

An Interior Point–Proximal Method of Multipliers for Convex Quadratic Programming

Spyridon Pougkakiotis

Jacek Gondzio

April 23, 2019

Abstract

In this paper, we combine an infeasible Interior Point Method (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 very mild assumptions. To our knowledge, this is the first polynomial complexity result for a primal–dual regularized IPM. 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. The numerical results demonstrate the benefits of using regularization in IPMs as well as the reliability of the method.

1 Introduction

1.1 Primal–Dual Pair of Convex Quadratic Programming Problems

In this paper, we consider the following primal–dual pair of linearly constrained convex quadratic programming problems, in the standard form:

$$\min_x \left(c^T x + \frac{1}{2} x^T Q x \right), \quad \text{s.t. } Ax = b, \quad x \geq 0 \quad (\text{P})$$

$$\max_{x,y,z} \left(b^T y - \frac{1}{2} x^T Q x \right), \quad \text{s.t. } -Qx + A^T y + z = c, \quad z \geq 0, \quad (\text{D})$$

where $c, x, z \in \mathbb{R}^n$, $b, y \in \mathbb{R}^m$, $A \in \mathbb{R}^{m \times n}$, $Q \succeq 0 \in \mathbb{R}^{n \times n}$. Without loss of generality $m \leq n$. Duality, between the problems (P)–(D), arises by introducing the Lagrangian function of the primal, using $y \in \mathbb{R}^m$ and $z \in \mathbb{R}^n$, $z \geq 0$, as the Lagrange multipliers for the equality and inequality constraints, respectively. Hence, we have:

$$\mathcal{L}(x, y, z) = c^T x + \frac{1}{2} x^T Q x - y^T (Ax - b) - z^T x \quad (1.1)$$

Using the Lagrangian function, one can formulate the first–order optimality conditions (known as Karush–Kuhn–Tucker (KKT) conditions) for this primal–dual pair. In particular, we define

the vector $w = (x^T, y^T, z^T)^T$, and compute the gradient of $\mathcal{L}(\cdot)$, that is, $F(w) = \nabla \mathcal{L}(w)$, where $F(w) : \mathbb{R}^{2n+m} \mapsto \mathbb{R}^{2n+m}$. Then, equating F to the zero vector gives the desired conditions, i.e.:

$$F(w) = \begin{bmatrix} c + Qx - A^T y - z \\ Ax - b \\ XZe \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \quad (1.2)$$

where e denotes the vector of ones of appropriate dimension, while $X, Z \in \mathbb{R}^{n \times n}$ denote the diagonal matrices satisfying $X_{ii} = x_i$ and $Z_{ii} = z_i, \forall i \in \{1, \dots, n\}$. For simplicity of exposition, when referring to convex quadratic programming problems, we implicitly assume that the problems are linearly constrained. If both (P) and (D) are feasible problems, it can easily be verified that there exists an optimal primal–dual triple (x, y, z) , satisfying the KKT optimality conditions of this primal–dual pair (see for example Prop. 2.3.4 in [7]).

1.2 A Primal–Dual Interior Point Method

Primal–dual Interior Point Methods (IPMs) are popular for simultaneously solving (P) and (D), iteratively. 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 [16] for an extended literature review. In this paper, an infeasible primal–dual IPM is presented. Such methods are called infeasible because they allow intermediate iterates of the method to be infeasible for (P)–(D), in contrast to feasible IPMs, which require intermediate iterates to be strictly feasible.

Interior point methods, replace the non–negativity constraints of the problems with logarithmic barriers in the objective. That is, at each iteration, we choose a *barrier parameter* μ and form the logarithmic *barrier primal–dual pair*:

$$\min_x \left(c^T x + \frac{1}{2} x^T Q x - \mu \sum_{j=1}^n \ln x_j \right) \quad \text{s.t.} \quad Ax = b, \quad (1.3)$$

$$\max_{x,y,z} \left(b^T y - \frac{1}{2} x^T Q x - \mu \sum_{j=1}^n \ln z_j \right) \quad \text{s.t.} \quad -Qx + A^T y + z = c, \quad (1.4)$$

in which non–negativity constraints $x > 0$ and $z > 0$ are implicit. As before, we can form the KKT optimality conditions of the pair (1.3)–(1.4), by introducing the Lagrangian of the primal barrier problem:

$$\mathcal{L}_\mu^{\text{IPM}}(x, y) = c^T x + \frac{1}{2} x^T Q x - y^T (Ax - b) - \mu \sum_{i=1}^n \ln x_i.$$

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

$$\begin{aligned} \nabla_x \mathcal{L}_\mu^{\text{IPM}}(x, y) &= c + Qx - A^T y - \mu X^{-1} e = 0, \\ \nabla_y \mathcal{L}_\mu^{\text{IPM}}(x, y) &= Ax - b = 0. \end{aligned}$$

Applying the same methodology for (1.4), gives the final condition, namely:

$$XZe = \mu e.$$

At each IPM iteration, we want to solve the following mildly non-linear system:

$$F_{\sigma,\mu}^{\text{IPM}}(w) = \begin{bmatrix} c + Qx - A^T y - z \\ b - Ax \\ XZe \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \sigma\mu e \end{bmatrix}, \quad (1.5)$$

where $F_{\sigma,\mu}^{\text{IPM}}(w) = 0$ is a slightly perturbed form of the previously presented optimality conditions. In particular, $\sigma \in (0, 1)$ is a *centering parameter*, which determines how fast μ will be forced to decrease at the next IPM iteration. For $\sigma = 1$, we recover the barrier optimality conditions, while for $\sigma = 0$ we recover the initial problems' optimality conditions (1.2). The efficiency of the method depends heavily on the choice of σ . 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 σ and of the right hand side, at each IPM iteration. These are the so called *predictor-corrector* schemes, proposed for the first time in [22]. Various extensions of such methods have been proposed and analysed in the literature (e.g. see [15], [32] and references therein). However, for simplicity of exposition, we will follow the traditional approach; σ is chosen heuristically at each iteration, and the previous system is solved only once.

In order to solve the system $F_{\sigma,\mu}^{\text{IPM}}(w) = 0$ at each iteration, the most common approach is to apply the Newton method. Newton method is favored for systems of this form, due to the *self-concordance* of the function $\ln(\cdot)$. For more details on this subject, the interested reader is referred to Chapter 2 in [23]. At the beginning of the k -th iteration of the IPM, for $k \geq 0$, we have available an iterate $w_k = (x_k^T, y_k^T, z_k^T)^T$, and a barrier parameter μ_k , defined as $\mu_k = \frac{x_k^T z_k}{n}$. We choose a value for the centering parameter $\sigma_k \in (0, 1)$ and evaluate the Jacobian of $F_{\sigma_k, \mu_k}^{\text{IPM}}(w^k)$. Then, the Newton direction is determined by solving the following system of equations:

$$\begin{bmatrix} -Q & A^T & I \\ A & 0 & 0 \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = - \begin{bmatrix} -c - Qx_k + A^T y_k + z_k \\ Ax_k - b \\ XZe - \sigma_k \mu_k e \end{bmatrix}. \quad (1.6)$$

For ease of presentation, when a matrix depends on the iteration k (in this case X and Z), we omit the subscript and state the dependence whenever it is not obvious. Notice that as $\mu \rightarrow 0$, the optimal solution of (1.3)–(1.4) converges to the optimal solution of (P)–(D). Polynomial convergence of such methods has been proved multiple times in the literature (see for example Chapter 6 in [33], [34] and the references therein).

1.3 Proximal Point Methods

1.3.1 Primal Proximal Point Method

One possible method for solving the primal problem (P), 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:

$$x_{k+1} = \arg \min_x \left\{ c^T x + \frac{1}{2} x^T Q x + \frac{\rho_k}{2} \|x - x_k\|^2, \text{ s.t. } Ax = b, x \geq 0 \right\},$$

where $\rho_k > 0$ is a non-increasing sequence of penalty parameters, and x_k is the current estimate for the optimal solution of (P). Notice that such an algorithm is not practical, since we have to solve a sequence of sub-problems, of similar difficulty to that of (P). Nevertheless, the proximal method contributes the $\rho_k I$ term 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 superlinear if $\rho_k \rightarrow 0$), as shown in [29, 13], 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 (see [9, 18, 11] and references therein). 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 [26].

1.3.2 Dual Proximal Point Method

It is a well-known fact, observed for the first time in [29], that the application of the proximal point method on the dual problem, is equivalent to the application of the *augmented Lagrangian method* on the primal problem, which was proposed for the first time in [17, 28]. In view of the previous fact, we will present the derivation of the augmented Lagrangian method for the primal problem (P), while having in mind that an equivalent reformulation of the model, results in the proximal point method for the dual (D) (see [29, 12] or Chapter 3.4.4 in [8]).

We start by defining the augmented Lagrangian function of (P). Given an arbitrary starting guess for the dual variables y_0 , the augmented Lagrangian function is defined at the k -th iteration as:

$$\mathcal{L}_{\delta_k, y_k}^{\text{ALM}}(x) = c^T x + \frac{1}{2} x^T Q x - y_k^T (Ax - b) + \frac{1}{2\delta_k} \|Ax - b\|^2, \quad (1.7)$$

where $\delta_k > 0$ is a non-increasing sequence of penalty parameters and y_k is the current estimate of the optimal Lagrange multiplier vector. The augmented Lagrangian method (ALM) is summarized below:

$$\begin{aligned} x_{k+1} &= \arg \min_x \left\{ \mathcal{L}_{\delta_k, y_k}^{\text{ALM}}(x), \text{ s.t. } x \geq 0 \right\}, \\ y_{k+1} &= y_k - \frac{1}{\delta_k} (Ax_{k+1} - b). \end{aligned}$$

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 $\delta_k > 0$). Convergence and iteration complexity of such methods have been established multiple times (see for example [29, 20]). 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 [6]. For a convergence result of various extended versions of the method, the reader is referred to [11].

1.3.3 Primal–Dual Proximal Point Method

In [29], 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 (P)–(D). For an arbitrary starting point (x_0, y_0) , and using the already defined augmented Lagrangian function, given in (1.7), PMM can be summarized by the following iteration:

$$\begin{aligned} x_{k+1} &= \arg \min_x \{ \mathcal{L}_{\delta_k, y_k}^{\text{ALM}}(x) + \frac{\rho_k}{2} \|x - x_k\|^2, \text{ s.t. } x \geq 0 \}, \\ y_{k+1} &= y_k - \frac{1}{\delta_k} (Ax_{k+1} - b), \end{aligned} \tag{1.8}$$

where ρ_k, δ_k are positive non-increasing penalty parameters. 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 [29], the addition of the term $\rho_k \|x - x_k\|^2$, 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 [11].

1.4 Regularization in Interior Point Methods

In the context of interior point methods, it is often beneficial to include a regularization, in order to improve the spectral properties of the system matrix in (1.6). For example, notice that if the constraint matrix A is rank-deficient, then the matrix in (1.6) is not invertible. The latter can be immediately addressed by the introduction of a dual regularization, say $\delta > 0$, ensuring that $\text{rank}([A \mid \delta I_m]) = m$. For a detailed analysis of the effect of dual regularization on such systems, the reader is referred to [2]. On the other hand, the most common and efficient approach in practice, is to eliminate the variables Δz from system (1.6) and solve the following symmetric reduced (*augmented*) system instead:

$$\begin{bmatrix} -(Q + \Theta^{-1}) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} c + Qx_k - A^T y_k - \sigma_k \mu_k X^{-1} e \\ b - Ax_k \end{bmatrix}, \tag{1.9}$$

where $\Theta = XZ^{-1}$. Since $Xz \rightarrow 0$ close to optimality, one can observe that the matrix Θ will contain some very large and some very small elements. Hence, the matrix in (1.9) will be increasingly ill-conditioned, as we approach optimality. The introduction of a primal regularization, say $\rho > 0$, can ensure that the matrix $Q + \Theta^{-1} + \rho I_n$, will have eigenvalues that are bounded away from zero, and hence a significantly better worst-case conditioning than that of $Q + \Theta^{-1}$. In other words, regularization is commonly employed in IPMs, as a means of improving robustness and numerical stability. As we will discuss later, by introducing regularization in IPMs, one can also gain efficiency. This is because regularization transforms the matrix in (1.9) into a quasi-definite one. It is known that such matrices can be factorized efficiently (see [31]).

In view of the previous discussion, one can observe that a very natural way of introducing primal regularization to problem (P), 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, and interpreted this regularization as the application of the proximal point

method. Consequently, the authors in [12] developed a primal–dual regularized IPM, through the primal–dual application of the proximal point method on the primal barrier problem (1.3). There, global convergence of the method was proved, under some assumptions. A variation of the method proposed in [12] is given in [27], where general non–diagonal regularization matrices are employed, as a means of improving factorization efficiency.

Similar ideas have been applied in the context of IPMs for general non–linear programming problems. For example, the authors in [3], presented an interior point method combined with the augmented Lagrangian method, and proved that, under some general assumptions, the method converges at a superlinear rate. Similar approaches can be found in [5, 14, 4], among others.

In this paper, we develop a path–following primal–dual regularized IPM for solving convex quadratic programming problems. The algorithm is obtained by applying one or a few iterations of the infeasible primal–dual interior point method, in order to solve sub–problems arising from the primal–dual proximal point method. Under some mild assumptions, we prove polynomial complexity of the algorithm and provide global convergence guarantees. To our knowledge, this is the first 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 [35]. However, the authors significantly alter the Newton directions, making their method very difficult to generalize and hard to achieve efficiency in practice. 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 paper, 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. Such a method is implemented and 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 side–product of the theory, an implementable infeasibility detection mechanism is also derived and tested in practice.

In Section 2, we provide the algorithmic framework of the method. Consequently, in Section 3, we prove polynomial complexity of the algorithm, and global convergence is established. In Section 4, an infeasibility test is derived. Numerical results of the implemented method, are presented and discussed in Section 5. Finally, we derive some conclusions in Section 6.

2 Algorithmic Framework

In the previous section, we presented all the necessary tools for deriving the proposed Interior Point–Proximal Method of Multipliers (IP–PMM). Effectively, we would like to merge the proximal method of multipliers with the infeasible interior point method. For that purpose, assume that, at some iteration k of the method, we have available an estimate λ_k for the optimal Lagrange multiplier vector y^* . Similarly, we denote by ζ_k the estimate of the primal solution x^* . Now, we define the proximal penalty function that has to be minimized at the k –th iteration of proximal method of multipliers, for solving (P), given the estimates λ_k, ζ_k :

$$\mathcal{L}_{\delta_k, \rho_k, \lambda_k, \zeta_k}^{PMM}(x) = c^T x + \frac{1}{2} x^T Q x - \lambda_k^T (Ax - b) + \frac{1}{2\delta_k} \|Ax - b\|^2 + \frac{\rho_k}{2} \|x - \zeta_k\|^2,$$

with $\delta_k > 0$, $\rho_k > 0$ some non-increasing penalty parameters. In order to solve the PMM sub-problem (1.8), we will apply one (or a few) iterations of the previously presented infeasible IPM. To do that, we alter the previous penalty function, by including logarithmic barriers, that is:

$$\mathcal{L}_{\delta_k, \rho_k, \lambda_k, \zeta_k}^{IP-PMM}(x) = c^T x + \frac{1}{2} x^T Q x - \lambda_k^T (Ax - b) + \frac{1}{2\delta_k} \|Ax - b\|^2 + \frac{\rho_k}{2} \|x - \zeta_k\|^2 - \mu_k \sum_{j=1}^n \ln x_j, \quad (2.1)$$

where $\mu_k > 0$ is the barrier parameter. In order to form the optimality conditions of this sub-problem, we equate the gradient of $\mathcal{L}_{\delta_k, \rho_k, \lambda_k, \zeta_k}^{IP-PMM}(\cdot)$ to the zero vector, i.e.:

$$c + Qx - A^T \lambda_k + \frac{1}{\delta_k} A^T (Ax - b) + \rho_k (x - \zeta_k) - \mu_k X^{-1} e = 0.$$

Following the developments in [3], we can define the variables: $y = \lambda_k - \frac{1}{\delta_k} (Ax - b)$ and $z = \mu_k X^{-1} e$, to get the following (equivalent) system of equations (first-order optimality conditions):

$$\begin{bmatrix} c + Qx - A^T y - z + \rho_k (x - \zeta_k) \\ Ax + \delta_k (y - \lambda_k) - b \\ Xz - \mu_k e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}.$$

As shown before, interior point methods introduce a perturbation in the third block equation of the previous conditions. Hence, at each IPM iteration, we choose a centering parameter $\sigma_k > 0$, and we want to solve the following system:

$$\begin{bmatrix} c + Qx - A^T y - z + \rho_k (x - \zeta_k) \\ Ax + \delta_k (y - \lambda_k) - b \\ Xz - \sigma_k \mu_k e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}. \quad (2.2)$$

Observe the similarity between the systems given in (1.2), (1.5) and (2.2). In fact, the PMM system can be written as:

$$F(w) + \begin{bmatrix} \rho_k (x - \zeta_k) \\ \delta_k (y - \lambda_k) \\ 0 \end{bmatrix} = 0,$$

while the IPM system as:

$$F(w) + \begin{bmatrix} 0 \\ 0 \\ -\sigma_k \mu_k e \end{bmatrix} = 0,$$

for $w = (x^T, y^T, z^T)^T$ and $F(w)$ as defined in (1.2). Notice that each PMM sub-problem is parametrized by δ , ρ , λ and ζ . Here, we interpret $\delta > 0$ as the dual regularization parameter, while $\rho > 0$ as the primal one. The interpretations of λ and ζ have been already stated. Hence,

for each such sub–problem, we can define the following sets:

$$\begin{aligned}
\mathcal{A}_{\delta,\rho,\lambda,\zeta} &:= \left\{ (x, y, z) : \begin{array}{l} Ax + \delta(y - \lambda) = b \\ -Qx + A^T y + z - \rho(x - \zeta) = c \end{array} \right\} \\
\mathcal{F}_{\delta,\rho,\lambda,\zeta} &:= \left\{ (x, y, z) \in \mathcal{A}_{\delta,\rho,\lambda,\zeta} : x \geq 0, z \geq 0 \right\} \\
\mathcal{F}_{\delta,\rho,\lambda,\zeta}^+ &:= \left\{ (x, y, z) \in \mathcal{A}_{\delta,\rho,\lambda,\zeta} : x > 0, z > 0 \right\} \\
\mathcal{S}_{\delta,\rho,\lambda,\zeta} &:= \left\{ (x, y, z) \in \mathcal{F}_{\delta,\rho,\lambda,\zeta} : Xz = 0 \right\}
\end{aligned} \tag{2.3}$$

Notice that $\mathcal{F}_{\delta,\rho,\lambda,\zeta}$ and $\mathcal{F}_{\delta,\rho,\lambda,\zeta}^+$ represent the feasibility and strict feasibility sets of the sub–problem, respectively. $\mathcal{S}_{\delta,\rho,\lambda,\zeta}$ represents the solution set. Let us introduce the following notation, that will be used later.

Definition 1. *Given two real-valued functions $T(x)$, $f(x)$, $x \in \mathbb{R}_+$, we say that:*

- $T(x) = O(f(x))$ if and only if, \exists constants c , x_0 , such that:

$$T(x) \leq cf(x), \forall x \geq x_0.$$

- $T(x) = \Omega(f(x))$ if and only if, \exists constants c , x_0 , such that:

$$T(x) \geq cf(x), \forall x \geq x_0.$$

- $T(x) = \Theta(f(x))$ if and only if, $T(x) = O(f(x))$ and $T(x) = \Omega(f(x))$.

We mentioned earlier that 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, the neighbourhood that we define depends on the PMM sub–problem. In other words, we define the following family of neighbourhoods:

$$\mathcal{N}_{\delta,\rho,\lambda,\zeta} := \left\{ (x, y, z) : \begin{array}{l} x \geq 0 \\ z \geq 0 \end{array}, \begin{array}{l} Ax + \delta(y - \lambda) = b + \frac{x^T z}{\mu_0 n} \tilde{b} \\ -Qx + A^T y + z - \rho(x - \zeta) = c + \frac{x^T z}{\mu_0 n} \tilde{c}, \quad x_i z_i \geq \gamma \frac{x^T z}{n} \end{array}, \begin{array}{l} \|\tilde{b}\| \leq C, \|\tilde{c}\| \leq C \end{array} \right\}, \tag{2.4}$$

where $C \geq 0$ is a large constant and $\gamma \in (0, 1)$ a constant preventing component–wise complementarity products from approaching zero faster than $\mu = \frac{x^T z}{n}$. The terms \tilde{b} and \tilde{c} represent the current infeasibility and will vary at every iteration. The dependence of these vectors on the iteration of the algorithm, k , is omitted for simplicity of exposition. The only requirement for \tilde{b} , \tilde{c} , is that their respective norms do not exceed C . We are now able to derive Algorithm [IP–PMM](#), summarizing the proposed interior point–proximal method of multipliers. We will prove polynomial complexity of this scheme in the next section, under some mild assumptions (to be specified later).

Notice, in Algorithm [IP–PMM](#), that we force σ 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

Algorithm IP–PMM Interior Point–Proximal Method of Multipliers

Input: A, Q, b, c, tol .

Parameters: $0 < \sigma_{\min} \leq \sigma_{\max} \in (0, 0.5]$ (bounds for the centering parameter).

Initial point: Choose $\delta_0 > 0, \rho_0 > 0, \lambda_0 \in \mathbb{R}^m (\|\lambda_0\| = O(C))$ and $\zeta_0 \in \mathbb{R}_+^n (\|\zeta_0\| = O(C))$.

Find a triple $w_0 = (x_0^T, y_0^T, z_0^T)^T \in \mathcal{N}_{\delta_0, \rho_0, \lambda_0, \zeta_0}$ with $x_0, z_0 > 0$, and set $\mu_0 = \frac{x_0^T z_0}{n}$, $k = 0$.

for ($k = 0, 1, 2, \dots$) **do**

if ($(\|Ax_k - b\| < \text{tol}) \wedge (\|c + Qx_k - A^T y_k - z_k\| < \text{tol}) \wedge (\mu_k < \text{tol})$) **then**

Declare convergence and return the optimal solution.

return (x_k, y_k, z_k) .

else

Choose $\sigma_k \in [\sigma_{\min}, \sigma_{\max}]$ and solve (Newton step):

$$\begin{bmatrix} -(Q + \rho_k I_n) & A^T & I \\ A & \delta_k I_m & 0 \\ Z & 0 & X \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = - \begin{bmatrix} -c - Qx_k + A^T y_k + z_k - \rho_k(x_k - \zeta_k) \\ Ax_k + \delta_k(y_k - \lambda_k) - b \\ XZe - \sigma_k \mu_k e \end{bmatrix}. \quad (2.5)$$

Choose step-length α_k , as the largest $\alpha \in (0, 1]$ such that $\mu_k(\alpha) \leq (1 - 0.01\alpha)\mu_k$ and:

$$(x_k(\alpha), y_k(\alpha), z_k(\alpha)) \in \mathcal{N}_{\delta_k, \rho_k, \lambda_k, \zeta_k}.$$

$$(x_{k+1}, y_{k+1}, z_{k+1}) = (x_k + \alpha_k \Delta x_k, y_k + \alpha_k \Delta y_k, z_k + \alpha_k \Delta z_k), \mu_{k+1} = \frac{x_{k+1}^T z_{k+1}}{n}.$$

Choose $\delta_{k+1} = \Theta(\mu_{k+1})$, $\rho_{k+1} = \Theta(\mu_{k+1})$, so that:

$$(x_{k+1}, y_{k+1}, z_{k+1}) \in \mathcal{N}_{\delta_{k+1}, \rho_{k+1}, \lambda_k, \zeta_k}.$$

if ($\|Ax_{k+1} - b\| \leq C \frac{\mu_{k+1}}{\mu_0}$) **then**

$$\lambda_{k+1} = y_{k+1}.$$

else

$$\lambda_{k+1} = \lambda_k.$$

end if

if ($\|c + Qx_{k+1} - A^T y_{k+1} - z_{k+1}\| \leq C \frac{\mu_{k+1}}{\mu_0}$) **then**

$$\zeta_{k+1} = x_{k+1}.$$

else

$$\zeta_{k+1} = \zeta_k.$$

end if

$$k = k + 1.$$

end if

end for

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 σ . The parameter tol , represents the error tolerance (chosen by the user). The terminating conditions, require the 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 λ, ζ are not updated if primal or dual feasibility (respectively) are not sufficiently decreased. In this case, we keep the estimates constant while continuing decreasing the penalty parameters δ, ρ . This is the typical approach employed in augmented Lagrangian methods, and is very useful in proving polynomial complexity of the algorithm. Following the usual practice with proximal and augmented Lagrangian methods, we accept a new estimate when the respective residual is sufficiently decreased. Finally, for reasons to become clear later, we force the regularization parameters δ and ρ to be decreased at the same rate as μ . This is a key feature of the approach, that will prove very useful in the convergence analysis.

3 Convergence Analysis of IP–PMM

For the rest of this section, we will make use of the following feasibility assumption.

Assumption 1. *The constraint matrix has full row rank, that is $\text{rank}(A) = m$ and there exists an optimal solution (x^*, y^*, z^*) for the primal-dual pair (P)–(D), such that: $\|x^*\| \leq C$, $\|y^*\| \leq C$ and $\|z^*\| \leq C$, for some large constant $C \geq 0$, independent of n .*

Note that this assumption is commonly employed when deriving polynomial complexity for an interior point method. Even in the case where C does depend polynomially on n , the polynomial complexity still holds. In other words, the independence on n is mainly employed for simplicity of exposition. In any case, in the next section, this assumption will be relaxed, in order to assess the behaviour of the method under infeasibility.

One can observe, for every positive parameters δ, ρ and estimates λ, ζ , that there exists a unique optimal triple $(x_r^*, y_r^*, z_r^*) \in \mathcal{S}_{\delta, \rho, \lambda, \zeta}$. To see this, notice that convexity of (P) implies that the functions $\mathcal{L}(\cdot, y, z)$ and $-\mathcal{L}(x, \cdot, \cdot)$ cannot decrease, along any direction, faster than linearly, where $\mathcal{L}(x, y, z)$ is defined in (1.1). As a consequence, the functions $\mathcal{L}(\cdot, y, z) + \frac{\rho}{2}\|\cdot - \zeta\|^2$ and $-\mathcal{L}(x, \cdot, \cdot) + \frac{\delta}{2}\|\cdot - \lambda\|^2$, must have bounded level sets. For a rigorous proof of this, the reader is referred to Section 3.4 of [8]. Hence, we can search for the solution of the following problem:

$$\arg \min_{x \geq 0} \max_{z \geq 0, y} \left\{ \mathcal{L}(x, y, z) + \frac{\rho}{2}\|x - \zeta\|^2 - \frac{\delta}{2}\|y - \lambda\|^2 \right\},$$

in a compact subset of $\mathbb{R}^{2n} \times \mathbb{R}^m$. Using Weierstrass' extreme value theorem, we know that the solution is attained. Uniqueness follows from the strong convexity of the terms $\frac{\rho}{2}\|x - \zeta\|^2$ and $\frac{\delta}{2}\|y - \lambda\|^2$. A similar result is shown in [29]. Let us now use these remarks, to derive the following important Lemma.

Lemma 3.1. *Given Assumption 1, and for all $\delta > 0, \rho > 0$ and $\lambda \in \mathbb{R}^m, \zeta \in \mathbb{R}^n$ produced by Algorithm IP–PMM, there exists a triple $(x_r^*, y_r^*, z_r^*) \in \mathcal{S}_{\delta, \rho, \lambda, \zeta}$ such that:*

$$\|x_r^*\| = O(C), \quad \|y_r^*\| = O(C), \quad \|z_r^*\| = O(C),$$

where C is a constant independent of n .

Proof. We prove the claim, by induction, firstly for x_r^* . For that, we observe that during an arbitrary iteration, say k , of Algorithm IP-PMM, the estimate ζ either remains the same, or takes the value of x_k . We know from the algorithm that $\|\zeta_0\| = O(C)$. If, during the first k iterations of the method, ζ remains constant ($\zeta_k = \zeta_0$), then we know that for any such iterate, there exists a unique triple $(x_r^*, y_r^*, z_r^*) \in \mathcal{S}_{\delta_k, \rho_k, \lambda_k, \zeta_0}$. Following the developments in [30]-Theorem 1, we can observe that under Assumption 1, the following inequality holds (which is based on the non-expansiveness of the proximal operator):

$$\|x_r^* - x^*\| \leq \|\zeta_0 - x^*\|$$

where x^* is defined in Assumption 1. A simple manipulation yields that $\|x_r^*\| = O(C)$. On the other hand, let us assume that at iteration k , we have $\|\zeta_k\| = O(C)$ and that the estimate will be updated. We know, from Algorithm (IP-PMM), that the estimate ζ is updated only if the following holds:

$$\|c + Qx_{k+1} - A^T y_{k+1} - z_{k+1}\| \leq C \frac{\mu_{k+1}}{\mu_0}.$$

On the other hand, from the neighbourhood conditions, we have that:

$$\|c + Qx_{k+1} - A^T y_{k+1} - z_{k+1} + \rho_{k+1}(x_{k+1} - \zeta_k)\| \leq C \frac{\mu_{k+1}}{\mu_0}.$$

Combining the previous two inequalities gives:

$$\rho_{k+1}\|x_{k+1} - \zeta_k\| = O(C\mu_{k+1})$$

But since $\rho_{k+1} = \Theta(\mu_{k+1})$ and $\|\zeta_k\| = O(C)$, we can conclude that $\|x_{k+1}\| = O(C)$. Hence, $\zeta_{k+1} = x_{k+1}$ and $\|\zeta_{k+1}\| = O(C)$. Again, we know that there exists a unique optimal solution $(x_r^*, y_r^*, z_r^*) \in \mathcal{S}_{\delta_{k+1}, \rho_{k+1}, \lambda_{k+1}, \zeta_{k+1}}$. Therefore, as before, non-expansiveness of the proximal operator gives the result, that is:

$$\|x_r^* - x^*\| \leq \|\zeta_{k+1} - x^*\| \Rightarrow \|x_r^*\| = O(C).$$

This completes the proof of boundedness of x_r^* , $\forall k \geq 0$. We work similarly to prove boundedness of y_r^* . As before, we observe that the initial estimate λ_0 satisfies: $\|\lambda_0\| = O(C)$. At each iteration, k , of the method, we have that either $\lambda_k = \lambda_{k-1}$ or $\lambda_k = y_k$. Assume that λ_0 stays constant for the first k iterations. Then, using the non-expansiveness of the proximal operator, we know that:

$$\|y_r^* - y^*\| \leq \|\lambda_0 - y^*\|,$$

where y^* is defined in Assumption 1. One can easily observe that as long as λ_0 is not updated, we have: $\|y_r^*\| = O(C)$. Assume now, that at iteration k , the estimate is such that $\|\lambda_k\| = O(C)$ and that $\lambda_{k+1} = y_{k+1}$. We know, from Algorithm IP-PMM, that the latter will only happen if:

$$\|Ax_{k+1} - b\| \leq C \frac{\mu_{k+1}}{\mu_0}.$$

On the other hand, from the neighbourhood conditions we know that:

$$\|Ax_{k+1} + \delta_{k+1}(y_{k+1} - \lambda_k) - b\| \leq C \frac{\mu_{k+1}}{\mu_0}.$$

Combining the previous two inequalities, yields that:

$$\delta_{k+1}\|y_{k+1} - \lambda_k\| = O(C\mu_{k+1}).$$

Since $\delta_{k+1} = \Theta(\mu_{k+1})$, and $\lambda_k = O(C)$, we can conclude that: $\|y_{k+1}\| = O(C)$. We set $\lambda_{k+1} = y_{k+1}$, and by using the non-expansiveness of the proximal operator, we get:

$$\|y_r^* - y^*\| \leq \|\lambda_{k+1} - y^*\|,$$

which in turn yields that $\|y_r^*\| = O(C)$, at iteration $k + 1$. Having proven boundedness of x_r^* and y_r^* , for every iteration of the method, we can use the dual feasibility condition, that is:

$$c + Qx_r^* - A^T y_r^* - z_r^* + \rho_k(x_r^* - \zeta_k) = 0,$$

to conclude that $\|z_r^*\| = O(C)$, $\forall k \geq 0$, which concludes the proof. \square

For simplicity of exposition, we employ the following Assumption:

Assumption 2. *For all $\delta_k, \rho_k, \lambda_k$ and ζ_k produced at iteration $k \geq 0$ of Algorithm IP-PMM, there exists a triple $(\tilde{x}, \tilde{y}, \tilde{z})$, which satisfies the following system of linear equations:*

$$\begin{aligned} A\tilde{x} + \delta_k\tilde{y} &= b + \delta_k\lambda_k + \tilde{b} \\ -Q\tilde{x} + A^T\tilde{y} + \tilde{z} - \rho_k\tilde{x} &= c - \rho_k\zeta_k + \tilde{c}, \end{aligned} \quad (3.1)$$

with $(\tilde{x}, \tilde{z}) > (0, 0)$, $\|\tilde{x}\| = O(C)$ and $\|\tilde{z}\| = O(C)$, where \tilde{b} and \tilde{c} are defined in $\mathcal{N}_{\delta_k, \rho_k, \lambda_k, \zeta_k}$.

In order to support the previous assumption, we provide Lemma 3.2, in which we prove the existence of a bounded solution for system (3.1), with $(\tilde{x}, \tilde{z}) > (0, 0)$.

Lemma 3.2. *Given Assumption 1 and for all $\delta_k, \rho_k, \lambda_k$ and ζ_k produced at iteration $k \geq 0$ of Algorithm IP-PMM, there exists a triple $(\tilde{x}, \tilde{y}, \tilde{z})$ which satisfies the following system of linear equations:*

$$\begin{aligned} A\tilde{x} + \delta_k\tilde{y} &= b + \delta_k\lambda_k + \tilde{b} \\ -Q\tilde{x} + A^T\tilde{y} + \tilde{z} - \rho_k\tilde{x} &= c - \rho_k\zeta_k + \tilde{c}, \end{aligned}$$

with $(\tilde{x}, \tilde{z}) > (0, 0)$.

Proof. Let $\hat{b} = b + \delta_k\lambda_k + \tilde{b}$ and $\hat{c} = c - \rho_k\zeta_k + \tilde{c}$. From the previous developments, as well as from the definition of $\mathcal{N}_{\delta_k, \rho_k, \lambda_k, \zeta_k}$, we know that \hat{b} and \hat{c} are bounded. In other words, there is a large constant $K \geq 0$, such that: $\|\hat{b}\| \leq K$ and $\|\hat{c}\| \leq K$, $\forall k \geq 0$. We want to find a solution for the following system:

$$\begin{aligned} A\tilde{x} + \delta_k\tilde{y} &= \hat{b} \\ -Q\tilde{x} + A^T\tilde{y} + \tilde{z} - \rho_k\tilde{x} &= \hat{c}. \end{aligned}$$

Since there is no restriction on the sign of \tilde{y} , we can set:

$$\tilde{y} = \frac{1}{\delta_k}(\hat{b} - A\tilde{x}),$$

for any choice of \tilde{x} . Substituting \tilde{y} in the second block equation of the previous system, gives:

$$-Q\tilde{x} - \frac{1}{\delta_k}A^T A\tilde{x} - \rho_k\tilde{x} + \tilde{z} = \hat{c} - \frac{1}{\delta_k}A^T\hat{b}. \quad (3.2)$$

Let $M = (Q + \frac{1}{\delta_k} A^T A + \rho_k I)$ and observe that $M \succ 0$, since $\rho_k > 0$. Hence, all eigenvalues of M , ψ_i , $i \in \{1, \dots, n\}$, are real and positive. Notice that $\|\hat{c} - \frac{1}{\delta_k} A^T \hat{b}\|$ depends on δ_k and will have a large magnitude if δ_k is close to zero. However, as δ_k decreases, the positive eigenvalues of matrix $\frac{1}{\delta_k} A^T A$ increase (notice that $\text{rank}(A) = m$). Hence, we can find positive scalars a_1, \dots, a_n , such that, if u_1, \dots, u_n denote the eigenvectors of M corresponding to eigenvalues ψ_1, \dots, ψ_n , then:

$$\sum_{i=1}^n \psi_i a_i u_i \odot u_i > \|\hat{c} - \frac{1}{\delta_k} A^T \hat{b}\|_{\infty} e,$$

where $u_i \odot u_i$ denotes component-wise multiplication of the vector u_i with itself, while e denotes the vector of ones of dimension n . Given the previous scalars a_1, \dots, a_n , we can set:

$$\tilde{x} = \sum_{i=1}^n a_i u_i \odot u_i > 0.$$

By substituting \tilde{x} in equation (3.2), we get:

$$\begin{aligned} \tilde{z} &= M\tilde{x} + \hat{c} - \frac{1}{\delta_k} A^T \hat{b} \\ &= M \sum_{i=1}^n a_i u_i \odot u_i + \hat{c} - \frac{1}{\delta_k} A^T \hat{b} \\ &= \sum_{i=1}^n a_i \psi_i u_i \odot u_i + \hat{c} - \frac{1}{\delta_k} A^T \hat{b} \\ &> \|\hat{c} - \frac{1}{\delta_k} A^T \hat{b}\|_{\infty} e + \hat{c} - \frac{1}{\delta_k} A^T \hat{b} \geq 0, \end{aligned}$$

where we used the fact that u_i denotes the i -th eigenvector of M . The last two inequalities follow by construction. This proves that there exists a solution to system (3.1) such that $(\tilde{x}, \tilde{z}) > (0, 0)$, and completes the proof. \square

In view of Lemma 3.2, one can observe that Assumption 2 is very mild and is mainly employed for simplicity of exposition. In the following Lemma, we derive boundedness of the iterates of Algorithm IP-PMM.

Lemma 3.3. *Given Assumptions 1 and 2, and that Algorithm IP-PMM produces iterates $(x_k, y_k, z_k) \in \mathcal{N}_{\delta_k, \rho_k, \lambda_k, \zeta_k}$, $\forall k \geq 0$, we have that:*

$$\|(x_k, z_k)\|_1 = O(nC^2)$$

and y_k is bounded by some large constant independent of the iteration k .

Proof. From Assumption 2, we know that, at each iteration k , there exists a triple $(\tilde{x}, \tilde{y}, \tilde{z})$ solving system (3.1), such that: $\|\tilde{x}\| = O(C)$, $\|\tilde{z}\| = O(C)$ and $(\tilde{x}, \tilde{z}) > \theta(e, e)$, for some positive constant θ . On the other hand, from Lemma 3.1, we know that at every iteration k , there exists a triple $(x_r^*, y_r^*, z_r^*) \in \mathcal{S}_{\delta_k, \rho_k, \lambda_k, \zeta_k}$, such that: $\|x_r^*\| = O(C)$ and $\|z_r^*\| = O(C)$. Let us consider the following auxiliary triple:

$$\left(\left(1 - \frac{\mu_k}{\mu_0}\right) x_r^* + \frac{\mu_k}{\mu_0} \tilde{x} - x_k, \left(1 - \frac{\mu_k}{\mu_0}\right) y_r^* + \frac{\mu_k}{\mu_0} \tilde{y} - y_k, \left(1 - \frac{\mu_k}{\mu_0}\right) z_r^* + \frac{\mu_k}{\mu_0} \tilde{z} - z_k \right). \quad (3.3)$$

Using (3.3), one can observe that:

$$\begin{aligned}
& A\left(\left(1 - \frac{\mu_k}{\mu_0}\right)x_r^* + \frac{\mu_k}{\mu_0}\tilde{x} - x_k\right) + \delta_k\left(\left(1 - \frac{\mu_k}{\mu_0}\right)y_r^* + \frac{\mu_k}{\mu_0}\tilde{y} - y_k\right) = \\
& \left(1 - \frac{\mu_k}{\mu_0}\right)(Ax_r^* + \delta_k y_r^*) + \frac{\mu_k}{\mu_0}(A\tilde{x} + \delta_k \tilde{y}) - Ax_k - \delta_k y_k = \\
& \left(1 - \frac{\mu_k}{\mu_0}\right)(b + \delta_k \lambda_k) + \frac{\mu_k}{\mu_0}(b + \delta_k \lambda_k + \tilde{b}) - Ax_k - \delta_k y_k = \\
& \qquad \qquad \qquad b + \delta_k \lambda_k + \frac{\mu_k}{\mu_0}\tilde{b} - Ax_k - \delta_k y_k = 0,
\end{aligned}$$

where the last equality follows from the definition of the neighbourhood $\mathcal{N}_{\delta_k, \rho_k, \lambda_k, \zeta_k}$. Similarly, one can show that:

$$-(Q + \rho_k I)\left(\left(1 - \frac{\mu_k}{\mu_0}\right)x_r^* + \frac{\mu_k}{\mu_0}\tilde{x} - x_k\right) + A^T\left(\left(1 - \frac{\mu_k}{\mu_0}\right)y_r^* + \frac{\mu_k}{\mu_0}\tilde{y} - y_k\right) + \left(\left(1 - \frac{\mu_k}{\mu_0}\right)z_r^* + \frac{\mu_k}{\mu_0}\tilde{z} - z_k\right) = 0.$$

By combining the previous two relations, we have:

$$\begin{aligned}
& \left(\left(1 - \frac{\mu_k}{\mu_0}\right)x_r^* + \frac{\mu_k}{\mu_0}\tilde{x} - x_k\right)^T \left(\left(1 - \frac{\mu_k}{\mu_0}\right)z_r^* + \frac{\mu_k}{\mu_0}\tilde{z} - z_k\right) = \\
& \left(\left(1 - \frac{\mu_k}{\mu_0}\right)x_r^* + \frac{\mu_k}{\mu_0}\tilde{x} - x_k\right)^T (Q + \rho_k I)\left(\left(1 - \frac{\mu_k}{\mu_0}\right)x_r^* + \frac{\mu_k}{\mu_0}\tilde{x} - x_k\right) + \tag{3.4} \\
& \delta_k \left(\left(1 - \frac{\mu_k}{\mu_0}\right)y_r^* + \frac{\mu_k}{\mu_0}\tilde{y} - y_k\right)^T \left(\left(1 - \frac{\mu_k}{\mu_0}\right)y_r^* + \frac{\mu_k}{\mu_0}\tilde{y} - y_k\right) \geq 0.
\end{aligned}$$

From (3.4), it can be seen that:

$$\begin{aligned}
& \left(\left(1 - \frac{\mu_k}{\mu_0}\right)x_r^* + \frac{\mu_k}{\mu_0}\tilde{x}\right)^T z_k + \left(\left(1 - \frac{\mu_k}{\mu_0}\right)z_r^* + \frac{\mu_k}{\mu_0}\tilde{z}\right)^T x_k \leq \\
& \left(\left(1 - \frac{\mu_k}{\mu_0}\right)x_r^* + \frac{\mu_k}{\mu_0}\tilde{x}\right)^T \left(\left(1 - \frac{\mu_k}{\mu_0}\right)z_r^* + \frac{\mu_k}{\mu_0}\tilde{z}\right) + x_k^T z_k.
\end{aligned}$$

However, from Assumption 2 and Lemma 3.1, we have that: $\tilde{x} \geq \theta e$, $\tilde{z} \geq \theta e$, for some positive constant $\theta = O(1)$, while $\|x_r^*\| = O(C)$, $\|z_r^*\| = O(C)$, $\|\tilde{x}\| = O(C)$ and $\|\tilde{z}\| = O(C)$. By combining all the previous, we get:

$$\begin{aligned}
& \frac{\mu_k}{\mu_0}\theta(e^T x_k + e^T z_k) \leq \\
& \left(\left(1 - \frac{\mu_k}{\mu_0}\right)x_r^* + \frac{\mu_k}{\mu_0}\tilde{x}\right)^T z_k + \left(\left(1 - \frac{\mu_k}{\mu_0}\right)z_r^* + \frac{\mu_k}{\mu_0}\tilde{z}\right)^T x_k \leq \\
& \left(\left(1 - \frac{\mu_k}{\mu_0}\right)x_r^* + \frac{\mu_k}{\mu_0}\tilde{x}\right)^T \left(\left(1 - \frac{\mu_k}{\mu_0}\right)z_r^* + \frac{\mu_k}{\mu_0}\tilde{z}\right) + x_k^T z_k = \tag{3.5} \\
& \frac{\mu_k}{\mu_0}\left(1 - \frac{\mu_k}{\mu_0}\right)(x_r^*)^T \tilde{z} + \frac{\mu_k}{\mu_0}\left(1 - \frac{\mu_k}{\mu_0}\right)\tilde{x}^T z_r^* + \left(\frac{\mu_k}{\mu_0}\right)^2 \tilde{x}^T \tilde{z} + x_k^T z_k = \\
& \qquad \qquad \qquad O\left(\frac{\mu_k}{\mu_0}C^2\right) + O\left(\left(\frac{\mu_k}{\mu_0}\right)^2 C^2\right) + O(\mu_k n),
\end{aligned}$$

where we used the fact that $(x_r^*)^T z_r^* = 0$. Hence, (3.5) implies that:

$$\|(x_k, z_k)\|_1 = O(nC^2).$$

Finally, we have from the neighbourhood conditions that:

$$c + Qx_k - A^T y_k - z_k + \rho_k(x_k - \zeta_k) + \frac{\mu_k}{\mu_0} \tilde{b} = 0.$$

Boundedness of all terms (except for y_k), as well as the fact that $\text{rank}(A) = m$, yields boundedness of y_k , and completes the proof. \square

As in a typical IPM convergence analysis, we proceed by bounding some components of the scaled Newton direction. The proof is similar to that presented in [33], Lemma 6.5.

Lemma 3.4. *Given Assumptions 1 and 2, there exists a constant $K = O(nC^3)$, independent of the iteration k , such that:*

$$\|D^{-1}\Delta x_k\| \leq K\mu^{\frac{1}{2}}, \quad \|D\Delta z_k\| \leq K\mu^{\frac{1}{2}},$$

with $D^2 = XZ^{-1}$, for every iteration k of Algorithm IP-PMM, where $(\Delta x_k, \Delta y_k, \Delta z_k)$ is the Newton direction obtained by solving system (2.5).

Proof. Let us define the following matrix: $W = (XZ)^{\frac{1}{2}}$. At each iteration of Algorithm IP-PMM, consider the following auxiliary triple:

$$(\bar{x}, \bar{y}, \bar{z}) = (\Delta x_k, \Delta y_k, \Delta z_k) + \frac{\mu_k}{\mu_0}(\tilde{x}, \tilde{y}, \tilde{z}) - \frac{\mu_k}{\mu_0}(x_r^*, y_r^*, z_r^*), \quad (3.6)$$

where $(\tilde{x}, \tilde{y}, \tilde{z})$ is defined in Assumption 2, and (x_r^*, y_r^*, z_r^*) is defined in Lemma 3.1. Using the neighbourhood conditions in (2.4), one can observe that the auxiliary triple in (3.6) solves the following homogeneous system:

$$\begin{aligned} A\bar{x} + \delta_k \bar{y} &= 0, \\ -Q\bar{x} + A^T \bar{y} + \bar{z} - \rho_k \bar{x} &= 0. \end{aligned}$$

The previous implies that:

$$\bar{x}^T \bar{z} = \bar{x}^T (Q\bar{x} - A^T \bar{y} + \rho_k \bar{x}) = \bar{x}^T (Q + \rho_k I) \bar{x} + \delta_k \bar{y}^T \bar{y} \geq 0. \quad (3.7)$$

On the other hand, using the last block equation of the Newton system (2.5), we have:

$$Z\bar{x} + X\bar{z} = -XZe + \sigma_k \mu_k e + Z \frac{\mu_k}{\mu_0} (\tilde{x} - x_r^*) + X \frac{\mu_k}{\mu_0} (\tilde{z} - z_r^*).$$

By multiplying both sides of the previous equation by W^{-1} , we get:

$$D^{-1}\bar{x} + D\bar{z} = -W^{-1}(XZe - \sigma_k \mu_k e) + \frac{\mu_k}{\mu_0} D^{-1}(\tilde{x} - x_r^*) + \frac{\mu_k}{\mu_0} D(\tilde{z} - z_r^*). \quad (3.8)$$

But, from (3.7), we know that $\bar{x}^T \bar{z} \geq 0$ and hence:

$$\|D^{-1}\bar{x} + D\bar{z}\|^2 \geq \|D^{-1}\bar{x}\|^2 + \|D\bar{z}\|^2.$$

Combining (3.8) with the previous, gives:

$$\|D^{-1}\bar{x}\|^2 + \|D\bar{z}\|^2 \leq \left\{ \|W^{-1}\| \|XZe - \sigma_k \mu_k e\| + \frac{\mu_k}{\mu_0} \|D^{-1}(\tilde{x} - x_r^*)\| + \frac{\mu_k}{\mu_0} \|D(\tilde{z} - z_r^*)\| \right\}^2.$$

We isolate one of the two terms of the left hand side of the previous inequality, take square roots on both sides and apply the triangular inequality to it, to get:

$$\|D^{-1}\Delta x_k\| \leq \|W^{-1}\| \|XZe - \sigma_k \mu_k e\| + 2\frac{\mu_k}{\mu_0} \|D^{-1}(\tilde{x} - x_r^*)\| + \frac{\mu_k}{\mu_0} \|D(\tilde{z} - z_r^*)\|. \quad (3.9)$$

We now proceed to bounding the terms in the right hand side of (3.9). Firstly, notice from the neighbourhood conditions (2.4), that: $\gamma\mu_k \leq x_k^i z_k^i$. This in turn implies that:

$$\|W^{-1}\| = \max_i \frac{1}{(x_k^i z_k^i)^{\frac{1}{2}}} \leq \frac{1}{(\gamma\mu_k)^{\frac{1}{2}}}.$$

On the other hand, we have that:

$$\begin{aligned} \|XZe - \sigma_k \mu_k e\|^2 &= \|XZe\|^2 - 2\sigma_k \mu_k x_k^T z_k + \sigma_k^2 \mu_k^2 n \\ &\leq \|XZe\|_1^2 - 2\sigma_k \mu_k x_k^T z_k + \sigma_k^2 \mu_k^2 n \\ &= (\mu_k n)^2 - 2\sigma_k \mu_k^2 n + \sigma_k \mu_k^2 n \\ &\leq \mu_k^2 n^2. \end{aligned}$$

Hence, combining the previous two relations yields:

$$\|W^{-1}\| \|XZe - \sigma_k \mu_k e\| \leq \frac{n}{\gamma^{\frac{1}{2}}} \mu_k^{\frac{1}{2}} = O(n\mu^{\frac{1}{2}}).$$

We proceed by bounding $\|D^{-1}\|$. For that, we have:

$$\|D^{-1}\| = \max_i \|D_{ii}^{-1}\| = \|D^{-1}e\|_\infty = \|W^{-1}Ze\|_\infty \leq \|W^{-1}\| \|z\|_1 = O\left(\frac{nC^2}{(\gamma\mu_k)^{\frac{1}{2}}}\right).$$

Similarly, we have that:

$$\|D\| = O\left(\frac{nC^2}{(\gamma\mu_k)^{\frac{1}{2}}}\right).$$

Hence:

$$\begin{aligned} 2\frac{\mu_k}{\mu_0} \|D^{-1}(\tilde{x} - x_r^*)\| + \frac{\mu_k}{\mu_0} \|D(\tilde{z} - z_r^*)\| &\leq 2\frac{\mu_k}{\mu_0} (\|D^{-1}\| + \|D\|) \max\{\|\tilde{x} - x_r^*\|, \|\tilde{z} - z_r^*\|\} \\ &= O(nC^3\mu_k^{\frac{1}{2}}). \end{aligned}$$

Combining all the previous bounds yields the claimed bound for $\|D^{-1}\Delta x_k\|$. One can bound $\|D\Delta z_k\|$ in the same way and this completes the proof. \square

We are now able to prove that at every iteration of Algorithm [IP-PMM](#), there exists a step-length $\alpha_k > 0$, using which, the new iterate satisfies the conditions required by the algorithm.

Lemma 3.5. *Given Assumptions 1 and 2, there exists a step-length $\bar{\alpha} \in (0, 1)$, such that: $\forall \alpha \in [0, \bar{\alpha}]$ and $\forall k \geq 0$, the following relations hold:*

$$(x_k + \alpha\Delta x_k)^T (z_k + \alpha\Delta z_k) \geq (1 - \alpha(1 - \beta_1)) x_k^T z_k, \quad (3.10)$$

$$(x_k^i + \alpha\Delta x_k^i)(z_k^i + \alpha\Delta z_k^i) \geq \frac{\gamma}{n} (x_k + \alpha\Delta x_k)^T (z_k + \alpha\Delta z_k), \quad (3.11)$$

$$(x_k + \alpha \Delta x_k)^T (z_k + \alpha \Delta z_k) \leq (1 - \alpha(1 - \beta_2)) x_k^T z_k, \quad (3.12)$$

where, without loss of generality, $\beta_1 = \frac{\sigma_{\min}}{2}$ and $\beta_2 = 0.99$. Moreover, $\bar{\alpha} \geq \frac{\bar{\kappa}}{n^2}$, for some constant $\bar{\kappa} > 0$ independent of n .

Proof. The proof follows the developments in Chapter 6, Lemma 6.7 in [33], but we provide it here for completeness. From Lemma 3.4, we have:

$$(\Delta x_k)^T \Delta z_k = (D^{-1} \Delta x_k)^T (D \Delta z_k) \leq \|D^{-1} \Delta x_k\| \|D \Delta z_k\| \leq K^2 \mu_k,$$

where $K = O(nC^3)$. Similarly, it is easy to see that:

$$\|\Delta x_k^i \Delta z_k^i\| \leq K^2 \mu_k.$$

On the other hand, by summing over all n components of the last block equation of the Newton system (2.5), we have:

$$z_k^T \Delta x_k + x_k^T \Delta z_k = e^T (Z \Delta x_k + X \Delta z_k) = e^T (-X Z e + \sigma_k \mu_k e) = (\sigma_k - 1) x_k^T z_k, \quad (3.13)$$

or component-wise:

$$z_k^i \Delta x_k^i + x_k^i \Delta z_k^i = -x_k^i z_k^i + \sigma_k \mu_k. \quad (3.14)$$

We proceed by proving (3.10). Using (3.13), we have:

$$\begin{aligned} & (x_k + \alpha \Delta x_k)^T (z_k + \alpha \Delta z_k) - (1 - \alpha(1 - \beta_1)) x_k^T z_k = \\ & x_k^T z_k + \alpha(\sigma_k - 1) x_k^T z_k + \alpha^2 \Delta x_k^T \Delta z_k - (1 - \alpha) x_k^T z_k - \alpha \beta_1 x_k^T z_k \geq \\ & \alpha(\sigma_k - \beta_1) x_k^T z_k - \alpha^2 K^2 \mu_k \geq \alpha \left(\frac{\sigma_{\min}}{2} \right) n \mu_k - \alpha^2 K^2 \mu_k, \end{aligned}$$

where we set (without loss of generality) $\beta_1 = \frac{\sigma_{\min}}{2}$. The most-right hand side of the previous inequality will be non-negative for every α satisfying:

$$\alpha \leq \frac{n \sigma_{\min}}{2K^2}.$$

In order to prove (3.11), we will use (3.14) and the fact that from the neighbourhood conditions we have: $x_k^i z_k^i \geq \gamma \mu_k$. In particular, we have:

$$\begin{aligned} (x_k^i + \alpha \Delta x_k^i)(z_k^i + \alpha \Delta z_k^i) & \geq (1 - \alpha) x_k^i z_k^i + \alpha \sigma_k \mu_k - \alpha^2 K^2 \mu_k \\ & \geq \gamma(1 - \alpha) \mu_k + \alpha \sigma_k \mu_k - \alpha^2 K^2 \mu_k. \end{aligned}$$

By combining all the previous, we get:

$$\begin{aligned} (x_k^i + \alpha \Delta x_k^i)(z_k^i + \alpha \Delta z_k^i) - \frac{\gamma}{n} (x_k + \alpha \Delta x_k)^T (z_k + \alpha \Delta z_k) & \geq \\ \alpha \sigma_k (1 - \gamma) \mu_k - \left(1 + \frac{\gamma}{n}\right) \alpha^2 K^2 \mu_k & \geq \\ \alpha \sigma_{\min} (1 - \gamma) \mu_k - 2\alpha^2 K^2 \mu_k. \end{aligned}$$

In turn, the most-right hand side of the previous relation will be non-negative for every α satisfying:

$$\alpha \leq \frac{\sigma_{\min}(1 - \gamma)}{2K^2}.$$

Finally, to prove (3.12), we set (without loss of generality) $\beta_2 = 0.99$. We know, from Algorithm IP-PMM, that $\sigma_{\max} \leq 0.5$. With the previous two remarks in mind, we have:

$$\begin{aligned} & \frac{1}{n}(x_k + \alpha\Delta x_k)^T(z_k + \alpha\Delta z_k) - (1 - 0.01\alpha)\mu_k \leq \\ & (1 - \alpha)\mu_k + \alpha\sigma_k\mu_k + \alpha^2\frac{K^2}{n}\mu_k - (1 - 0.01\alpha)\mu_k \leq \\ & -0.99\alpha\mu_k + 0.5\alpha\mu_k + \alpha^2\frac{K^2}{n}\mu_k = \\ & -0.49\alpha\mu_k + \alpha^2\frac{K^2}{n}\mu_k. \end{aligned}$$

The last term will be non-positive for every α satisfying:

$$\alpha \leq \frac{0.49n}{K^2}.$$

Hence, by combining all the bounds on the step-length, we have that:

$$\bar{\alpha} = \min \left\{ \frac{n\sigma_{\min}}{2K^2}, \frac{\sigma_{\min}(1 - \gamma)}{2K^2}, \frac{0.49n}{K^2}, 1 \right\}. \quad (3.15)$$

Further noticing, from Lemma 3.4, that $K = O(nC^3)$, where C is a positive constant independent of n , completes the proof. \square

In the following Lemma, we show that given an initial point in the parametrized neighbourhood (2.4), the iterates of the algorithm will lie in their respective parametrized neighbourhoods.

Lemma 3.6. *Given Assumptions 1 and 2, and any iterate $(x_k, y_k, z_k) \in \mathcal{N}_{\delta_k, \rho_k, \lambda_k, \zeta_k}$, Algorithm IP-PMM produces an iterate $(x_{k+1}, y_{k+1}, z_{k+1}) \in \mathcal{N}_{\delta_{k+1}, \rho_{k+1}, \lambda_{k+1}, \zeta_{k+1}}$.*

Proof. In Lemma 3.5, we showed that (3.11) holds, which in turn is one of the neighbourhood conditions. Hence, we have to prove that the new iterate will satisfy the approximate feasibility conditions of the new neighbourhood. We have that: $(x_k, y_k, z_k) \in \mathcal{N}_{\delta_k, \rho_k, \lambda_k, \zeta_k}$. From Algorithm IP-PMM, we know that if:

$$\|Ax_{k+1} - b\| \leq C\frac{\mu_{k+1}}{\mu_0}$$

then, $\lambda_{k+1} = y_{k+1}$. In that case, we have that:

$$\|Ax_{k+1} + \delta_{k+1}(y_{k+1} - \lambda_{k+1}) - b\| = \|Ax_{k+1} - b\| \leq C\frac{\mu_{k+1}}{\mu_0},$$

which implies that $\exists \tilde{b} : \|\tilde{b}\| \leq C$ such that: $Ax_{k+1} = b + \frac{\mu_{k+1}}{\mu_0}\tilde{b}$. Let us assume now that: $\|Ax_{k+1} - b\| > C\frac{\mu_{k+1}}{\mu_0}$, which implies that $\lambda_{k+1} = \lambda_k$. From the Newton system (2.5), we have that:

$$Ax_{k+1} + \delta_k(y_{k+1} - \lambda_{k+1}) - b = (1 - \alpha)(Ax_k + \delta_k(y_k - \lambda_k) - b).$$

At this point, we want to update δ_k such that $\delta_{k+1} = \Theta(\mu_{k+1})$. Let us assume that: $\delta_{k+1} = (1 - \alpha_k\psi)\delta_k$. Then, we would like to show that there exists $\psi > 0$, independent of the iteration k , such that:

$$\|Ax_{k+1} + (1 - \alpha_k\psi)\delta_k(y_{k+1} - \lambda_{k+1}) - b\| \leq C\frac{\mu_{k+1}}{\mu_0}. \quad (3.16)$$

With this in mind, we see that:

$$\begin{aligned} & \|Ax_{k+1} + (1 - \alpha_k\psi)\delta_k(y_{k+1} - \lambda_{k+1}) - b\| \leq \\ & \|Ax_{k+1} + \delta_k(y_{k+1} - \lambda_{k+1}) - b\| + \alpha_k\psi\delta_k\|y_{k+1} - \lambda_{k+1}\| \leq \\ & (1 - \alpha_k)C\frac{\mu_k}{\mu_0} + \alpha_k\psi\delta_k\|y_{k+1} - \lambda_k\| \end{aligned}$$

From (3.10), we know that: $\mu_{k+1} \geq (1 - \alpha_k(1 - \frac{\sigma_{\min}}{2}))\mu_k$. Hence, given the previous and (3.10), we know that (3.16) will hold for every $\psi > 0$, such that:

$$\psi\delta_k\|y_{k+1} - \lambda_k\| \leq \frac{\sigma_{\min}}{2}C\frac{\mu_k}{\mu_0}$$

or equivalently:

$$\psi \leq \bar{\psi} = \min \left\{ \frac{\sigma_{\min}C}{2\mu_0\|y_{k+1} - \lambda_k\|} \frac{\mu_k}{\delta_k}, 1 \right\}.$$

This implies that (3.16) holds, for every $\psi \in (0, \bar{\psi})$. However, from Algorithm IP-PMM, we know that $\delta_k = \Theta(\mu_k)$, while, from Lemma 3.3, we know that y_{k+1} must be bounded by a constant independent of the iteration k (and hence λ_k is also bounded). In other words, $\bar{\psi}$ must be bounded away from zero, independently of the iteration.

Similarly, if:

$$\| -Qx_{k+1} + A^T y_{k+1} + z_{k+1} - c \| \leq C\frac{\mu_{k+1}}{\mu_0},$$

then there exists \tilde{c} such that: $\|\tilde{c}\| \leq C$ and $-Qx_{k+1} + A^T y_{k+1} + z_{k+1} = c + \frac{\mu_{k+1}}{\mu_0}\tilde{c}$. On the other hand, if the previous does not hold, as before, we can show that there must exist a constant $\psi > 0$ independent of the iteration k , such that:

$$\|Qx_{k+1} - A^T y_{k+1} - z_{k+1} + (1 - \alpha_k\psi)\rho_k(x_{k+1} - \zeta_{k+1}) + c\| \leq C\frac{\mu_{k+1}}{\mu_0}. \quad (3.17)$$

We omit the details, since the latter can be treated similarly to the approximate primal feasibility condition. \square

The following Theorem summarizes our results.

Theorem 3.7. *The sequence $\{\mu_k\}$ generated by Algorithm IP-PMM converges Q -linearly to zero, and the sequences of regularized residual norms*

$$\{\|Ax_k + \delta_k(y_k - \lambda_k) - b\|\} \text{ and } \{\| -Qx_k + A^T y_k + z_k - \rho_k(x_k - \zeta_k) - c\|\}$$

converge R -linearly to zero.

Proof. From (3.12) we have that:

$$\mu_{k+1} \leq (1 - 0.01\alpha_k)\mu_k,$$

while, from (3.15), we know that $\forall k \geq 0, \exists \bar{\alpha} \geq \frac{\bar{\kappa}}{n^2}$ 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 $k \geq 0$:

$$\|Ax_k + \delta_k(y_k - \lambda_k) - b\| \leq C\frac{\mu_k}{\mu_0}$$

and

$$\| -Qx_k + A^T y_k + z_k - \rho_k(x_k - \zeta_k) - c \| \leq C \frac{\mu_k}{\mu_0}.$$

This completes the proof. \square

Theorem 3.8. *Let $\epsilon \in (0, 1)$ be a given error tolerance. Suppose also, that a suitable starting point for Algorithm IP-PMM is available, such that $\mu_0 \leq \frac{C}{\epsilon^\omega}$ for some positive constants C, ω . Given Assumptions 1 and 2, there exists an index K with:*

$$K = O(n^2 |\log \epsilon|)$$

such that the iterates $\{w_k\} = \{(x_k^T, y_k^T, z_k^T)^T\}$ generated from Algorithm IP-PMM satisfy:

$$\mu_k \leq \epsilon, \quad \forall k \geq K.$$

Proof. We omit the proof, which follows standard developments. Such a proof can be found in [33, 34], among others. \square

Theorem 3.9. *Suppose that Algorithm IP-PMM generates the iterates $(x_k, y_k, z_k) \in \mathcal{N}_{\delta_k, \rho_k, \lambda_k, \zeta_k}$ and terminates when a limit point is reached. Then, if Assumptions 1 and 2 hold, every limit point of $\{(x_k, y_k, z_k)\}$, determines a primal-dual solution of the non-regularized pair (P)-(D).*

Proof. From Theorem 3.7, we know that $\{\mu_k\} \rightarrow 0$, and hence, there exists a sub-sequence $\mathcal{K} \subseteq \mathbb{N}$, such that:

$$\{Ax_k + \delta_k(y_k - \lambda_k) - b\}_{\mathcal{K}} \rightarrow 0, \quad \{-Qx_k + A^T y_k + z_k - \rho_k(x_k - \zeta_k) - c\}_{\mathcal{K}} \rightarrow 0.$$

Now, from Algorithm IP-PMM we have that $\delta_k = \Theta(\mu_k)$, $\rho_k = \Theta(\mu_k)$ and from Lemma 3.3 we know that $\{(x_k, y_k, z_k)\}$ is a bounded sequence. Hence, we obtain that:

$$\{Ax_k - b\}_{\mathcal{K}} \rightarrow 0, \quad \{-Qx_k + A^T y_k + z_k - c\}_{\mathcal{K}} \rightarrow 0.$$

One can readily observe that, under our assumptions, the limit point of the algorithm satisfies the conditions given in (1.2). \square

Note that a complexity result for the case where $\text{rank}(A) < m$ (rank deficient case) could be possible, however, it would substantially complicate the analysis and hence it is left for a future work. Nevertheless, global convergence for this case could be derived, by following the developments in [12]. For ease of presentation, we do not present it here.

4 Infeasible Problems

Let us now drop our previous assumptions (Assumption 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, the sequences: $\{\|y_k - \lambda_k\|\}$ and $\{\|x_k - \zeta_k\|\}$, $\forall k \geq 0$, remain bounded.*

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

The following analysis is based on the developments in [12] and [2]. 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 our problem is infeasible), then we will show that Premise 1 cannot hold.

Lemma 4.1. *Given Premise 1, and by assuming that $x_k^T z_k > \epsilon$, for some $\epsilon > 0$, for all iterations k of Algorithm IP-PMM, the Newton direction produced by (2.5) is uniformly bounded by a constant dependent only on n .*

Proof. Let us use a variation of Theorem 1 given in [2]. This theorem states that if the following conditions are satisfied,

- (i) Q, A are bounded matrices,
- (ii) $\delta > 0$,
- (iii) $\exists \bar{\epsilon} > 0 : x_i z_i \geq \bar{\epsilon}, \forall i = \{1, 2, \dots, n\}, \forall k \geq 0$,
- (iv) and the matrix $H = \rho I + Q + X^{-1}Z + \frac{1}{\delta}A^T A$ is positive definite,

then the coefficient matrix of (2.5) is non-singular and has a uniformly bounded inverse. Note that (i), (ii) and (iv) are trivial to verify. Condition (iii) follows since we know that our iterates lie on $\mathcal{N}_{\delta_k, \rho_k, \lambda_k, \zeta_k}$, while we have $x^T z \geq \epsilon$, by assumption. Indeed, from the neighbourhood conditions (2.4), we have that $x_k^i z_k^i \geq \frac{\gamma}{n} x_k^T z_k$. Hence, there exists $\bar{\epsilon} = \frac{\gamma \epsilon}{n} > 0$ such that $x_k^i z_k^i \geq \bar{\epsilon}, \forall k \geq 0, \forall i = \{1, \dots, n\}$.

Finally, we have to show that the right hand side of (2.5) is uniformly bounded. But that follows immediately from the neighbourhood conditions (2.4), and completes the proof. \square

In the following Lemma, given Premise 1, we prove by contradiction that the parameter μ_k of Algorithm IP-PMM converges to zero. The proof is based on the developments in [19, 12] and is only partially given here, for ease of presentation.

Lemma 4.2. *Given Premise 1, and a sequence $(x_k, y_k, z_k) \in \mathcal{N}_{\delta_k, \rho_k, \lambda_k, \zeta_k}$ produced by Algorithm IP-PMM, the sequence $\{\mu_k\}$ converges to zero.*

Proof. Assume, by virtue of contradiction, that $\mu_k > \epsilon, \forall k \geq 0$. Then, we know that the Newton direction obtained by the algorithm at every iteration, after solving (2.5), will be uniformly bounded by a constant dependent only on n . Take any $k \geq 0$ and define the following functions:

$$\begin{aligned} f_i(\alpha) &= (x_k^i + \alpha \Delta x_k^i)(z_k^i + \alpha \Delta z_k^i) - \gamma \mu_k(\alpha), \quad i = 1, \dots, n, \\ h(\alpha) &= (1 - 0.01\alpha)x_k^T z_k - (x_k(\alpha))^T (z_k(\alpha)), \\ g_p(\alpha) &= \frac{\mu_k(\alpha)C}{\mu_0} - \|Ax_k(\alpha) + \delta_{k+1}(y_k(\alpha) - \lambda_{k+1}) - b\|, \\ g_d(\alpha) &= \frac{\mu_k(\alpha)C}{\mu_0} - \|c + Qx_k(\alpha) - A^T y_k(\alpha) - z_k(\alpha) + \rho_{k+1}(x_k(\alpha) - \zeta_{k+1})\|, \end{aligned}$$

where $\mu_k(\alpha) = \frac{(x_k + \alpha \Delta x_k)^T (z_k + \alpha \Delta z_k)}{n}$, $(x_k(\alpha), y_k(\alpha), z_k(\alpha)) = (x_k + \alpha \Delta x_k, y_k + \alpha \Delta y_k, z_k + \alpha \Delta z_k)$, $\delta_{k+1} = (1 - \alpha \psi_1) \delta_k$ and $\rho_{k+1} = (1 - \alpha \psi_2) \rho_k$, with ψ_1 and ψ_2 determined in a similar manner to Lemma 3.6 (see (3.16) and (3.17) respectively). We would like to show that there exists $\alpha^* > 0$, such that:

$$f_i(\alpha^*) \geq 0, \quad h(\alpha^*) \geq 0, \quad g_p(\alpha^*) \geq 0, \quad g_d(\alpha^*) \geq 0.$$

These conditions model the requirement that the next iteration of Algorithm IP-PMM must lie in the updated neighbourhood: $\mathcal{N}_{\delta_{k+1}, \rho_{k+1}, \lambda_{k+1}, \zeta_{k+1}}$. Proving the existence of $\alpha^* > 0$ such that $f_i(\alpha^*) \geq 0, \forall i = 1, \dots, n$ and $h(\alpha^*) \geq 0$, follows exactly the developments in Section 3 of [19]. We prove the claim for $g_p(\cdot)$ and observe that a similar analysis is needed for $g_d(\cdot)$. We will treat the case where $\lambda_{k+1} = \lambda_k$ here, since proving the claim for the case where $\lambda_{k+1} = y_{k+1}$ follows easily. Using Lemma 4.1, we know that there exists a constant π , depending only on n , such that:

$$\Delta x_k^T \Delta z_k \leq \pi.$$

Then, from the second and the third block equations of (2.5), by taking the triangular inequality, and by using the neighbourhood conditions, we have that:

$$\begin{aligned} g_p(\alpha) &\geq \frac{C}{n\mu_0} ((1 - \alpha)x_k^T z_k + \alpha \sigma_k x_k^T z_k + \alpha^2 \Delta x_k^T \Delta z_k) \\ &\quad - (1 - \alpha) \|Ax_k + \delta_k(y_k - \lambda_k) - b\| - \alpha \psi_1 \delta_k \|y_{k+1} - \lambda_k\| \\ &\geq \frac{C}{n\mu_0} ((1 - \alpha)x_k^T z_k + \alpha \sigma_{\min} x_k^T z_k) \\ &\quad - (1 - \alpha) \frac{C x_k^T z_k}{n\mu_0} - \alpha \psi_1 \delta_k \|y_{k+1} - \lambda_k\| - \alpha^2 \frac{\pi C}{n\mu_0} \\ &= \alpha \left(\frac{\sigma_{\min} C}{\mu_0} \mu_k - \psi_1 \delta_k \|y_{k+1} - \lambda_k\| \right) - \alpha^2 \frac{\pi C}{n\mu_0}. \end{aligned}$$

Notice however, using a similar reasoning as in Lemma 3.6, that:

$$\psi_1 \leq \min \left\{ \frac{\sigma_{\min} C}{2\mu_0 \|y_{k+1} - \lambda_k\|} \frac{\mu_k}{\delta_k}, 1 \right\},$$

and from Premise 1 as well as from Algorithm IP-PMM, we know that there always exists such a ψ_1 , bounded away from zero. In other words, we have that:

$$g_p(\alpha) \geq \alpha \epsilon^* - \alpha^2 \frac{\pi C}{n\mu_0},$$

where $\epsilon^* \geq \frac{\sigma_{\min} C}{2\mu_0}$, and hence, $g_p(\alpha) \geq 0, \forall \alpha \in [0, \frac{\epsilon^* n \mu_0}{\pi C}]$. Working similarly, we can show that all the functions are non-negative, for every $\alpha \in [0, \alpha^*]$, where:

$$\alpha^* = \min \left\{ 1, \frac{(1 - \gamma) \sigma_{\min} \epsilon}{\pi}, \frac{(0.99 - \sigma_{\max}) \epsilon}{\pi}, \frac{\epsilon^* n \mu_0}{\pi C} \right\}. \quad (4.1)$$

However, using the inequality:

$$\mu_{k+1} \leq (1 - 0.01\alpha) \mu_k, \quad \forall \alpha \in [0, \alpha^*]$$

we get that $\mu_k \rightarrow 0$, which contradicts our assumption that $\mu_k > \epsilon, \forall k \geq 0$. \square

Finally, using the following Theorem, we will derive a condition for detecting infeasible problems.

Theorem 4.3. *Given Premise 2, i.e. there does not exist a KKT triple for the pair (P)–(D), then Premise 1 fails to hold.*

Proof. By virtue of contradiction, let Premise 1 hold. In Lemma 4.2, we proved that given Premise 1, Algorithm IP–PMM produces iterates that belong to the neighbourhood (2.4) and $\mu_k \rightarrow 0$. But from the neighbourhood conditions we can observe that:

$$\|Ax_k + \delta_k(y_k - \lambda_k) - b\| \leq C \frac{\mu_k}{\mu_0},$$

and

$$\| -Qx_k + A^T y_k + z_k - \rho_k(x_k - \zeta_k) - c \| \leq C \frac{\mu_k}{\mu_0}.$$

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

$$\{Ax_k + \delta_k(y_k - \lambda_k) - b\}_{\mathcal{K}} \rightarrow 0, \text{ and } \{c + Qx_k - A^T y_k - z_k + \rho_k(x_k - \zeta_k)\}_{\mathcal{K}} \rightarrow 0.$$

But since $\|y_k - \lambda_k\|$ and $\|x_k - \zeta_k\|$ are bounded, while $\delta_k \rightarrow 0$ and $\rho_k \rightarrow 0$, we have that:

$$\{Ax_k - b\}_{\mathcal{K}} \rightarrow 0, \text{ and } \{c + Qx_k - A^T y_k - z_k\}_{\mathcal{K}} \rightarrow 0.$$

This contradicts Premise 2, i.e. that the pair (P)–(D) does not have a KKT triple, and completes the proof. \square

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 can be found here:

<https://www.maths.ed.ac.uk/ERGO/software.html> ([source link](#)).

5.1 Implementation Details

Our implementation slightly deviates from the theory, in order to gain some additional computational efficiency.

5.1.1 Free Variables

The method takes as input problems in the following form:

$$\min_x \left(c^T x + \frac{1}{2} x^T Q x \right), \text{ s.t. } Ax = b, x^I \geq 0, x^F \text{ free,}$$

where $I = \{1, \dots, n\} \setminus F$ 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 [29]. In general, convex 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.

5.1.2 Constraint Matrix Scaling

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

$$\left(\max_{i \in \{1, \dots, m\}, j \in \{1, \dots, n\}} (|A_{ij}|) < 10 \right) \wedge \left(\min_{i \in \{1, \dots, m\}, j \in \{1, \dots, n\}: |A_{ij}| > 0} (|A_{ij}|) > 0.1 \right).$$

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:

$$d_i = \frac{1}{\sqrt{\max_{j \in \{1, \dots, n\}} (|A(i, :)|) \cdot \min_{j \in \{1, \dots, n\}: |A_{ij}| > 0} (|A(i, :)|)}}, \quad \forall i \in \{1, \dots, m\}$$

However, for numerical stability, we find the largest integer p_i , such that: $2^{p_i} \leq d_i$ and we set $d_i = 2^{p_i}$, $\forall i \in \{1, \dots, m\}$. Hence, the scaling factors will be 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.

5.1.3 Starting Point, Newton-step Computation and Step-length

We use a starting point, based on the developments in [22]. To construct it, we try to solve the pair of problems (P)–(D), ignoring the non-negativity constraints.

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

However, in order to ensure stability and efficiency, we regularize the matrix AA^T and we employ the Preconditioned Conjugate Gradient (PCG) method to solve these systems (in order to avoid forming AA^T). We use the classical Jacobi preconditioner to accelerate PCG, i.e. $P = \text{diag}(AA^T) + \delta I$, where $\delta = 8$, is set as the regularization parameter.

Then, in order to guarantee positivity and sufficient magnitude of x_I, z_I , we compute $\delta_x = \max\{-1.5 \cdot \min(\tilde{x}^I), 0\}$ and $\delta_z = \max\{-1.5 \cdot \min(\tilde{z}^I), 0\}$ and we obtain:

$$\begin{aligned} \tilde{\delta}_x &= \delta_x + 0.5 \frac{(\tilde{x}^I + \delta_x e^{|I|})^T (\tilde{z}^I + \delta_z e^{|I|})}{\sum_{i=1}^{|I|} (\tilde{z}^{I(i)} + \delta_z)}, \\ \tilde{\delta}_z &= \delta_z + 0.5 \frac{(\tilde{x}^I + \delta_x e^{|I|})^T (\tilde{z}^I + \delta_z e^{|I|})}{\sum_{i=1}^{|I|} (\tilde{x}^{I(i)} + \delta_x)}, \end{aligned}$$

where $e^{|I|}$ is the vector of ones of appropriate dimension. Finally, we set:

$$y_0 = \tilde{y}, \quad z_0^I = \tilde{z}^I + \tilde{\delta}_z e^{|I|}, \quad z_0^F = 0, \quad x_0^I = \tilde{x}^I + \tilde{\delta}_x e^{|I|}, \quad x_0^F = \tilde{x}^F.$$

In order to find the Newton step, we employ a widely used predictor-corrector method. We provide the algorithmic scheme in Algorithm PC for completeness, but the reader is referred to [22], for an extensive review of the method.

We solve the systems (5.1) and (5.2), using the build in Matlab symmetric decomposition (i.e. `ldl`). Such a decomposition always exists, with D diagonal, for the aforementioned systems, since after introducing the regularization, the system matrices are guaranteed to be quasi-definite; a class of matrices known to be strongly factorizable, [31]. In order to exploit that, we

Algorithm PC Predictor–Corrector method

Compute the predictor:

$$\begin{bmatrix} -(Q + \Theta^{-1} + \rho_k I) & A^T \\ A & \delta_k I \end{bmatrix} \begin{bmatrix} \Delta_p x \\ \Delta_p y \end{bmatrix} = \begin{bmatrix} c + Qx_k - A^T y_k - \rho_k(x_k - \zeta_k) - d_1 \\ b - Ax_k - \delta_k(y_k - \lambda_k) \end{bmatrix}, \quad (5.1)$$

where $d_1^I = -\mu_k(X^I)^{-1}e_{|I|}$ and $d_1^F = 0$.

Retrieve $\Delta_p z$:

$$\Delta_p z^I = d_1^I - (X^I)^{-1}(z^I \odot \Delta_p x^I), \quad \Delta_p z^F = 0.$$

where $a \odot b$ denotes component–wise multiplication between the two vectors.

Compute the step in the non–negativity orthant:

$$\alpha_x^{\max} = \min_{(\Delta_p x^{I(i)} < 0)} \left\{ 1, -\frac{x^{I(i)}}{\Delta_p x^{I(i)}} \right\}, \quad \alpha_z^{\max} = \min_{(\Delta_p z^{I(i)} < 0)} \left\{ 1, -\frac{z_k^{I(i)}}{\Delta_p z^{I(i)}} \right\},$$

for $i = 1, \dots, |I|$, and set:

$$\alpha_x = \tau \alpha_x^{\max}, \quad \alpha_z = \tau \alpha_z^{\max},$$

with $\tau = 0.995$ (avoid going too close to the boundary).

Compute a centrality measure:

$$g_\alpha = (x^I + \alpha_x \Delta_p x^I)^T (z^I + \alpha_z \Delta_p z^I).$$

Set: $\mu = \left(\frac{g_\alpha}{(x_k^I)^T z_k^I} \right)^2 \frac{g_\alpha}{|I|}$

Compute the corrector:

$$\begin{bmatrix} -(Q + \Theta^{-1} + \rho_k I) & A^T \\ A & \delta_k I \end{bmatrix} \begin{bmatrix} \Delta_c x \\ \Delta_c y \end{bmatrix} = \begin{bmatrix} d_2 \\ 0 \end{bmatrix}, \quad (5.2)$$

with $d_2^I = \mu(X^I)^{-1}e^{|I|} - (X^I)^{-1}\Delta_p x^I \odot \Delta_p z^I$ and $d_2^F = 0$.

Retrieve $\Delta_c z$:

$$\Delta_c z^I = d_2^I - (X^I)^{-1}(z^I \odot \Delta_c x^I), \quad \Delta_c z^F = 0.$$

$$(\Delta x, \Delta y, \Delta z) = (\Delta_p x + \Delta_c x, \Delta_p y + \Delta_c y, \Delta_p z + \Delta_c z).$$

Compute the step in the non–negativity orthant:

$$\alpha_x^{\max} = \min_{\Delta x^{I(i)} < 0} \left\{ 1, -\frac{x^{I(i)}}{\Delta x^{I(i)}} \right\}, \quad \alpha_z^{\max} = \min_{\Delta z^{I(i)} < 0} \left\{ 1, -\frac{z^{I(i)}}{\Delta z^{I(i)}} \right\},$$

and set:

$$\alpha_x = \tau \alpha_x^{\max}, \quad \alpha_z = \tau \alpha_z^{\max}.$$

Update:

$$(x_{k+1}, y_{k+1}, z_{k+1}) = (x_k + \alpha_x \Delta x, y_k + \alpha_z \Delta y, z_k + \alpha_z \Delta z).$$

change the default pivot threshold of `ld1` to a value slightly lower than the minimum allowed regularization value (`regthr`; specified in sub-section 5.1.4). Using such a small pivot threshold guarantees that no 2x2 pivots will be employed during the factorization routine.

5.1.4 Regularization Parameters

At the beginning of the optimization, we set: $\delta_0 = 8$, $\rho_0 = 8$, $\lambda_0 = y_0$, $\zeta_0 = x_0$. 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 δ or ρ to become smaller than a suitable positive value, `regthr`. We set: $\text{reg}_{thr} = \max\left\{\frac{\text{tol}}{\max\{\|A\|_\infty^2, \|Q\|_\infty^2\}}, 10^{-10}\right\}$. This value is based on the developments in [27], in order to ensure that we introduce a controlled perturbation in the eigenvalues of the non-regularized linear system. The reader is referred to [27] for an extensive study on the subject. If the factorization fails, we increase the regularization parameters by a factor of 10 and repeat the factorization. If this happens while either δ or ρ have reached their minimum allowed value (`regthr`), 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.

Algorithm `PEU` Penalty and Estimate Updates

```

 $r = \frac{|\mu_k - \mu_{k+1}|}{\mu_k}$  (rate of decrease of  $\mu$ ).
if ( $\|Ax_{k+1} - b\| \leq 0.95 \cdot \|Ax_k - b\|$ ) then
     $\lambda_{k+1} = y_{k+1}$ .
     $\delta_{k+1} = (1 - r) \cdot \delta_k$ .
else
     $\lambda_{k+1} = \lambda_k$ .
     $\delta_{k+1} = (1 - \frac{1}{3}r) \cdot \delta_k$ .
end if
 $\delta_{k+1} = \max\{\delta_{k+1}, \text{reg}_{thr}\}$ , for numerical stability (ensure quasi-definiteness).
if ( $\|c + Qx_{k+1} - A^T y_{k+1} - z_{k+1}\| \leq 0.95 \cdot \|c + Qx_k - A^T y_k - z_k\|$ ) then
     $\zeta_{k+1} = x_{k+1}$ .
     $\rho_{k+1} = (1 - r) \cdot \rho_k$ .
else
     $\zeta_{k+1} = \zeta_k$ .
     $\rho_{k+1} = (1 - \frac{1}{3}r) \cdot \rho_k$ .
end if
 $\rho_{k+1} = \max\{\rho_{k+1}, \text{reg}_{thr}\}$ .
 $k = k + 1$ .

```

5.1.5 Termination Criteria

There are three possible termination criteria.

1. The method declares convergence, when all the following three conditions are satisfied:

$$\frac{\|c - A^T y + Qx - z\|}{\max\{\|c\|, 1\}} \leq \text{tol},$$

$$\frac{\|b - Ax\|}{\max\{\|b\|, 1\}} \leq \text{tol},$$

and

$$\mu \leq \text{tol},$$

where tol is the tolerance specified by the user.

2. Primal–dual infeasibility is declared, if the primal or the dual estimate has not been updated for 5 consecutive iterations and either of the following conditions holds:

$$(\|c + Qx_k - A^T y_k - z_k + \rho_k(x_k - \zeta_k)\| \leq \text{tol}) \wedge (\|x_k - \zeta_k\| > 10^{10}),$$

or

$$(\|b - Ax_k - \delta_k(y_k - \lambda_k)\| \leq \text{tol}) \wedge (\|y_k - \lambda_k\| > 10^{10}).$$

3. Finally, the method is terminated if the number of maximum iterations, specified by the user, is reached before any of the previous termination criteria is met.

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 IP–PMM, simply by fixing δ and ρ 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 will show 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 2.2GHz Intel Core i7 processor (hexa–core), 16GB RAM, run under Windows 10 operating system. The Matlab version used was R2018b.

5.2.1 Linear Programming Problems

Let us compare the proposed method with the respective non–regularized implementation, over the Netlib collection, [25]. The test set consists of 96 linear programming problems. We set the desired tolerance to $\text{tol} = 10^{-6}$. 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 143 seconds (and a total of 3,015 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 non-regularized 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 182 seconds (and a total of 3,035 iterations). We notice here, that IP-PMM converges slower when solving the pre-solved problems. This is most likely the case due to the extra restrictions that are introduced from the pre-solve. Nevertheless, two more benefits of regularization become obvious here. Firstly, we can observe that numerical stability can be a problem, even if we ensure that the constraint matrices are of full row rank. Secondly, notice that despite the fact that IP-PMM required 61% more iterations, it still solved the whole set in 47% 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.

In Table 1, we collect statistics from the runs of the two methods over some medium scale instances of the presolved Netlib test set. Notice that the number of constraint/variables presented, occurs after adding slack variables and extra constraints needed to transform the problem to the accepted IP-PMM format.

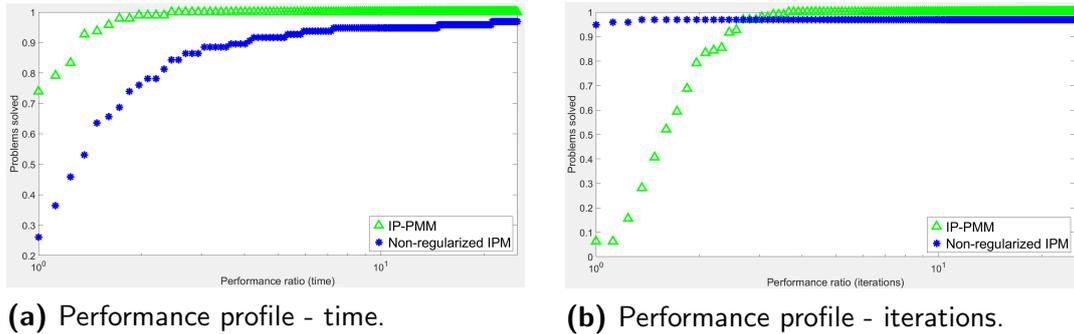
Table 1: Netlib medium scale instances

Name	m	n	IP-PMM		Non-regularized IPM	
			Iter.	t_{total} (s)	Iter.	t_{total} (s)
80BAU3B	8,466	16,156	62	1.98	40	9.68
D6CUBE	5,846	10,886	29	3.43	22	9.64
PILOT87	5,037	9,077	54	6.61	46	95.04
QAP15	27,916	44,493	44	91.67	18	93.56

From Table 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 1, which contains the performance profiles, over the pre-solved Netlib 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 1a, we present the performance profile with respect to time required for convergence, while in Figure 1b, the performance profile with respect to the number of iterations. In both figures, the horizontal axis is in logarithmic scale, and it represents the ratio with respect to the best performance achieved by one of the two methods, for each problem. The vertical axis shows the percentage of problems solved by each method, for different values of the performance ratio. Robustness is “measured” by the maximum achievable percentage, while efficiency by the rate of increase of each of the lines (faster increase translates to better efficiency). For more information about performance profiles, we refer the reader to [10], where this benchmarking was proposed. One can see that all of our previous observations are verified in Figure 1.

Figure 1: Performance profiles over the pre-solved Netlib test set.



5.2.2 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 ([25], 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 4 problems, for which the method terminated after reaching the maximum number of iterations. For 1 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 recognized 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 [4, 24] and the references therein, for various other infeasibility detection methods.

5.2.3 Quadratic Programming Problems

Next, we present the comparison of the two methods over the Maros–Mészáros test set ([21]), 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 the paper is not in 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 2,756 iterations). As before, the required tolerance was set to 10^{-6} .

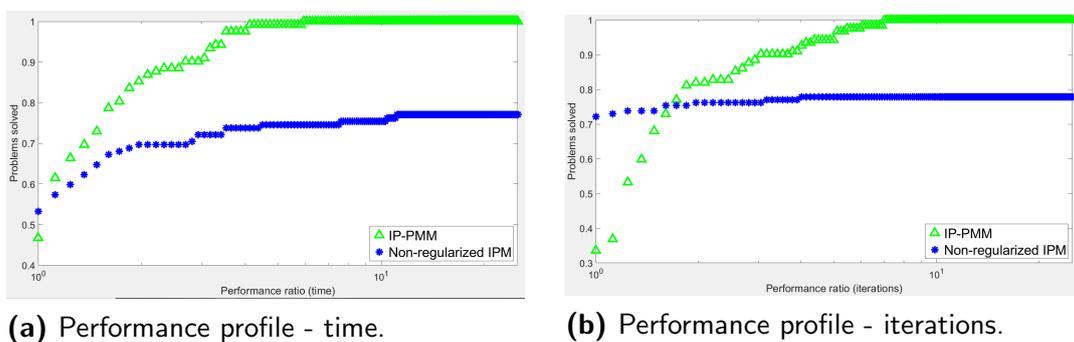
In Table 2, we collect statistics from the runs of the two methods over some medium scale instances of the Maros–Mészáros collection. Again, the number of constraints/variables presented, occurs after adding slack variables and extra constraints needed to transform the problem in the accepted IP-PMM format.

Table 2: Maros-Mészáros medium scale instances

Name	m	n	IP-PMM			Non-regularized IPM		
			Iter.	t_{total} (s)	Stat.	Iter.	t_{total} (s)	Stat.
CVXQP1.1	15,000	20,000	47	39.06	opt	16	293.90	opt
CVXQP3.1	17,500	20,000	74	80.55	opt	200	-	n-opt
LISWET1	10,000	20,002	25	0.38	opt	34	1.43	opt
STCQP1	6,149	8,194	16	0.50	opt	12	5.79	opt

In order to summarize the comparison of the two methods, we include Figure 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 2a, we present the performance profile with respect to time required for convergence, while in Figure 2b, the performance profile with respect to the number of iterations.

Figure 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).

5.2.4 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 [6], among others). In what follows, we provide a “table of robustness” of IP-PMM. 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: 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, 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%.

5.2.5 Large Scale Problems

All of our previous experiments were conducted on small to medium scale linear and convex quadratic programming problems. We have showed (both theoretically and practically) that the proposed method is reliable. However, it is worth mentioning the limitations of the method. Since we employ exact factorizations 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 4 we provide the statistics of the runs of the method over a small set of large scale problems. It contains the dimensions of the problems (after transforming them in the IP-PMM format), the number of non-zeros, as well as the time needed to convergence. The tolerance used in these experiments was 10^{-6} .

Table 4: Large scale problems

Name	Collection	m	n	nnz	time (s)	Stat.
fome21	Mittelmann	137,469	286,071	751,365	567.26	opt
pds-10	Mittelmann	32,706	66,080	250,307	40.00	opt
pds-30	Mittelmann	103,456	212,001	564,988	447.81	opt
pds-60	Mittelmann	222,285	459,275	1,320,986	2,265.39	opt
pds-100	Mittelmann	365,890	724,224	1,953,757	-	no memory
rail582	Mittelmann	582	56097	402,290	91.10	opt
cre-b	Kennington	9,648	77,137	347,742	24.48	opt
cre-d	Kennington	8,926	73,948	323,287	23.49	opt
stocfor3	Kennington	16,675	23,541	72,721	4.56	opt
ken-18	Kennington	259,826	309,398	667,569	77.94	opt
osa-30	Kennington	4,350	104,374	604,488	1723.96	opt
nug-20	QAPLIB	15,240	72,600	304,800	386.12	opt
nug-30	QAPLIB	52,260	379,350	1,567,800	-	no memory

From Table 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.

The reason for that, is that the required memory could exceed the available RAM memory, 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 iterations) seems to be a viable and competitive alternative and will be addressed in a future work.

6 Conclusions

In this paper, we present 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 reasonable assumptions. Then, an easily implementable infeasibility detection mechanism is derived and tested in practice. 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 noted that embedding iterative methods in the current scheme can significantly 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 solutions and still expect fast convergence. Hence, a future goal is to extend the theory as well as the implementation, in order to allow for inexact Newton directions.

References

- [1] A. Altman and J. Gondzio. Regularized Symmetric Indefinite Systems in Interior Point Methods for Linear and Quadratic Optimization. *Optim. Meth. and Soft.*, Vol. 11 & 12 : 275–302, 1999.
- [2] P. Armand and J. Benoist. Uniform Boundedness of the Inverse of a Jacobian Matrix Arising in Regularized Interior-Point Methods. *Math. Prog.*, Vol. 137 (No. 1&2): 587–592, 2013.
- [3] P. Armand and R. Omhenni. A Mixed Logarithmic Barrier-Augmented Lagrangian Method for Nonlinear Optimization. *J. Optim. Theory and Appl.*, Vol. 173 : 523–547, 2017.
- [4] P. Armand and N. N. Tran. Rapid Infeasibility Detection in a Mixed Logarithmic Barrier-Augmented Lagrangian Method for Nonlinear Optimization. *Optim. Meth. and Soft.* <https://doi.org/10.1080/10556788.2018.1528250>, 2018.

- [5] S. Arreckx and D. Orban. A Regularized Factorization-Free Method for Equality-Constrained Optimization. *SIAM J. Optim.*, Vol. 28 (No. 2): 1613–1639, 2018.
- [6] P. D. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Athena Scientific, 1996.
- [7] P. D. Bertsekas, A. Nedic, and E. Ozdaglar. *Convex Analysis and Optimization*. Athena Scientific, 2003.
- [8] P. D. Bertsekas and N. J. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods*. Athena Scientific, 1997.
- [9] Y. Censor and A. Zenios. Proximal Minimization Algorithm with D-Functions. *J. of Optim. Theory and Appl.*, Vol. 73 (No. 3): 451–464, 1992.
- [10] D. E. Dolan and J. J. Moré. Benchmarking Optimization Software with Performance Profiles. *Math. Prog. Ser. A.*, Vol. 91 : 201–213, 2002.
- [11] J. Eckstein. Nonlinear Proximal Point Algorithms Using Bregman Functions, with Applications to Convex Programming. *Math. of Operational Research*, Vol. 18 (No. 1), 1993.
- [12] M. P. Friedlander and D. Orban. A Primal-Dual Regularized Interior-Point Method for Convex Quadratic Programs. *Math. Prog. Comp.*, Vol. 4 (No. 1): 71–107, 2012.
- [13] O. Güler. New Proximal Point Algorithms for Convex Minimization. *SIAM J. Optim.*, Vol. 2 (No. 4): 649–664, 1992.
- [14] E. P. Gill and P. D. Robinson. A Primal-Dual Augmented Lagrangian. *Comp. Optim. Appl.*, Vol. 51 : 1–25, 2012.
- [15] J. Gondzio. Multiple Centrality Corrections in a Primal-Dual Method for Linear Programming. *Comp. Optim. and Appl.*, Vol. 6 : 137–156, 1996.
- [16] J. Gondzio. Interior Point Methods 25 Years Later. *European J. of Operational Research*, Vol. 218 (No. 3): 587–601, 2013.
- [17] M. R. Hestenes. Multiplier and Gradient Methods. *J. of Optim. Theory and Appl.*, (No. 4):303–320, 1969.
- [18] A. N. Iusem. Some Properties of Generalized Proximal Point Methods for Quadratic and Linear Programming. *J. of Optim. Theory and Appl.*, Vol. 85 (No. 3): 593–612, 1995.
- [19] M. Kojima, N. Megiddo, and S. Mizuno. A Primal-Dual Infeasible-Interior-Point Algorithm for Linear Programming. *Math. Prog.*, Vol. 61 : 263–280, 1993.
- [20] G. Lan and R. D. C. Monteiro. Iteration-Complexity of First-Order Augmented Lagrangian Methods for Convex Programming. *Math. Prog.*, Vol. 155 (No. 1-2): 511–547, 2016.
- [21] I. Maros and C. Mészáros. A Repository of Convex Quadratic Programming Problems. *Optim. Meth. and Soft.*, Vol. 11 & 12 : 671–681, 1999.

- [22] S. Mehrotra. On the Implementation of a Primal-Dual Interior-Point Method. *SIAM J. Optim.*, Vol. 2 (No. 4): 575–601, 1992.
- [23] Y. Nesterov and A. Nemirovski. *Interior-Point Polynomial Algorithms in Convex Programming*. SIAM, 1993.
- [24] Y. Nesterov, M. J. Todd, and Y. Ye. Infeasible-start primal-dual methods and infeasibility detectors for nonlinear programming problems. *Math. Prog.*, Vol. 82 (No. 2): 227–267, 1999.
- [25] Netlib. <http://netlib.org/lp>, 2011.
- [26] N. Parikh and S. Boyd. Proximal Algorithms. *Foundations and Trends in Optim.*, Vol. 3 (No. 1): 123–231, 2014.
- [27] S. Pougkakiotis and J. Gondzio. Dynamic Non-diagonal Regularization in Interior Point Methods for Linear and Convex Quadratic Programming. *J. of Optim. Theory and Appl.* <https://doi.org/10.1007/s10957-019-01491-1>, 2019.
- [28] M. J. D. Powell. A Method for Nonlinear Constraints in Minimization Problems. *Optim.*, (No. 4): 283–298, 1969.
- [29] R. T. Rockafellar. Augmented Lagrangians and Applications of the Proximal Point Algorithm in Convex Programming. *Math. of Operations Research*, Vol. 1 (No. 2): 97–116, 1976.
- [30] R. T. Rockafellar. Monotone Operators and the Proximal Point Algorithm. *SIAM J. Control and Optim.*, Vol. 14 (No. 5), 1976.
- [31] R. Vanderbei. Symmetric Quasidefinite Matrices. *SIAM J. Optim.*, Vol. 5 (No. 1): 100–113, 1995.
- [32] Y. Wang and P. Fei. An Infeasible Mizuno-Todd-Ye Type Algorithm for Convex Quadratic Programming with Polynomial Complexity. *Numer. Funct. Anal. and Optim.*, Vol. 28 (No. 3-4): 487–502, 2007.
- [33] S. J. Wright. *Primal-Dual Interior Point Methods*. SIAM, 1997.
- [34] Y. Zhang. On the Convergence of a Class of Infeasible Interior-Point Methods for the Horizontal Linear Complementarity Problem. *SIAM J. Optim.*, Vol. 4 (No. 1): 208–227, 1994.
- [35] 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. *Comp. Optim. Appl.*, Vol. 27 (No. 3): 269–283, 2004.