programming problems, Mathematical Programming, 82 (1999), pp. 227-267, https://doi.org/ 10.1007/s10107980009a.
[130] NetLib. http://netlib.org/lp, 2011.
[131] J. Nocedal and S. J. Wright, Numerical Optimization, Springer Series in Operations Research and Financial Engineering, Springer, New York, NY, 2006, https://doi.org/10.1007/978-0-387-40065-5.
[132] Y. Notay, A new analysis of block preconditioners for saddle point systems, SIAM Journal on Matrix Analysis and Applications, 35 (2014), pp. 143-173, https://doi.org/10.1137/130911962.
[133] A. R. L. Oliveira and D. C. Sorensen, A new class of preconditioners for large-scale linear systems from interior point methods for linear programming, Linear Algebra and its Applications, 394 (2005), pp. 1-24, https://doi.org/10.1016/j.laa.2004.08.019.
[134] D. Orban, Limited-memory $L D L^{T}$ factorization of symmetric quasidefinite matrices with application to constrained optimization, Numerical Algorithms, 70 (2015), pp. 9-41, https://doi.org/10.1007/ s11075-014-9933-x.
[135] C. C. Paige and M. A. Saunders, Solution of sparse indefinite systems of linear equations, SIAM Journal on Numerical Analysis, 12 (1975), pp. 617-629, https://doi.org/10.1137/0712047.
[136] C. C. Paige and M. A. Saunders, LSQR: An algorithm for sparse linear equations and sparse least squares, ACM Transactions on Mathematical Software, 8 (1982), pp. 43-71, https://doi.org/10.1145/355984. 355989.
[137] N. Parikh and S. Boyd, Proximal algorithms, Foundations and Trends in Optimization, 1 (2014), pp. 127-239, https://doi.org/10.1561/ 2400000003.
[138] J. W. Pearson and J. Gondzio, Fast interior point solution of quadratic programming problems arising from PDE-constrained optimization, Numerische Mathematik, 137 (2017), pp. 959-999, https://doi.org/10. 1007/s00211-017-0892-8.
[139] J. W. Pearson and J. Pestana, Preconditioning for Krylov subspace methods: An overview, GAMM-Mitteilungen, 43 (2020), p. e202000015, https://doi.org/10.1002/gamm. 202000015.
[140] J. W. Pearson, M. Porcelli, and M. Stoll, Interior-point methods and preconditioning for PDE-constrained optimization problems involving sparsity terms, Numerical Linear Algebra with Applications, 27 (2019), p. e2276, https://doi.org/10.1002/nla. 2276.
[141] S. Pougkakiotis and J. Gondzio, Dynamic non-diagonal regularization in interior point methods for linear and convex quadratic programming, Journal of Optimization Theory and Applications, 181 (2019), pp. 905-945, https://doi.org/10.1007/s10957-019-01491-1.
[142] S. Pougkakiotis and J. Gondzio, An interior point-proximal method of multipliers for convex quadratic programming, Computational Optimization and Applications, 78 (2021), pp. 307-351, https://doi. org/10.1007/ s10589-020-00240-9.
[143] S. Pougkakiotis and J. Gondzio, An interior point-proximal method of multipliers for linear positive semi-definite programming, Journal of Optimization Theory and Applications, (2021), https://doi.org/10.1007/ s10957-021-01954-4.
[144] S. Pougkakiotis, J. W. Pearson, S. Leveque, and J. Gondzio, Fast solution methods for convex quadratic optimization of fractional differential equations, SIAM Journal on Matrix Analysis and Applications, 41 (2020), pp. 1443-1476, https://doi.org/10.1137/19M128288X.
[145] M. J. D. Powell, A method for nonlinear constraints in minimization problems, in Optimization, F. R., ed., Academic Press, 1969, pp. 283-298.
[146] R. T. Rockafellar, Augmented Lagrangians and applications of the proximal point algorithm in convex programming, Mathematics of Operations Research, 1 (1976), pp. 97-116, https://doi.org/doi.org/10. 1287/moor.1.2.97.
[147] R. T. Rockafellar, Monotone operators and the proximal point algorithm, SIAM Journal on Control and Optimization, 14 (1976), pp. 877-898, https://doi.org/10.1137/0314056.
[148] R. T. Rockafellar, Convex Analysis, Princeton University Press, 1996.
[149] M. J. Rosa, L. Portugal, T. Hahn, A. J. Fallgatter, M. I. Garrido, J. Shawe-Taylor, and J. Mourão-Miranda, Sparse networkbased models for patient classification using fMRI, NeuroImage, 105 (2015), pp. 493-506, https://doi.org/10.1016/j.neuroimage.2014.11.021.
[150] L. I. Rudin, S. Osher, and E. Fatemi, Nonlinear total variation based noise removal algorithms, Physica D, 60 (1992), pp. 259-268, https:// doi.org/10.1016/0167-2789(92) 90242-F.
[151] T. Rusten and R. Winther, A preconditioned iterative method for saddle-point problems, SIAM Journal on Matrix Analysis and Applications, 13 (1992), pp. 887-904, https://doi.org/10.1137/0613054.
[152] S. Ryali, K. Supekar, D. A. Abrams, and V. Menon, Sparse logistic regression for whole-brain classification of fMRI data, NeuroImage, 51 (2010), pp. 752-764, https://doi.org/10.1016/j.neuroimage. 2010. 02.040 .
[153] Y. Saad and M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM Journal on Scientific Computing, 7 (1986), pp. 856-869, https://doi.org/10.1137/ 09070583.
[154] M. Saunders and J. A. Tomlin, Solving regularized linear programs using barrier methods and KKT systems, Tech. Report SOL 96-4, Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, CA 94305, USA, December 1996.
[155] M. A. Saunders, Cholesky-based methods for sparse least squares: the benefits of regularization, in Linear and nonlinear conjugate gradient-related methods (Seattle, WA, 1995), SIAM, Philadelphia, PA, 1996, pp. 92-100.
[156] O. Schenk, A. WÄtcher, and M. Weiser, Inertia-revealing preconditioning for large-scale nonconvex constrained optimization, SIAM Journal on Scientific Computing, 31 (2008), pp. 939-960, https://doi.org/10. 1137/070707233.
[157] M. Schmidt, D. Kim, and S. Sra, Projected Newton-type methods in machine learning, Optimization for Machine Learning, MIT Press, 2011, pp. 305-330, https://doi.org/10.7551/mitpress/8996.003.0013.
[158] J. Scott and M. TŮMA, Solving mixed sparse-dense linear least-squares problems by preconditioned iterative methods, SIAM Journal on Scientific Computing, 39 (2017), pp. A2422-A2437, https://doi.org/10.1137/ 16M1108339.
[159] J. Scott and M. TŮma, A Schur complement approach to preconditioning sparse linear least-squares problems with some dense rows, Numerical Algorithms, 79 (2018), pp. 1147-1168, https://doi.org/10.1007/ s11075-018-0478-2.
[160] A. Shapiro, Duality and Optimality Conditions. In: Wolkowicz, H., Saigal, R., Vandenberghe, L. (eds.) Handbook of Semidefinite Programming: Theory, Algorithms and Applications, 67-110, Kluwer Academic Publishers, Boston, 2000.
[161] D. J. Silvester and A. J. Wathen, Fast iterative solution of stabilized Stokes systems part II: Using general block preconditioners, SIAM Journal on Numerical Analysis, 31 (1994), pp. 1352-1367, https://doi.org/10. 1137/0731070.
[162] M. Souto, J. D. Garcia, and A. Veiga, Exploiting low-rank structure in semidefinite programming by approximate operator splitting, Optimization, (2020), https://doi.org/10.1080/02331934.2020.1823387.
[163] G. Stadler, Elliptic optimal control problems with l ${ }^{1}$-control cost and applications for the placement of control devices, Computational Optimization and Applications, 44 (2009), pp. 159-181, https://doi. org/10.1007/ s10589-007-9150-9.
[164] G. W. Stewart and J. G. Sun, Matrix Perturbation Theory, Academic Press, 1990.
[165] R. Tapia, Y. Zhang, M. Saltzman, and A. Weiser, The Mehrotra predictor-corrector interior-point method as a perturbed composite Newton method, SIAM Journal on Optimization, 6 (1996), pp. 47-56, https:// doi.org/10.1137/0806004.
[166] R. Tibshirani, R. Saunders, S. Rosset, J. Zhu, and K. Knight, Sparsity and smoothness via the fused lasso, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67 (2005), pp. 91-108, https : //doi.org/10.1111/j.1467-9868.2005.00490.x.
[167] M. J. Todd, A study of search directions in primal-dual interiorpoint methods for semidefinite programming, Optimization Methods and Software, 11 (1999), pp. 1-46, https://doi.org/10.1080/ 10556789908805745.
[168] J. A. Tropp and S. J. Wright, Computational methods for sparse solution of linear inverse problems, Proceedings of the IEEE, 98 (2010), pp. 948-958, https://doi.org/10.1109/JPROC.2010.2044010.
[169] A. M. Turing, Rounding-off errors in matrix processes, The Quarterly Journal of Mechanics and Applied Mathematics, 1 (1948), https://doi. org/10.1093/qjmam/1.1.287.
[170] L. Vandenberghe and S. Boyd, Applications of semidefinite programming, Applied Numerical Mathematics, 29 (1999), pp. 283-299, https: //doi.org/10.1016/S0168-9274(98)00098-1.
[171] R. J. Vanderbei, Symmetric quasidefinite matrices, SIAM Journal on Optimization, 5 (1995), pp. 100-113, https://doi.org/10.1137/0805005.
[172] V. N. Vapnik, Statistical Learning Theory, John Wiley \& Sons, New York, 1998.
[173] R. A. Waltz, J. L. Morales, J. Nocedal, and D. Orban, An interior algorithm for nonlinear optimization that combines line search and trust region steps, Mathematical Programming, 107 (2006), pp. 391-408, https://doi.org/10.1007/s10107-004-0560-5.
[174] C. Wang and P. Tang, A primal majorized semismooth Newton-CG augmented Lagrangian method for large-scale linearly constrained convex programming, Computational Optimization and Applications, 68 (2017), pp. 503-532, https://doi.org/10.1007/s10589-017-9930-9.
[175] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, Image quality assessment: From error visibility to structural similarity, IEEE Transactions on Image Processing, 13 (2004), pp. 600-612, https://doi.org/10.1109/ TIP. 2003. 819861.
[176] A. J. Wathen, Realistic eigenvalue bounds for the Galerkin mass matrix, IMA Journal of Numerical Analysis, 7 (1987), pp. 449-457, https://doi. org/10.1093/imanum/7.4.449.
[177] A. G. Weldeyesus, J. Gondzio, L. He, M. Gilbert, P. ShepERD, AND A. TYAS, Adaptive solution of truss layout optimization problems with global stability constraints, Structural and Multidisciplinary Optimization, 60 (2019), pp. 2093-2111, https://doi.org/10.1007/ s00158-019-02312-9.
[178] Y.-W. Wen, R. H. Chan, and T.-Y. Zeng, Primal-dual algorithms for total variation based image restoration under Poisson noise, Science China Mathematics, 59 (2016), pp. 141-160, https://doi.org/10.1007/ s11425-015-5079-0.
[179] A. C. Williams, Complementarity theorems for linear programming, SIAM Review, 12 (1970), pp. 135-137, https://doi.org/10.1137/ 1012015.
[180] M. A. Woodbury, Inverting Modified Matrices, Statistical Research Group, Memorandum report 42, Princeton University, Princeton, NJ, 1950.
[181] S. J. Wright, Primal-Dual Interior Point Methods, SIAM Publications, SIAM, Philadelphia, USA, 1997, https://doi.org/10.1137/1. 9781611971453.
[182] O. Yamashita, M. Sato, T. Yoshioka, F. Tong, and Y. Kamitani, Sparse estimation automatically selects voxels relevant for the decoding of fMRI activity patterns, NeuroImage, 42 (2008), pp. 1414-1429, https:// doi.org/10.1016/j.neuroimage. 2008.05.050.
[183] Y. Ye, M. J. Todd, and S. Mizuno, An $O(\sqrt{n L})$-iteration homogeneous and self-dual linear programming algorithm, Mathematics of Operational Research, 19 (1994), pp. 1-256, https://doi.org/10.1287/moor.19.1. 53.
[184] G.-X. Yuan, C.-H. Ho, and C.-J. Lin, An improved GLMNET for L1regularized logistic regression, Journal of Machine Learning Research, 13 (2012), pp. 1999-2030, http://jmlr.org/papers/v13/yuan12a.html.
[185] M.-C. Yue, Z. Zhou, and A. M.-C. So, A family of inexact SQA methods for non-smooth convex minimization with provable convergence guarantees based on the Luo-Tseng error bound property, Mathematical Programming, 174 (2019), pp. 327-358, https://doi.org/10.1007/ s10107-018-1280-6.
[186] J. Zhang, On the convergence of a class of infeasible interior-point methods for the linear complementarity problem, SIAM Journal on Optimization, 4 (1994), pp. 208-227, https://doi.org/10.1137/0804012.
[187] J. Zhang, Y. Hu, and J. G. Nagy, A scaled gradient method for digital tomographic image reconstruction, Inverse Problems \& Imaging, 12 (2018), pp. 239-259, https://doi.org/10.3934/ipi. 2018010.
[188] Y. Zhang, On extending some primal-dual interior-point algorithms from linear programming to semidefinite programming, SIAM Journal on Optimization, 8 (1998), pp. 365-386, https://doi.org/10.1137/ S1052623495296115.
[189] X. Y. Zhao, D. Sun, and K. C. Toh, A Newton-CG augmented Lagrangian method for semidefinite programming, SIAM Journal on Optimization, 20 (2010), pp. 1737-1765, https://doi.org/10.1137/080718206.
[190] G. Zhou and K. C. Toh, Polynomiality of an inexact infeasible interior point algorithm for semidefinite programming, Mathematical Programming, 99 (2004), pp. 261-282, https://doi.org/10.1007/s10107-003-0431-5.
[191] G. Zhou, K. C. Toh, and G. Zhao, Convergence analysis of an infeasible interior point algorithm based on a regularized central path for linear complementarity problems, Computational Optimization and Applications, 27 (2004), pp. 269-283, https://doi.org/10.1023/B:COAP. 0000013059. 84424.af.

## Appendix A

## Convergence Analysis of IP-PMM for SDP: residual proofs

## A. 1 Proof of Lemma 4.4.2

Proof. We prove the claim by induction on the iterates, $k \geq 0$, of Algorithm IP-PMM-SDP. At iteration $k=0$, we have that $\eta_{0}=y_{0}$ and $\Xi_{0}=X_{0}$. But from the construction of the starting point in (4.18), we know that $\left\|\left(X_{0}, y_{0}\right)\right\|_{2}=\boldsymbol{O}(\sqrt{n})$. Hence, $\left\|\left(\Xi_{0}, \eta_{0}\right)\right\|_{2}=\boldsymbol{O}(\sqrt{n})$ (assuming $\left.n>m\right)$. Invoking Lemma 4.4.1, there exists a unique pair $\left(X_{r_{0}}^{*}, y_{r_{0}}^{*}\right)$ such that:

$$
\left(X_{r_{0}}^{*}, y_{r_{0}}^{*}\right)=\mathcal{P}_{0}\left(\Xi_{0}, \eta_{0}\right), \quad\left\|\left(\boldsymbol{X}_{r_{0}}^{*}, y_{r_{0}}^{*}\right)-\left(\boldsymbol{X}^{*}, y^{*}\right)\right\|_{2} \leq\left\|\left(\mathbf{\Xi}_{0}, \eta_{0}\right)-\left(\boldsymbol{X}^{*}, y^{*}\right)\right\|_{2}
$$

where $\left(X^{*}, y^{*}, Z^{*}\right)$ solves (SDP)-(SDD), and from Remark 4.4.1, is such that $\left\|\boldsymbol{X}^{*}, y^{*}, \boldsymbol{Z}^{*}\right\|_{2}=\boldsymbol{O}(\sqrt{n})$. Using the triangular inequality, and combining the latter inequality with our previous observations, yields that $\left\|\left(\boldsymbol{X}_{r_{0}}^{*}, y_{r_{0}}^{*}\right)\right\|_{2}=\boldsymbol{O}(\sqrt{n})$. From the definition of the operator in (4.6), we know that:

$$
-C+\mathcal{A}^{*} y_{r_{0}}^{*}-\mu\left(X_{r_{0}}^{*}-\Xi_{0}\right) \in \partial \delta_{\mathcal{S}_{+}^{n}}\left(X_{r_{0}}^{*}\right), \quad \mathcal{A} X_{r_{0}}^{*}+\mu\left(y_{r_{0}}^{*}-\eta_{0}\right)-b=0_{m}
$$

where $\partial\left(\delta_{\mathcal{S}_{+}^{n}}(\cdot)\right)$ is the sub-differential of the indicator function defined in (4.4). Hence, there must exist $-Z_{r_{0}}^{*} \in \partial \delta_{\mathcal{S}_{+}^{n}}\left(X_{r_{0}}^{*}\right)$ (and thus, $Z_{r_{0}}^{*} \in \mathcal{S}_{+}^{n},\left\langle X_{r_{0}}, Z_{r_{0}}\right\rangle=0$ ), such that:

$$
Z_{r_{0}}^{*}=C-\mathcal{A}^{*} y_{r_{0}}^{*}+\mu\left(X_{r_{0}}^{*}-\Xi_{0}\right), \quad\left\langle X_{r_{0}}^{*}, Z_{r_{0}}^{*}\right\rangle=0, \quad\left\|\boldsymbol{Z}_{r_{0}}^{*}\right\|_{2}=\boldsymbol{O}(\sqrt{n}),
$$

where $\left\|\boldsymbol{Z}_{r_{0}}^{*}\right\|_{2}=\boldsymbol{O}(\sqrt{n})$ follows from Assumption 4, combined with $\left\|\left(\boldsymbol{X}_{r_{0}}^{*}, y_{r_{0}}^{*}\right)\right\|_{2}=$ $\boldsymbol{O}(\sqrt{n})$.

Let $k$ be an arbitrary iteration of Algorithm IP-PMM-SDP. There are two cases for the subsequent iterations:

1. The proximal estimates are updated, that is $\left(\Xi_{k+1}, \eta_{k+1}\right)=\left(X_{k+1}, y_{k+1}\right)$, or
2. the proximal estimates stay the same, i.e. $\left(\Xi_{k+1}, \eta_{k+1}\right)=\left(\Xi_{k}, \eta_{k}\right)$.

Case 1. We know by construction that this occurs only if the following is satisfied:

$$
\left\|\left(r_{p}, \boldsymbol{R}_{d}\right)\right\|_{2} \leq K_{N} \frac{\mu_{k+1}}{\mu_{0}}
$$

where $r_{p}, R_{d}$ are defined in Algorithm IP-PMM-SDP. However, from the neighbourhood conditions in (4.19), we know that:

$$
\left\|\left(r_{p}+\mu_{k+1}\left(y_{k+1}-\eta_{k}\right), \boldsymbol{R}_{d}+\mu_{k+1}\left(\boldsymbol{X}_{k+1}-\boldsymbol{\Xi}_{k}\right)\right)\right\|_{2} \leq K_{N} \frac{\mu_{k+1}}{\mu_{0}} .
$$

Combining the last two inequalities by applying the triangular inequality, and using the properties of Algorithm IP-PMM-SDP recursively, yields that

$$
\left\|\left(\boldsymbol{X}_{k+1}, y_{k+1}\right)\right\|_{2} \leq \max \left\{k^{\dagger} \frac{2 K_{N}}{\mu_{0}}+\left\|\left(\boldsymbol{\Xi}_{0}, \eta_{0}\right)\right\|_{2}, \bar{K} \sqrt{n}\right\}=\boldsymbol{O}(\sqrt{n}) .
$$

Hence, $\left\|\left(\boldsymbol{\Xi}_{k+1}, \eta_{k+1}\right)\right\|_{2}=\boldsymbol{O}(\sqrt{n})$. Then, we can invoke Lemma 4.4.1, with $\eta=$ $\eta_{k+1}, \Xi=\Xi_{k+1}$ and any $\mu \geq 0$, which gives

$$
\left\|\left(\boldsymbol{X}_{r_{k+1}}^{*}, y_{r_{k+1}}^{*}\right)-\left(\boldsymbol{X}^{*}, y^{*}\right)\right\|_{2} \leq\left\|\left(\boldsymbol{\Xi}_{k+1}, \eta_{k+1}\right)-\left(\boldsymbol{X}^{*}, y^{*}\right)\right\|_{2} .
$$

A simple manipulation shows that $\left\|\left(\boldsymbol{X}_{r_{k+1}}^{*}, y_{r_{k+1}}^{*}\right)\right\|_{2}=\boldsymbol{O}(\sqrt{n})$. As before, we use (4.6) alongside Assumption 4 to show the existence of $-Z_{r_{k+1}}^{*} \in \partial \delta_{\mathcal{S}_{+}^{n}}\left(X_{r_{k+1}}^{*}\right)$, such that the triple $\left(X_{r_{k+1}}^{*}, y_{r_{k+1}}^{*}, Z_{r_{k+1}}^{*}\right)$ satisfies (4.24) with $\left\|\boldsymbol{Z}_{r_{k+1}}^{*}\right\|_{2}=\boldsymbol{O}(\sqrt{n})$.

Case 2. In this case, we have $\left(\Xi_{k+1}, \eta_{k+1}\right)=\left(\Xi_{k}, \eta_{k}\right)$. However, from the properties of Algorithm IP-PMM-SDP we can show that $\left\|\left(\boldsymbol{\Xi}_{\boldsymbol{k}}, \eta_{k}\right)\right\|_{2}=\boldsymbol{O}(\sqrt{n})$, for all $k$. Thus, $\left\|\left(\boldsymbol{\Xi}_{k+1}, \eta_{k+1}\right)\right\|_{2}=\boldsymbol{O}(\sqrt{n})$. As before, there exists a triple $\left(X_{r_{k+1}}^{*}, y_{r_{k+1}}^{*}, Z_{r_{k+1}}^{*}\right)$ satisfying (4.24), with $\left\|\left(\boldsymbol{X}_{r_{k+1}}^{*}, y_{r_{k+1}}^{*}, \boldsymbol{Z}_{r_{k+1}}^{*}\right)\right\|_{2}=\boldsymbol{O}(\sqrt{n})$.

## A. 2 Proof of Lemma 4.4.4

Proof. Let an iterate $\left(X_{k}, y_{k}, Z_{k}\right) \in \mathscr{N}_{\mu_{k}}\left(\Xi_{k}, \eta_{k}\right)$, produced by Algorithm IP-PMM-SDP during an arbitrary iteration $k \geq 0$, be given. Firstly, we invoke Lemma 4.4.3, from which we have a triple $(\tilde{X}, \tilde{y}, \tilde{Z})$ satisfying (4.25), for $\mu=$ $\mu_{k}$. Similarly, by invoking Lemma 4.4.2, we know that there exists a triple $\left(X_{r_{k}}^{*}, y_{r_{k}}^{*}, Z_{r_{k}}^{*}\right)$ satisfying (4.24), with $\mu=\mu_{k}$. Consider the following auxiliary point:

$$
\begin{equation*}
\left(\left(1-\frac{\mu_{k}}{\mu_{0}}\right) X_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{X}-X_{k},\left(1-\frac{\mu_{k}}{\mu_{0}}\right) y_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{y}-y_{k},\left(1-\frac{\mu_{k}}{\mu_{0}}\right) Z_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{Z}-Z_{k}\right) . \tag{A.1}
\end{equation*}
$$

Using (A.1) and (4.24)-(4.25) (for $\mu=\mu_{k}$ ), one can observe that:

$$
\begin{aligned}
A\left(\left(1-\frac{\mu_{k}}{\mu_{0}}\right) \boldsymbol{X}_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{\boldsymbol{X}}-\boldsymbol{X}_{k}\right)+\mu_{k}\left(\left(1-\frac{\mu_{k}}{\mu_{0}}\right) y_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{y}-y_{k}\right) & = \\
\left(1-\frac{\mu_{k}}{\mu_{0}}\right)\left(A \boldsymbol{X}_{r_{k}}^{*}+\mu_{k} y_{r_{k}}^{*}\right)+\frac{\mu_{k}}{\mu_{0}}\left(A \tilde{\boldsymbol{X}}+\mu_{k} \tilde{y}\right)-A \boldsymbol{X}_{k}-\mu_{k} y_{k} & = \\
\left(1-\frac{\mu_{k}}{\mu_{0}}\right)\left(b+\mu_{k} \eta_{k}\right)+\frac{\mu_{k}}{\mu_{0}}\left(b+\mu_{k} \eta_{k}+\tilde{b}_{k}+\bar{b}\right)-A \boldsymbol{X}_{k}-\mu_{k} y_{k} & = \\
b+\mu_{k} \eta_{k}+\frac{\mu_{k}}{\mu_{0}}\left(\tilde{b}_{k}+\bar{b}\right)-A \boldsymbol{X}_{k}-\mu_{k} y_{k} & =0_{m},
\end{aligned}
$$

where the last equality follows from the definition of the neighbourhood $\mathscr{N}_{\mu_{k}}\left(\Xi_{k}, \eta_{k}\right)$. Similarly, one can show that:

$$
\begin{aligned}
-\mu_{k}\left(\left(1-\frac{\mu_{k}}{\mu_{0}}\right) \boldsymbol{X}_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{\boldsymbol{X}}-\boldsymbol{X}_{k}\right) & +A^{\top}\left(\left(1-\frac{\mu_{k}}{\mu_{0}}\right) y_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{y}-y_{k}\right) \\
& +\left(\left(1-\frac{\mu_{k}}{\mu_{0}}\right) \boldsymbol{Z}_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{\boldsymbol{Z}}-\boldsymbol{Z}_{k}\right)=0_{n^{2}}
\end{aligned}
$$

By combining the previous two relations, we have:

$$
\begin{gather*}
\left(\left(1-\frac{\mu_{k}}{\mu_{0}}\right) \boldsymbol{X}_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{\boldsymbol{X}}-\boldsymbol{X}_{k}\right)^{\top}\left(\left(1-\frac{\mu_{k}}{\mu_{0}}\right) \boldsymbol{Z}_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{\boldsymbol{Z}}-\boldsymbol{Z}_{k}\right)= \\
\mu_{k}\left(\left(1-\frac{\mu_{k}}{\mu_{0}}\right) \boldsymbol{X}_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{\boldsymbol{X}}-\boldsymbol{X}_{k}\right)^{\top}\left(\left(1-\frac{\mu_{k}}{\mu_{0}}\right) \boldsymbol{X}_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{\boldsymbol{X}}-\boldsymbol{X}_{k}\right)  \tag{A.2}\\
+\mu_{k}\left(\left(1-\frac{\mu_{k}}{\mu_{0}}\right) y_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{y}-y_{k}\right)^{\top}\left(\left(1-\frac{\mu_{k}}{\mu_{0}}\right) y_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{y}-y_{k}\right) \geq 0 .
\end{gather*}
$$

Observe that (A.2) can equivalently be written as:

$$
\begin{aligned}
& \left\langle\left(1-\frac{\mu_{k}}{\mu_{0}}\right) X_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{X}, Z_{k}\right\rangle+\left\langle\left(1-\frac{\mu_{k}}{\mu_{0}}\right) Z_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{Z}, X_{k}\right\rangle \\
& \quad \leq\left\langle\left(1-\frac{\mu_{k}}{\mu_{0}}\right) X_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{X},\left(1-\frac{\mu_{k}}{\mu_{0}}\right) Z_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{Z}\right\rangle+\left\langle X_{k}, Z_{k}\right\rangle .
\end{aligned}
$$

However, from Lemmas 4.4.2 and 4.4.3, we have that $\tilde{X} \succeq \xi I_{n}$ and $\tilde{Z} \succeq \xi I_{n}$, for some positive constant $\xi=\Theta(1),\left\langle X_{r_{k}}^{*}, Z_{k}\right\rangle \geq 0,\left\langle Z_{r_{k}}^{*}, X_{k}\right\rangle \geq 0$, while $\left\|\left(X_{r_{k}}^{*}, Z_{r_{k}}^{*}\right)\right\|_{F}=\boldsymbol{O}(\sqrt{n})$, and $\|(\tilde{X}, \tilde{Z})\|_{F}=\boldsymbol{O}(\sqrt{n})$. Furthermore, by definition we have that $n \mu_{k}=\left\langle X_{k}, Z_{k}\right\rangle$. By combining all the previous, we obtain:

$$
\begin{align*}
\frac{\mu_{k}}{\mu_{0}} \xi\left(\operatorname{Tr}\left(X_{k}\right)+\operatorname{Tr}\left(Z_{k}\right)\right) & = \\
\frac{\mu_{k}}{\mu_{0}} \xi\left(\left\langle I_{n}, X_{k}\right\rangle+\left\langle I_{n}, Z_{k}\right\rangle\right) & \leq \\
\left\langle\left(1-\frac{\mu_{k}}{\mu_{0}}\right) X_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{X}, Z_{k}\right\rangle+\left\langle\left(1-\frac{\mu_{k}}{\mu_{0}}\right) Z_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{Z}, X_{k}\right\rangle & \leq  \tag{A.3}\\
\left\langle\left(1-\frac{\mu_{k}}{\mu_{0}}\right) X_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{X},\left(1-\frac{\mu_{k}}{\mu_{0}}\right) Z_{r_{k}}^{*}+\frac{\mu_{k}}{\mu_{0}} \tilde{Z}\right\rangle+\left\langle X_{k}, Z_{k}\right\rangle & = \\
\frac{\mu_{k}}{\mu_{0}}\left(1-\frac{\mu_{k}}{\mu_{0}}\right)\left\langle X_{r_{k}}^{*}, \tilde{Z}\right\rangle+\frac{\mu_{k}}{\mu_{0}}\left(1-\frac{\mu_{k}}{\mu_{0}}\right)\left\langle\tilde{X}, Z_{r}^{*}\right\rangle+\left(\frac{\mu_{k}}{\mu_{0}}\right)^{2}\langle\tilde{X}, \tilde{Z}\rangle+\left\langle X_{k}, Z_{k}\right\rangle & =\boldsymbol{O}\left(n \mu_{k}\right),
\end{align*}
$$

where the first inequality follows since $X_{r_{k}}^{*}, Z_{r_{k}}^{*}, \tilde{X}, \tilde{Z} \in \mathcal{S}_{+}^{n}$ and $(\tilde{X}, \tilde{Z}) \succeq$ $\xi\left(I_{n}, I_{n}\right)$. In the penultimate equality we used (4.24) (i.e. $\left\langle X_{r_{k}}^{*}, Z_{r_{k}}^{*}\right\rangle=0$ ). Hence, (A.3) implies that:

$$
\operatorname{Tr}\left(X_{k}\right)=\boldsymbol{O}(n), \quad \operatorname{Tr}\left(Z_{k}\right)=\boldsymbol{O}(n)
$$

From positive definiteness we have that $\left\|\left(X_{k}, Z_{k}\right)\right\|_{F}=\boldsymbol{O}(n)$. Finally, from the neighbourhood conditions we know that:

$$
\boldsymbol{C}-A^{\top} y_{k}-\boldsymbol{Z}_{k}+\mu_{k}\left(\boldsymbol{X}_{k}-\boldsymbol{\Xi}_{k}\right)+\frac{\mu_{k}}{\mu_{0}}\left(\tilde{\boldsymbol{C}}_{k}+\overline{\boldsymbol{C}}\right)=0_{n^{2}} .
$$

All terms above (except for $y_{k}$ ) have a 2 -norm that is bounded by some quantity that is $\boldsymbol{O}(n)$ (note that $\|(\overline{\boldsymbol{C}}, \bar{b})\|_{2}=\boldsymbol{O}(\sqrt{n})$ using Assumption 4 and the definition in (4.18)). Hence, using again Assumption 4 (i.e. $A$ is full rank, with singular values independent of $n$ and $m$ ) yields that $\left\|y_{k}\right\|_{2}=\boldsymbol{O}(n)$, and completes the proof.

## A. 3 Proof of Theorem 4.4.1

Proof. From (4.36) we have that:

$$
\mu_{k+1} \leq\left(1-0.01 \alpha_{k}\right) \mu_{k},
$$

while, from (4.46), we know that $\forall k \geq 0, \exists \bar{\alpha} \geq \frac{\bar{\varepsilon}}{n^{4}}$ such that $\alpha_{k} \geq \bar{\alpha}$. Hence, we can easily see that $\mu_{k} \rightarrow 0$. On the other hand, from the neighbourhood conditions, we know that for all $k \geq 0$ :

$$
\left\|A \boldsymbol{X}_{k}+\mu_{k}\left(y_{k}-\eta_{k}\right)-b-\frac{\mu_{k}}{\mu_{0}} \bar{b}\right\|_{2} \leq K_{N} \frac{\mu_{k}}{\mu_{0}}
$$

and

$$
\left\|A^{\top} y_{k}+\boldsymbol{Z}_{k}-\mu_{k}\left(\boldsymbol{X}_{k}-\boldsymbol{\Xi}_{k}\right)-\boldsymbol{C}-\frac{\mu_{k}}{\mu_{0}} \overline{\boldsymbol{C}}\right\|_{2} \leq K_{N} \frac{\mu_{k}}{\mu_{0}} .
$$

This completes the proof.

## A. 4 Proof of Theorem 4.4.3

Proof. From Theorem 4.4.1, we know that $\left\{\mu_{k}\right\} \rightarrow 0$, and hence, there exists a sub-sequence $\mathcal{K} \subseteq \mathbb{N}$, such that:

$$
\left\{A \boldsymbol{X}_{k}+\mu_{k}\left(y_{k}-\eta_{k}\right)-b-\frac{\mu_{k}}{\mu_{0}} \bar{b}\right\}_{\mathcal{K}} \rightarrow 0_{m}
$$

and

$$
\left\{A^{\top} y_{k}+\boldsymbol{Z}_{k}-\mu_{k}\left(\boldsymbol{X}_{k}-\boldsymbol{\Xi}_{k}\right)-\boldsymbol{C}-\frac{\mu_{k}}{\mu_{0}} \overline{\boldsymbol{C}}\right\}_{\mathcal{K}} \rightarrow 0_{n^{2}}
$$

However, since Assumptions 3 and 4 hold, we know from Lemma 4.4.4 that $\left\{\left(X_{k}, y_{k}, Z_{k}\right)\right\}$ is a bounded sequence. Hence, we obtain that:

$$
\left\{A \boldsymbol{X}_{k}-b\right\}_{\mathcal{K}} \rightarrow 0_{m},\left\{A^{\top} y_{k}+\boldsymbol{Z}_{k}-\boldsymbol{C}\right\}_{\mathcal{K}} \rightarrow 0_{n^{2}}
$$

One can readily observe that the limit point of the algorithm satisfies the optimality conditions of (SDP)-(SDD), since $\left\langle X_{k}, Z_{k}\right\rangle \rightarrow 0$ and $X_{k}, Z_{k} \in \mathcal{S}_{+}^{n}$.

## A. 5 Proof of Theorem 4.5.1

Proof. By virtue of contradiction, let Premise 3 hold. In Lemma 4.5.2, we proved that given Premise 3, Algorithm IP-PMM-SDP produces iterates that belong to the neighbourhood (4.19) and $\mu_{k} \rightarrow 0$. But from the neighbourhood conditions we can observe that:

$$
\left\|A \boldsymbol{X}_{k}+\mu_{k}\left(y_{k}-\eta_{k}\right)-b-\frac{\mu_{k}}{\mu_{0}} \bar{b}\right\|_{2} \leq K_{N} \frac{\mu_{k}}{\mu_{0}},
$$

and

$$
\left\|A^{\top} y_{k}+\boldsymbol{Z}_{k}-\mu_{k}\left(\boldsymbol{X}_{k}-\boldsymbol{\Xi}_{k}\right)-\boldsymbol{C}-\frac{\mu_{k}}{\mu_{0}} \overline{\boldsymbol{C}}\right\|_{2} \leq K_{N} \frac{\mu_{k}}{\mu_{0}}
$$

Hence, we can choose a sub-sequence $\mathcal{K} \subseteq \mathbb{N}$, for which:

$$
\left\{A \boldsymbol{X}_{k}+\mu_{k}\left(y_{k}-\eta_{k}\right)-b-\frac{\mu_{k}}{\mu_{0}} \bar{b}\right\}_{\mathcal{K}} \rightarrow 0_{m}
$$

and

$$
\left\{A^{\top} y_{k}+\boldsymbol{Z}_{k}-\mu_{k}\left(\boldsymbol{X}_{k}-\boldsymbol{\Xi}_{k}\right)-\boldsymbol{C}-\frac{\mu_{k}}{\mu_{0}} \overline{\boldsymbol{C}}\right\}_{\mathcal{K}} \rightarrow 0_{n^{2}}
$$

But since $\left\|y_{k}-\eta_{k}\right\|_{2}$ and $\left\|X_{k}-\Xi_{k}\right\|_{F}$ are bounded, while $\mu_{k} \rightarrow 0$, we have that:

$$
\left\{A \boldsymbol{X}_{k}-b\right\}_{\mathcal{K}} \rightarrow 0_{m},\left\{\boldsymbol{C}-A^{\top} y_{k}-\boldsymbol{Z}_{k}\right\}_{\mathcal{K}} \rightarrow 0_{n^{2}}, \text { and }\left\{\left\langle X_{k}, Z_{k}\right\rangle\right\}_{\mathcal{K}} \rightarrow 0
$$

This contradicts Premise 4, i.e. that the pair (SDP)-(SDD) does not have a KKT triple, and completes the proof.


[^0]:    ${ }^{1}$ https://www.maths.ed.ac.uk/ERGO/software.html

[^1]:    ${ }^{1}$ https://github.com/spougkakiotis/IP-PMM_QP_Solver
    2https://github.com/spougkakiotis/Inexact_IP-PMM

[^2]:    ${ }^{1} *$ indicates that the solver was stopped due to excessive run time.
    ${ }^{2} \dagger$ indicates that the solver ran out of memory.

[^3]:    ${ }^{3}$ https://personalpages.manchester.ac.uk/staff/david.silvester/ifiss/default. htm

[^4]:    ${ }^{1}$ available from https://github.com/lucabaldassarre/neurosparse

[^5]:    ${ }^{2}$ http://ranger.uta.edu/~huang/R_CSMRI.htm

[^6]:    ${ }^{3}$ available from https://web.stanford.edu/group/SOL/software/minres/

[^7]:    ${ }^{4}$ available from https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/binary. html

[^8]:    ${ }^{5}$ available from http://www.stanford.edu/~boyd/papers/distr_opt_stat_learning_ admm.html
    ${ }^{6}$ available from https://github.com/ZiruiZhou/IRPN

