2019 IEEE 58th Conference on Decision and Control (CDC)
Palais des Congrès et des Expositions Nice Acropolis
Nice, France, December 11-13, 2019

QPALM: A Newton-type Proximal Augmented Lagrangian Method for Quadratic Programs

Ben Hermans,[1] Andreas Themelis[2] and Panagiotis Patrinos[2]

Abstract— **We present a proximal augmented Lagrangian based solver for general quadratic programs (QPs), relying on semismooth Newton iterations with exact line search to solve the inner subproblems. The exact line search reduces in this case to finding the zero of a one-dimensional monotone, piecewise affine function and can be carried out very efficiently. Our algorithm requires the solution of a linear system at every iteration, but as the matrix to be factorized depends on the active constraints, efficient sparse factorization updates can be employed like in active-set methods. Both primal and dual residuals can be enforced down to strict tolerances and otherwise infeasibility can be detected from intermediate iterates. A C implementation of the proposed algorithm is tested and benchmarked against other state-of-the-art QP solvers for a large variety of problem data and shown to compare favorably against these solvers.**

I. Introduction

This paper deals with convex quadratic programs (QPs)

$$\underset{x\in\mathbb{R}^n}{\text{minimize}}\ \tfrac{1}{2}\langle x, Qx\rangle + \langle q, x\rangle \quad \text{subject to}\ \ \ell \le Ax \le u, \quad \text{(QP)}$$

where $Q \in \mathbb{R}^{n\times n}$ is symmetric and positive semidefinite, $q \in \mathbb{R}^n$, $A \in \mathbb{R}^{m\times n}$, and $\ell, u \in \mathbb{R}^m$.

Efficiently and reliably solving QPs is a key challenge in optimization [15, §16]. QPs cover a wide variety of applications and problem classes, such as portfolio optimization, support vector machines, sparse regressor selection, real-time linear model predictive control (MPC), etc. QPs also often arise as subproblems in general nonlinear optimization techniques such as sequential quadratic programming. Therefore, substantial research has been performed to develop robust and efficient QP solvers. State-of-the-art solvers are typically based on interior point methods, such as MOSEK [1] and Gurobi [14], or active-set methods, such as qpOASES [11]. Both methods have their advantages and disadvantages. Interior point methods typically require few but expensive iterations, involving the solution of a linear system at every iteration. In contrast, active-set methods require more but cheaper iterations, as the linear system changes only slightly and low-rank factorization updates can be used instead of having to factorize from scratch. As a result, active-set methods can be efficiently warm started when solving a series of similar QPs, which is common in applications such as MPC, whereas interior-point methods in general do not have this capability. Recently, proximal algorithms, also known as operator splitting methods [16], have experienced a resurgence in popularity. Relying only on first-order information of the problems, such methods have as their advantage operational simplicity and cheap iterations, but they may exhibit slow asymptotic convergence for poorly scaled problems. The recently proposed OSQP solver [21], based on the alternating direction method of multipliers (ADMM), addresses this crucial issue by means of a tailored offline scaling that performs very well on some QP problems, although a more thorough benchmarking confirms the known limitations of proximal algorithms. Indeed, parameter tuning is typically set before the execution, with possibly minor online adjustments, and as such operator splitting methods do not take into account curvature information about the problem that can greatly speed up convergence.

In this work we propose a novel and reliable QP solver based on the augmented Lagrangian method (ALM) [5] and in particular proximal ALM [18], which is efficient and robust against ill conditioning. ALM involves penalizing the constraints and solving a series of unconstrained minimizations, where fast smooth optimization techniques can be employed. QPs turn out to be particularly amenable for this approach, as optimal stepsizes of exact line searches are available in closed form, similiar to what was shown in previous work [17], resulting in an extremely fast minimization strategy. In each unconstrained optimization, the iterates rely on the solution of a linear system, dependent on the set of active constraints. As such, similarly to active-set methods, our iterates can benefit from low-rank factorization update techniques and are therefore cheaper than in interior point methods. However, in contrast to active-set methods and more in the flavor of other algorithms such as the dual gradient projection method of [2], our method allows for substantial changes in the active set at each iteration and converges therefore much faster on average. To some extent, our algorithm strikes a balance between interior point and active-set methods. In regard to state-of-the-art ALM solvers, not much research is available. The authors of [13] presented an ALM solver for QPs, OQLA/QPALM, which solves the inner minimization problems using a combination of active-set and gradient projection methods. Although the authors of

[1]Ben Hermans is with the Department of Mechanical Engineering, KU Leuven, and DMMS lab, Flanders Make, Leuven, Belgium. His research benefits from KU Leuven-BOF PFV/10/002 Centre of Excellence: Optimization in Engineering (OPTEC), from project G0C4515N of the Research Foundation - Flanders (FWO - Flanders), from Flanders Make ICON: Avoidance of collisions and obstacles in narrow lanes, and from the KU Leuven Research project C14/15/067: B-spline based certificates of positivity with applications in engineering.

[2]Andreas Themelis and Panagiotis Patrinos are with the Department of Electrical Engineering (ESAT-STADIUS) – KU Leuven, Kasteelpark Arenberg 10, 3001 Leuven, Belgium. This work was supported by the *Research Foundation Flanders (FWO)* research projects G086518N and G086318N; *Research Council KU Leuven* C1 project No. C14/18/068; *Fonds de la Recherche Scientifique — FNRS and the Fonds Wetenschappelijk Onderzoek — Vlaanderen* under EOS project no 30468160 (SeLMA).
{ben.hermans2,andreas.themelis,panos.patrinos}@kuleuven.be

[9] discuss second order updates based on a generalized Hessian, similar to our method, their approach deals with convex (composite) optimization problems in general, and as such it can not make use of the efficient factorization updates and optimal stepsizes that are key to the efficiency of our algorithm. Finally, a link with proximal algorithms can also be established, in the sense that the form of an iterate of unconstrained minimization is very similar to what is obtained by applying ADMM to (QP). Differently from an ADMM approach where parameters are set before the execution, in our method penalties are adjusted online. Moreover, penalty matrices are used instead of scalars to allow for more flexibility in penalizing the constraints. As a result of the online adjustments, our algorithm requires in general fewer, although slightly more expensive, iterations.

The contribution of this paper is QPALM, a full-fledged QP solver based on proximal ALM. We describe the relevant theory and outline in detail the algorithmic steps for both the outer and inner minimization procedures. We also provide an open-source C implementation of the algorithm,[1] which we benchmark with state-of-the-art QP solvers, showing that QPALM can compete with and very often outperform them both in runtime and robustness in regard to problem scaling.

The remainder of this paper is structured as follows. Section II introduces notation used in the paper and underlying theoretical concepts. Section III outlines the application of the proximal ALM to (QP). Section IV discusses in detail the algorithmic steps, regarding both inner and outer minimization, of QPALM. Section V addresses some of the crucial implementation details that contribute to making QPALM an efficient and robust solver. Section VI presents simulation results on QPs of varying sizes and problem conditioning, benchmarking QPALM's performance against state-of-the-art solvers. Finally, Section VII draws concluding remarks and mentions future directions of research.

II. Preliminaries

We now introduce some notational conventions and briefly list some known facts needed in the paper. The interested reader is referred to [4] for an extensive discussion.

A. Notation and known facts

We denote the extended real line by $\overline{\mathbb{R}} := \mathbb{R} \cup \{\infty\}$. The scalar product on $\mathbb{R}^n$ is denoted by $\langle \cdot, \cdot \rangle$. With $[x]_+ := \mathbf{max}\{x, 0\}$ we indicate the positive part of vector $x \in \mathbb{R}^n$, meant in a componentwise sense. A sequence of vectors $(x^k)_{k\in\mathbb{N}}$ is said to be summable if $\sum_{k\in\mathbb{N}} \|x^k\| < \infty$.

With $\mathbf{Sym}(\mathbb{R}^n)$ we indicate the set of symmetric $\mathbb{R}^{n\times n}$ matrices, and with $\mathbf{Sym}_+(\mathbb{R}^n)$ and $\mathbf{Sym}_{++}(\mathbb{R}^n)$ the subset of those which are positive semidefinite and positive definite, respectively. Given $\Sigma \in \mathbf{Sym}_{++}(\mathbb{R}^n)$ we indicate with $\|\cdot\|_\Sigma$ the norm on $\mathbb{R}^n$ induced by Σ, namely $\|x\|_\Sigma := \sqrt{\langle x, \Sigma x\rangle}$.

Given a nonempty closed convex set $C \subseteq \mathbb{R}^n$, with $\mathbf{\Pi}_C(x)$ we indicate the projection of a point $x \in \mathbb{R}^n$ onto C, namely $\mathbf{\Pi}_C(x) = \mathbf{arg\,min}_{y\in C} \|y - x\|$ or, equivalently, the unique point $z \in C$ satisfying the inclusion

$$x - z \in \mathcal{N}_C(z), \tag{1}$$

where $\mathcal{N}_C(z) := \{v \in \mathbb{R}^n \mid \langle v, z - z'\rangle \le 0 \ \forall z' \in C\}$ is the normal cone of C at z. $\mathbf{dist}(x, C)$ and $\mathbf{dist}_\Sigma(x, C)$ denote the distance from x to set C in the Euclidean norm and in that induced by Σ, respectively, while δ_C is the indicator function of set C, namely $\delta_C(x) = 0$ if $x \in C$ and ∞ otherwise.

B. Convex functions and monotone operators

The *Fenchel conjugate* of a proper closed convex function $\varphi : \mathbb{R}^n \to \overline{\mathbb{R}}$ is the convex function $\varphi^* : \mathbb{R}^n \to \overline{\mathbb{R}}$ defined as $\varphi^*(y) = \mathbf{sup}_x \langle x, y\rangle - \varphi(x)$. The subdifferential of φ at $x \in \mathbb{R}^n$ is $\partial\varphi(x) := \{v \in \mathbb{R}^n \mid \varphi(x') \ge \varphi(x) + \langle v, x' - x\rangle \ \forall x' \in \mathbb{R}^n\}$. Having $y \in \partial\varphi(x)$ is equivalent to $x \in \partial\varphi^*(y)$.

A point-to-set mapping $\mathcal{M} : \mathbb{R}^n \rightrightarrows \mathbb{R}^n$ is *monotone* if $\langle x - x', \xi - \xi'\rangle \ge 0$ for all $x, x' \in \mathbb{R}^n$, $\xi \in \mathcal{M}(x)$ and $\xi' \in \mathcal{M}(x')$. It is *maximally monotone* if, additionally, there exists no monotone operator $\mathcal{M}' \ne \mathcal{M}$ such that $\mathcal{M}(x) \subseteq \mathcal{M}'(x)$ for all $x \in \mathbb{R}^n$. The *resolvent* of a maximally monotone operator $\mathcal{M}$ is the single-valued (in fact, Lipschitz-continuous) mapping $(\mathrm{id} + \mathcal{M})^{-1}$, where $(\mathrm{id} + \mathcal{M})^{-1}(x)$ is the unique point $\bar{x} \in \mathbb{R}^n$ such that $x - \bar{x} \in \mathcal{M}(\bar{x})$. $\mathbf{zer}\,\mathcal{M} := \{x \mid 0 \in \mathcal{M}(x)\}$ denotes the zero-set of $\mathcal{M}$, and for a linear mapping Σ, $\Sigma\mathcal{M}$ is the operator defined as $\Sigma M(x) := \{\Sigma y \mid y \in \mathcal{M}(x)\}$.

The subdifferential of a proper convex lower semicontinuous function φ is maximally monotone, and its resolvent is the proximal mapping $\mathbf{prox}_\varphi := (\mathrm{id} + \partial\varphi)^{-1}$.

III. Proximal ALM for QPs

This section is devoted to establishing the theoretical ground in support of the proposed Algorithm 1. We will show that our scheme amounts to proximal ALM and derive its convergence guarantees following the original analysis in [18], here generalized to account for scaling matrices (as opposed to scalars) and by including the possibility of having different Lagrangian and proximal weights. We start by observing that problem (QP) can equivalently be expressed as

$$\underset{x\in\mathbb{R}^n}{\mathbf{minimize}}\ f(x) + g(Ax), \tag{2}$$

where $f(x) := \frac{1}{2}\langle x, Qx\rangle + \langle q, x\rangle$, $g(z) := \delta_C(z)$ and $C := \{z \in \mathbb{R}^m \mid \ell \le z \le u\}$. The KKT conditions of (2) are

$$0 \in \mathcal{M}(x, y) := \begin{pmatrix} \nabla f(x) + A^\top y \\ -Ax + \partial g^*(y) \end{pmatrix}.$$

Let $\mathcal{V}_\star := \mathbf{zer}\,\mathcal{M}$ be the set of primal-dual solutions. Since $\mathcal{M}$ is maximally monotone, as first observed in [18] one can find KKT-optimal primal-dual pairs by recursively applying the resolvent of $c_k\mathcal{M}$, where $(c_k)_{k\in\mathbb{N}}$ is an increasing sequence of strictly positive scalars. This scheme is known as proximal point algorithm (PPA) [18]. We now show that these scalars can in fact be replaced by positive definite matrices.

Theorem 1. *Suppose that (2) has a solution. Starting from $(x^0, y^0) \in \mathbb{R}^n \times \mathbb{R}^m$, let $(x^k, y^k)_{k\in\mathbb{N}}$ be recursively defined as*

$$(x^{k+1}, y^{k+1}) = (\mathrm{id} + \Sigma_k\mathcal{M})^{-1}(x^k, y^k) + \varepsilon^k \tag{3}$$

for a summable sequence $(\varepsilon^k)_{k\in\mathbb{N}}$, and where $\Sigma_k := \begin{pmatrix}\Sigma_{x,k} & \\ & \Sigma_{y,k}\end{pmatrix}$ for some $\Sigma_{x,k} \in \mathbf{Sym}_{++}(\mathbb{R}^n)$ and $\Sigma_{y,k} \in \mathbf{Sym}_{++}(\mathbb{R}^m)$. If $\Sigma_k \le$

[1] https://github.com/Benny44/QPALM

$\Sigma_{k+1} \preceq \Sigma_\infty \in \mathbf{Sym}_{++}(\mathbb{R}^n \times \mathbb{R}^m)$ *holds for all* k*, then* $(x^k, y^k)_{k\in\mathbb{N}}$ *converges to a KKT-optimal pair for* (2).

Proof. We start by observing that for all k it holds that $\mathbf{zer}(\Sigma_k \mathcal{M}) = \mathbf{zer}(\mathcal{M}) = \mathcal{V}_\star$ and that $\Sigma_k \mathcal{M}$ is maximally monotone with respect to the scalar product induced by Σ_k^{-1}. The resolvent $(\mathrm{id} + \Sigma_k \mathcal{M})^{-1}$ is thus firmly nonexpansive in that metric (see [4, Prop. 23.8 and Def. 4.1]): that is, denoting $v^k := (x^k, y^k)$ and $\tilde{v}^{k+1} := v^{k+1} - \varepsilon^k = (\mathrm{id} + \Sigma_k \mathcal{M})^{-1}(v^k)$,

$$\|\tilde{v}^{k+1} - v_\star\|^2_{\Sigma_k^{-1}} \le \|v^k - v_\star\|^2_{\Sigma_k^{-1}} - \|\tilde{v}^{k+1} - v^k\|^2_{\Sigma_k^{-1}} \tag{4}$$

holds for every $v_\star \in \mathcal{V}_\star$. Therefore, since $\|v^{k+1} - v_\star\|_{\Sigma_{k+1}^{-1}} \le \|v^{k+1} - v_\star\|_{\Sigma_k^{-1}} \le \|\tilde{v}^{k+1} - v_\star\|_{\Sigma_k^{-1}} + \|\varepsilon^k\|_{\Sigma_k^{-1}} \le \|v^k - v_\star\|_{\Sigma_k^{-1}} + \|\varepsilon^k\|_{\Sigma_k^{-1}}$ (where the first inequality owes to the fact that $\Sigma_{k+1}^{-1} \le \Sigma_k^{-1}$), it follows from [8, Thm. 3.3] that the proof reduces to showing that any limit point of $(v^k)_{k\in\mathbb{N}}$ belongs to $\mathcal{V}_\star$.

From [8, Prop. 3.2(i)] it follows that the sequence is bounded and that $v^{k+1} - v^k \to 0$ as $k \to \infty$. Suppose that a subsequence $(v^{k_j})_{j\in\mathbb{N}}$ converges to v; then, so do v^{k_j+1} and $\tilde{v}^{k_j+1} = v^{k_j+1} - \varepsilon^{k_j} = (\mathrm{id} + \Sigma_{k_j}\mathcal{M})^{-1} v^{k_j}$. We have

$$\Sigma_{k_j}^{-1}(v^{k_j} - \tilde{v}^{k_j+1}) \in \mathcal{M}(\tilde{v}^{k_j+1});$$

since $(\Sigma_k^{-1})_{k\in\mathbb{N}}$ is upper bounded, the left-hand side converges to 0, and from outer semicontinuity of $\mathcal{M}$ [19, Ex. 12.8(b)] it then follows that $0 \in \mathcal{M}(v)$, proving the claim. □

Let us consider the iterates (3) under the assumptions of Theorem 1, and let us further assume that $\Sigma_{\mathsf{y},k}$ is diagonal for all k. Let $(\tilde{x}^{k+1}, \tilde{y}^{k+1}) := (x^{k+1}, v^{k+1}) - \varepsilon^k = (\mathrm{id} + \Sigma_k \mathcal{M})^{-1}(x^k, y^k)$. Equation (3) reads

$$0 = \Sigma_{\mathsf{x},k}^{-1}(\tilde{x}^{k+1} - x^k) + \nabla f(\tilde{x}^{k+1}) + A^\top \tilde{y}^{k+1} \tag{5a}$$

$$0 \in \tilde{y}^{k+1} - y^k + \Sigma_{\mathsf{y},k}(\partial g^*(\tilde{y}^{k+1}) - A\tilde{x}^{k+1}). \tag{5b}$$

The second condition is in fact equivalent to

$$\begin{aligned}\tilde{y}^{k+1} &= \mathbf{prox}_{\Sigma_{\mathsf{y},k} g^*}(y^k + \Sigma_{\mathsf{y},k} A\tilde{x}^{k+1}) \\ &= y^k + \Sigma_{\mathsf{y},k} A\tilde{x}^{k+1} - \Sigma_{\mathsf{y},k}\, \mathbf{prox}_{\Sigma_{\mathsf{y},k}^{-1} g}(A\tilde{x}^{k+1} + \Sigma_{\mathsf{y},k}^{-1} y^k) \\ &= \Sigma_{\mathsf{y},k}\Big(A\tilde{x}^{k+1} + \Sigma_{\mathsf{y},k}^{-1} y^k - \mathbf{\Pi}_C(A\tilde{x}^{k+1} + \Sigma_{\mathsf{y},k}^{-1} y^k)\Big),\end{aligned}$$

where the first equality follows from the Moreau decomposition [4, Thm. 14.3(ii)], and the second one from the fact that $\Sigma_{\mathsf{y},k}$ is diagonal and set C is separable, hence that the projection on C with respect to the Euclidean norm and that induced by $\Sigma_{\mathsf{y},k}$ coincide. Notice that $A^\top \tilde{y}^{k+1}$ is the gradient of $\frac{1}{2}\mathbf{dist}^2_{\Sigma_{\mathsf{y},k}}(A\cdot + \Sigma_{\mathsf{y},k}^{-1} y^k, C)$ at $\tilde{x}^{k+1}$. Using this in (5a), by introducing an auxiliary variable z^k we obtain that an (exact) resolvent step $(\tilde{x}^{k+1}, \tilde{y}^{k+1}) = (\mathrm{id} + \Sigma_k \mathcal{M})^{-1}(x^k, y^k)$ amounts to

$$\begin{cases} \tilde{x}^{k+1} = \mathbf{arg\,min}_x\, \varphi_k(x) \\ \tilde{z}^{k+1} = Z_k(\tilde{x}^{k+1}) \\ \tilde{y}^{k+1} = y^k + \Sigma_{\mathsf{y},k}(A\tilde{x}^{k+1} - \tilde{z}^{k+1}), \end{cases} \tag{6}$$

where

$$\begin{aligned} Z_k(x) &:= \mathbf{arg\,min}_{z\in C} \tfrac{1}{2}\|z - (Ax + \Sigma_{\mathsf{y},k}^{-1} y^k)\|^2_{\Sigma_{\mathsf{y},k}} = \mathbf{\Pi}_C(Ax + \Sigma_{\mathsf{y},k}^{-1} y^k) \\ &= Ax + \Sigma_{\mathsf{y},k}^{-1} y^k + [\ell - Ax - \Sigma_{\mathsf{y},k}^{-1} y^k]_+ - [Ax + \Sigma_{\mathsf{y},k}^{-1} y^k - u]_+ \end{aligned} \tag{7}$$

is a Lipschitz-continuous mapping, and

$$\varphi_k(x) := f(x) + \tfrac{1}{2}\mathbf{dist}^2_{\Sigma_{\mathsf{y},k}}(Ax + \Sigma_{\mathsf{y},k}^{-1} y^k, C) + \tfrac{1}{2}\|x - x^k\|^2_{\Sigma_{\mathsf{x},k}^{-1}} \tag{8}$$

is a (Lipschitz) differentiable and strongly convex function with gradient

$$\nabla\varphi_k(x) = \nabla f(x) + A^\top(y^k + \Sigma_{\mathsf{y},k}(Ax - Z_k(x))) + \Sigma_{\mathsf{x},k}^{-1}(x - x^k). \tag{9}$$

Algorithm 1 Quadratic Program ALM solver (**QPALM**)

Require $x^0 \in \mathbb{R}^n$, $y^0 \in \mathbb{R}^m$, $\varepsilon, \varepsilon_\mathrm{a}, \varepsilon_\mathrm{r} > 0$, $\vartheta, \rho \in (0,1)$
$\Delta_\mathsf{x}, \Delta_\mathsf{y} > 1$, $\sigma_\mathsf{x}^{\max}, \sigma_\mathsf{y}^{\max} > 0$, $\sigma_\mathsf{y} > 0$

Initialize $\sigma_\mathsf{x} = \mathbf{\Pi}_{[10^{-4},10^4]}(20\frac{\mathbf{max}(1,|f(x^0)|)}{\mathbf{max}(1,\|Ax^0 - \mathbf{\Pi}_C(Ax^0)\|^2)})$ [6, §12.4]
$\Sigma_{\mathsf{x},0} = \sigma_\mathsf{x}\mathrm{I}_m$, $\Sigma_{\mathsf{y},0} = \sigma_\mathsf{y}\mathrm{I}_m$

Repeat for $k = 0, 1, \ldots$

1.1: $x \leftarrow x^k$. Let φ_k, $\nabla\varphi_k$ and $\mathcal{J}_k$ be as in (8), (9) and (12)
1.2: **while** $\|\nabla\varphi_k(x)\|_{\Sigma_{\mathsf{x},k}} > \rho^k\varepsilon$ **do**
1.3: $x \leftarrow x + \tau_\star d$ with d as in (14) and $\tau_\star$ as in Algorithm 2
1.4: $x^{k+1} = x$, $z^{k+1} = \mathbf{\Pi}_C(Ax^{k+1} + \Sigma_{\mathsf{y},k}^{-1} y^k)$
1.5: $y^{k+1} = y^k + \Sigma_{\mathsf{y},k}(Ax^{k+1} - z^{k+1})$
1.6: **if** $(x^{k+1}, z^{k+1}, y^{k+1})$ satisfies (11) **then return** x^{k+1}; **end if**
1.7: $(\Sigma_{\mathsf{y},k+1})_{i,i} = \begin{cases} (\Sigma_{\mathsf{y},k})_{i,i} & \text{if } |(Ax^{k+1} - z^{k+1})_i| \le \vartheta|(Ax^k - z^k)_i|, \\ \mathbf{min}\{\Delta_\mathsf{y}(\Sigma_{\mathsf{y},k})_{i,i},\, \sigma_\mathsf{y}^{\max}\} & \text{otherwise,} \end{cases}$
1.8: $(\Sigma_{\mathsf{x},k+1})_{i,i} = \mathbf{min}\{\Delta_\mathsf{x}(\Sigma_{\mathsf{x},k})_{i,i},\, \sigma_\mathsf{x}^{\max}\}$

Remark 2 (Connection with the proximal ALM [18])**.** The (x,z)-update in (6) can equivalently be expressed as

$$(x^{k+1}, z^{k+1}) = \mathop{\mathbf{arg\,min}}_{(x,z)\in\mathbb{R}^n\times\mathbb{R}^m} \mathcal{L}_{\Sigma_{\mathsf{y},k}}(x, z, y^k) + \tfrac{1}{2}\|x - x^k\|^2_{\Sigma_{\mathsf{x},k}^{-1}},$$

where for $\Sigma_\mathsf{y} \in \mathbf{Sym}_{++}(\mathbb{R}^m)$

$$\mathcal{L}_{\Sigma_\mathsf{y}}(x, z, y) := f(x) + \delta_C(z) + \langle y, Ax\rangle + \tfrac{1}{2}\|Ax - z\|^2_{\Sigma_\mathsf{y}}$$

is the Σ_y-augmented Lagrangian associated to

$$\mathop{\mathbf{minimize}}_{x\in\mathbb{R}^n, z\in\mathbb{R}^m} f(x) + \delta_C(z) \quad \mathbf{subject\ to}\ \ Ax = z, \tag{10}$$

a formulation equivalent to (2). In fact, the iterative scheme (6) simply amounts to the proximal ALM applied to (10). □

IV. The QPALM algorithm

We now describe the proposed proximal ALM based Algorithm 1 for solving (QP). Steps 1.1-1.5 amount to an iteration of proximal ALM. As detailed in Section IV-A, the minimization of φ_k needed for the computation of x^{k+1} can be carried out inexactly. In fact, this is done by means of a tailored extremely fast semismooth Newton method with exact line search, discussed in Sections IV-B and IV-C. Finally, step 1.7 increases the penalty parameters where the constraint norm has not sufficiently decreased [5, §2].

A. Outer and inner loops: early termination criteria

Since the x-update is not available in closed form, each proximal ALM iteration $(x^k, z^k, y^k) \mapsto (x^{k+1}, z^{k+1}, y^{k+1})$ in (6) — which we refer to as an *outer step* — requires an *inner* procedure to find a minimizer x^{k+1} of φ_k. In this subsection we investigate termination criteria both for the outer loop, indicating when to stop the algorithm with a good candidate solution, and for the inner loops so as to ensure that x^{k+1} is computed with enough accuracy to preserve the convergence guarantees of Theorem 1.

1) Outer loop termination: Following the criterion in [21], for fixed absolute and relative tolerances $\varepsilon_\mathrm{a}, \varepsilon_\mathrm{r} > 0$ we say that (x, z, y) is an $(\varepsilon_\mathrm{a}, \varepsilon_\mathrm{r})$-optimal triplet if $y \in \mathcal{N}_C(z)$ and the following hold:

$$\|Qx + q + A^\top y\|_\infty \le \varepsilon_\mathrm{a} + \varepsilon_\mathrm{r}\mathbf{max}\Big\{\|Qx\|_\infty,\ \|A^\top y\|_\infty,\ \|q\|_\infty\Big\} \tag{11a}$$

$$\|Ax - z\|_\infty \le \varepsilon_\mathrm{a} + \varepsilon_\mathrm{r}\mathbf{max}\{\|Ax\|_\infty,\ \|z\|_\infty\}. \tag{11b}$$

Algorithm 2 Exact line search

REQUIRE $x, d \in \mathbb{R}^n$, diagonal $\Sigma \in \mathbf{Sym}_{++}(\mathbb{R}^n)$
PROVIDE optimal stepsize $\tau_\star \in \mathbb{R}$
2.1: Let $\psi' : \mathbb{R} \to \mathbb{R}$, $\alpha, \beta \in \mathbb{R}$ and $\delta, \eta \in \mathbb{R}^{2m}$ be as in (15)
2.2: Define the set of breakpoints of ψ'
$T = \left\{ \frac{\alpha_i + \ell_i}{\delta_i},\ \frac{\alpha_i + u_i}{\delta_i} \mid i = 1, \dots, 2m,\ \delta_i \neq 0 \right\}$
2.3: Sort $T = \{t_1, t_2, \dots\}$ such that $t_i < t_{i+1}$ for all i
2.4: Let $t_i \in T$ be the smallest such that $\psi'(t_i) \geq 0$
2.5: RETURN $\tau_\star = t_{i-1} - \frac{t_i - t_{i-1}}{\psi'(t_i) - \psi'(t_{i-1})} \psi'(t_{i-1})$ (or $\tau_\star = t_1$ if $i = 1$)

From the expression of z^{k+1} at step 1.4 it follows that $Ax^{k+1} + \Sigma_{\mathrm{y},k}^{-1} y^k - z^{k+1} \in \mathcal{N}_C(z^{k+1})$, cf. (1), and hence y^{k+1} as in step 1.5 satisfies $y^{k+1} \in \mathcal{N}_C(z^{k+1})$. A triplet (x^k, z^k, y^k) generated by Algorithm 1 is thus $(\varepsilon_\mathrm{a}, \varepsilon_\mathrm{r})$-optimal if it satisfies (11).

2) Inner loop termination: As shown in Theorem 1, convergence to a solution can still be ensured when the iterates are computed inexactly, provided that the errors have finite sum. Since φ_k as in (8) is $\Sigma_{\mathrm{x},k}^{-1}$-strongly convex and $\Sigma_{\mathrm{x},k}$ is diagonal, from (6) we have that

$$\|\nabla\varphi_k(x)\|_{\Sigma_{\mathrm{x},k}} = \|\nabla\varphi_k(x) - \nabla\varphi_k(\tilde{x}^{k+1})\|_{\Sigma_{\mathrm{x},k}} \geq \|x - \tilde{x}^{k+1}\|.$$

Consequently, condition at step 1.1 ensures that $\|x - \tilde{x}^{k+1}\| \leq \rho^k \varepsilon$ holds for all k. In turn, $\|y^{k+1} - \tilde{y}^{k+1}\| \leq \|A\| \rho^k \varepsilon$ follows from nonexpansiveness of $\mathbf{\Pi}_C$ and finally $\|y^{k+1} - \tilde{y}^{k+1}\| \leq 2\|\Sigma_{\mathrm{y},k}\|\|A\|\rho^k\varepsilon \leq 2\|\Sigma_{\mathrm{y},\infty}\|\|A\|\rho^k\varepsilon$. As a result, the inner termination criterion at step 1.1 guarantees that the error $\|(x^{k+1}, y^{k+1}) - (\mathrm{id} + \Sigma_k \mathcal{M})^{-1}(x^k, y^k)\|$ is summable as required in the convergence analysis of Theorem 1. In the rest of the section we describe how the x-update can be carried out with an efficient minimization strategy.

B. Semismooth Newton method

The diagonal matrix $P_k(x)$ with entries

$$(P_k(x))_{ii} = \begin{cases} 1 & \text{if } \ell_i \leq (Ax + \Sigma_{\mathrm{y},k}^{-1} y^k)_i \leq u_i, \\ 0 & \text{otherwise} \end{cases}$$

is an element of the generalized Jacobian, [10, §7.1], of $\mathbf{\Pi}_C$ at $Ax + \Sigma_{\mathrm{y},k}^{-1} y^k$, see *e.g.* [23, §6.2.d]. Consequently, one element $H_k(x) \in \mathbf{Sym}_{++}(\mathbb{R}^m)$ of the generalized Hessian of φ_k is

$$H_k(x) = Q + A^\top \Sigma_{\mathrm{y},k} (\mathrm{I} - P_k(x)) A + \Sigma_{\mathrm{x},k}^{-1}.$$

Denoting

$$\mathcal{J}_k(x) := \left\{ i \mid (Ax + \Sigma_{\mathrm{y},k}^{-1} y^k)_i \notin [\ell_i, u_i] \right\}, \tag{12}$$

one has that $(\mathrm{I} - P_k(x))_{ii}$ is 1 if $i \in \mathcal{J}_k(x)$ and 0 otherwise. Consequently, we may rewrite the generalized Hessian matrix $H_k(x)$ in a more economic form as

$$H_k(x) = Q + A_{\mathcal{J}_k(x)}^\top (\Sigma_{\mathrm{y},k})_{\mathcal{J}_k(x)} A_{\mathcal{J}_k(x)} + \Sigma_{\mathrm{x},k}^{-1}, \tag{13}$$

where $A_{\mathcal{J}_k(x)}$ is the stacking of the j-th rows of A with $j \in \mathcal{J}_k(x)$, and similarly $(\Sigma_{\mathrm{y},k})_{\mathcal{J}_k(x)}$ is obtained by removing from $\Sigma_{\mathrm{y},k}$ all the i-th rows and columns with $i \notin \mathcal{J}_k(x)$.

A semismooth Newton direction d at x solves $H_k(x)d = -\nabla\varphi_k(x)$. Denoting $\lambda := (\Sigma_{\mathrm{y},k})_{\mathcal{J}_k(x)} A_{\mathcal{J}_k(x)} d$, the computation of d is equivalent to solving the linear system

$$\begin{bmatrix} Q + \Sigma_{\mathrm{x},k}^{-1} & A_{\mathcal{J}_k(x)}^\top \\ A_{\mathcal{J}_k(x)} & (\Sigma_{\mathrm{y},k})_{\mathcal{J}_k(x)}^{-1} \end{bmatrix} \begin{bmatrix} d \\ \lambda \end{bmatrix} = \begin{bmatrix} -\nabla\varphi_k(x) \\ 0 \end{bmatrix}. \tag{14}$$

C. An exact line search

$\nabla\varphi_k(x)$ is piecewise linear, hence so are its sections

$$\mathbb{R} \ni \tau \mapsto \psi_{k,(x,d)}(\tau) := \varphi_k(x + \tau d)$$

for any $d \in \mathbb{R}^n$. This implies that, given a candidate update direction d, the minimization of φ_k can be carried out *without* the need to perform backtrackings, as an optimal stepsize

$$\tau_\star \in \operatorname*{\mathbf{arg\,min}}_{\tau \in \mathbb{R}^n} \psi_{k,(x,d)}(\tau)$$

can be explicitly computed, owing to the fact that $\psi'_{k,(x,d)}$ is a piecewise linear increasing function. Indeed, it follows from (7) and (9) that

$$\begin{aligned} \psi'(\tau) &= \langle \nabla\varphi_k(x + \tau d), d \rangle \\ &= \langle d, \nabla f(x + \tau d) + \Sigma_{\mathrm{x},k}^{-1}(x + \tau d) \rangle \\ &\quad + \langle Ad, y^k + \Sigma_{\mathrm{y},k}(A(x + \tau d) - Z_k(x + \tau d)) \rangle \\ &= \langle d, (Q + \Sigma_{\mathrm{x},k}^{-1})x + q \rangle + \langle \Sigma_{\mathrm{y},k} Ad, [Ax + \Sigma_{\mathrm{y},k}^{-1} y - u + \tau Ad]_+ \rangle \\ &\quad + \tau \langle d, (Q + \Sigma_{\mathrm{x},k}^{-1}) d \rangle - \langle \Sigma_{\mathrm{y},k} Ad, [\ell - Ax - \Sigma_{\mathrm{y},k}^{-1} y - \tau Ad]_+ \rangle \\ &= \eta\tau + \beta + \langle \delta, [\delta\tau - \alpha]_+ \rangle, \end{aligned} \tag{15a}$$

where

$$\begin{cases} \mathbb{R} \ni \eta := \langle d, (Q + \Sigma_{\mathrm{x},k}^{-1}) d \rangle, \\ \mathbb{R} \ni \beta := \langle d, (Q + \Sigma_{\mathrm{x},k}^{-1}) x + q \rangle, \\ \mathbb{R}^{2m} \ni \delta := [-\Sigma_{\mathrm{y},k}^{1/2} Ad \quad \Sigma_{\mathrm{y},k}^{1/2} Ad], \\ \mathbb{R}^{2m} \ni \alpha := \Sigma_{\mathrm{y},k}^{-1/2} [y + \Sigma_{\mathrm{y},k}(Ax - \ell) \quad \Sigma_{\mathrm{y},k}(u - Ax) - y]. \end{cases} \tag{15b}$$

Due to convexity, it now suffices to find τ such that the expression in (15a) is zero, as done in Algorithm 2. We remark that, since $\varphi_k \in C^1$ is strongly convex and piecewise quadratic, the proposed nonsmooth Newton method with exact linesearch would converge in *finitely many iterations even with zero tolerance* [22].

V. Implementation aspects

This section discusses some of the implementation details that are necessary to make QPALM an efficient and competitive algorithm, such as the solution of the linear system at every iteration, preconditioning and infeasibility detection.

A. Linear system

We solve the linear system (14) by means of sparse Cholesky factorization routines. In the first iteration and after every outer iteration, a sparse LDL factorization of the generalized Hessian matrix $H_k(x^k)$ as in (13) is computed. In between inner iterations, the set of active constraints $\mathcal{J}_k$ typically does not change much. Consequently, instead of doing an LDL factorization from scratch, two low rank updates are sufficient, one for the constraints that enter the active set and one for those that leave. As such, the algorithm allows for active set changes where more than one constraint is added and/or dropped, in contrast to active-set methods. Therefore, our algorithm typically requires substantially fewer iterations than active-set methods to find the set of constraints that is active at the solution, while still having the advantage of relatively cheap factorization updates. The aforementioned routines are carried out with software package CHOLMOD [7].

B. Preconditioning

Preconditioning of the problem data aims at mitigating possible adverse effects of ill conditioning. This amounts to scaling problem (QP) to

$$\underset{\bar{x}\in\mathbb{R}^n}{\textbf{minimize}}\ \tfrac{1}{2}\langle \bar{x}, \bar{Q}\bar{x}\rangle + \langle \bar{q}, \bar{x}\rangle \quad \textbf{subject to}\ \ \bar{\ell} \le \bar{A}\bar{x} \le \bar{u}, \quad (16)$$

with $\bar{x} = D^{-1}x$, $\bar{Q} = c_f DQD$, $\bar{q} = c_f Dq$, $\bar{A} = EAD$, $\bar{\ell} = E\ell$ and $\bar{u} = Eu$. The dual variables in this problem are $\bar{y} = c_f E^{-1} y$. Matrices $D \in \mathbb{R}^{n\times n}$ and $E \in \mathbb{R}^{m\times m}$ are diagonal and computed by performing a modified Ruiz equilibration [20] on the constraint matrix A, scaling its rows and columns to have an infinity norm close to 1, as we observed this tends to reduce the number and scope of changes of the active set. The scaling factor c_f for scaling the objective was obtained from [6, §12.5], namely $c_f = \textbf{max}(1, \|\nabla f(x_0)\|_\infty)^{-1}$.

We say the problem is solved when the unscaled termination criteria (11) holds in a triplet $(\bar{x}^k, \bar{z}^k, \bar{y}^k)$, resulting in the following unscaled criterion

$$\begin{aligned} &c_f^{-1}\|D^{-1}(\bar{Q}\bar{x}^k + \bar{q} + \bar{A}^\top \bar{y}^k)\|_\infty \le \\ &\qquad \varepsilon_a + \varepsilon_r c_f^{-1}\,\textbf{max}\left\{\|D^{-1}\bar{Q}\bar{x}^k\|_\infty,\ \|D^{-1}\bar{A}^\top\bar{y}^k\|_\infty,\ \|D^{-1}\bar{q}\|_\infty\right\}, \\ &\|E^{-1}(\bar{A}\bar{x}^k - \bar{z}^k)\|_\infty \le \varepsilon_a + \varepsilon_r\,\textbf{max}\left\{\|E^{-1}\bar{A}\bar{x}^k\|_\infty,\ \|E^{-1}\bar{z}^k\|_\infty\right\}. \end{aligned}$$

C. Infeasibility detection

The proposed method can also detect whether the problem is primal or dual infeasible from the iterates, making use of the criteria given in [3]. Let $\delta\bar{y}$ denote the (potential) change in the dual variable, $\delta\bar{y} = \Sigma_y(\bar{A}\bar{x} - \mathbf{\Pi}_C(\bar{A}\bar{x} + \Sigma_y^{-1}\bar{y}))$, then the problem is primal infeasible if for $\delta\bar{y} \neq 0$

$$\begin{aligned} \|D^{-1}\bar{A}^\top\delta\bar{y}\|_\infty &\le \varepsilon_p\|E\delta\bar{y}\|_\infty, \\ \bar{u}^\top[\delta\bar{y}]_+ + \bar{\ell}^\top[\delta\bar{y}]_- &\le -\varepsilon_p\|E\delta\bar{y}\|_\infty \end{aligned}$$

hold, with $c_f^{-1}E\delta\bar{y}$ the certificate of primal infeasibility. Let $\delta\bar{x}$ denote the update in the primal variable, then the problem is dual infeasible if for $\delta\bar{x} \neq 0$ the following conditions hold

$$\begin{aligned} \|D^{-1}\bar{Q}\delta\bar{x}\|_\infty &\le c_f\varepsilon_d\|D\delta\bar{x}\|_\infty, \\ \bar{q}^\top\delta\bar{x} &\le -c_f\varepsilon_d\|D\delta\bar{x}\|_\infty, \\ (E^{-1}\bar{A}\delta\bar{x})_i &\begin{cases} \ge \varepsilon_d\|D\delta\bar{x}\|_\infty & \text{if } \bar{u}_i = +\infty, \\ \le -\varepsilon_d\|D\delta\bar{x}\|_\infty & \text{if } \bar{\ell}_i = -\infty, \\ \in [-\varepsilon_d, \varepsilon_d]\|D\delta\bar{x}\|_\infty & \text{otherwise.} \end{cases} \end{aligned}$$

In that case, $D\delta\bar{x}$ is the certificate of dual infeasibility.

VI. Numerical simulations

The C implementation of the proposed algorithm was tested for various sets of QPs and benchmarked against state-of-the-art QP solvers: the interior point solver Gurobi [14], the operator splitting based solver OSQP [21] and the active-set solver qpOASES [12]. The first two are also programmed in C, and the third is programmed in C++. All simulations were performed on a notebook with Intel(R) Core(TM) i7-7600U CPU @ 2.80GHz x 2 processor and 16 GB of memory. The problems are solved to medium accuracy, with the tolerances ε_a and ε_r set to 10^{-6} for QPALM and OSQP, as are `terminationTolerance` for qpOASES and `OptimalityTol` and `FeasibilityTol` for Gurobi. Apart from the tolerances, all solvers were run with their default options. For QPALM, the default parameters in Algorithm 1 are: $\varepsilon = 1$, $\rho = 10^{-1}$, $\theta = 0.25$, $\Delta_x = \Delta_y = 10$, $\sigma_x^{\max} = \sigma_y^{\max} = 10^8$, and $\sigma_y = 10^4$. For the test sets below, the runtime in the figures comprises the average runtime on ten problems of the mentioned problem size or conditioning.

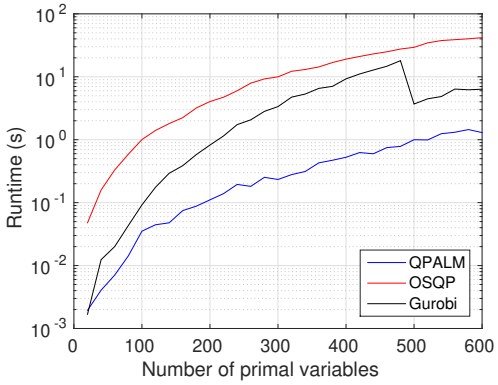


Fig. 1: *Runtime comparison for random LPs of varying sizes.*

A. Linear programs

The first set of tests are linear programs (LPs) with randomly generated data. Of course, an LP is a special case of a QP with a zero Q matrix. The LPs are constructed for 30 values of n equally spaced on a linear scale between 20 and 600. We take $m = 10n$, as typically optimization problems have more constraints than variables. The constraint matrix $A \in \mathbb{R}^{m\times n}$ is set to have 50% nonzero elements drawn from the standard normal distribution, $A_{ij} \sim \mathcal{N}(0,1)$. The linear part of the cost $q \in \mathbb{R}^n$ is a dense vector, $q_i \sim \mathcal{N}(0,1)$. Finally, the elements of the upper and lower bound vectors, $u, l \in \mathbb{R}^m$, are uniformly distributed on the intervals $[0,1]$ and $[-1,0]$ respectively. Figure 1 illustrates a comparison of the runtimes for QPALM, OSQP and Gurobi applied to random LPs of varying sizes. qpOASES is not included in this example, as according to its user manual [12, §4.5] it is not suited for LPs of sizes larger than few hundreds of primal variables, which was observed to be the case. Also OSQP is not suited for LPs as it hit the maximum number of iterations (10^5) for 10 cases, solved inaccurately for 13 cases and solved only in the remaining 3 cases. QPALM is shown to be an efficient solver for LPs, outperforming the simplex and interior point methods that are concurrently tried by Gurobi.

B. Quadratic programs

The second set of tests are QPs with data randomly generated in the same way as in Section VI-A, with an additional positive definite matrix $Q = MM^\top$, with $M \in \mathbb{R}^{n\times n}$ and 50% nonzero elements $M_{ij} \sim \mathcal{N}(0,1)$. Figure 2 illustrates the runtimes of the four solvers for such random QPs. It is clear that QPALM outperforms the other state-of-the-art solvers regardless of the problem size.

C. Ill-conditioned problems

The third test set concerns the conditioning of quadratic programs. In this example, the data from Section VI-B are reproduced for a QP with $n = 100$, but now the impact of the

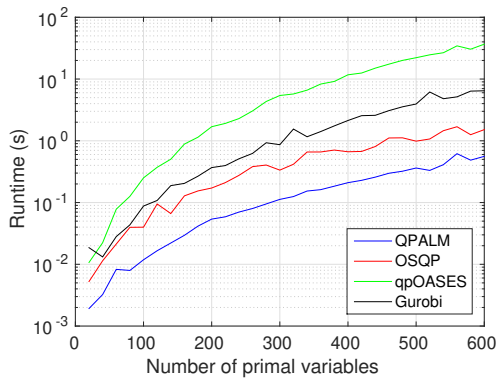


Fig. 2: *Runtime comparison for random QPs of varying sizes.*

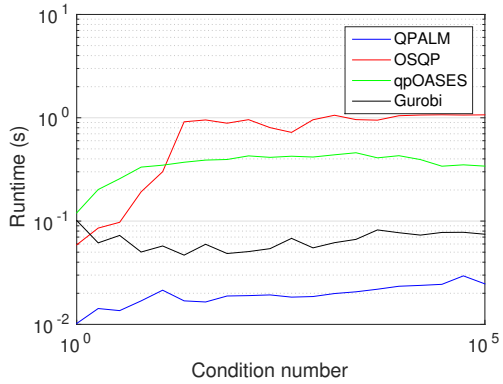


Fig. 3: *Runtime comparison for random QPs with varying conditioning.*

problem conditioning is investigated. For this, we set the condition number κ of the matrices Q and A (using `sprandn` in MATLAB), and also scale q with κ. Figure 3 shows the runtime results for 20 values of κ equally spaced on a logarithmic scale between 10^0 and 10^5. This figure clearly demonstrates that the first-order method OSQP suffers from ill conditioning in the problem despite the offline Ruiz equilibration it operates. From condition number 38 and onwards, OSQP hit the maximum number of iterations (10^5). Also qpOASES experienced difficulties with ill-conditioned problems. From condition number 4833 onwards, it started reporting that the problem was infeasible, while QPALM and Gurobi solved to the same optimal solution. From these results it follows that QPALM, supported by preconditioning as discussed in Section V-B, is competitive with other solvers in terms of robustness to the scaling of the problem data.

VII. Conclusion

This paper presented QPALM, a proximal augmented Lagrangian based solver for QPs that proved to be efficient and robust against scaling in the problem data. Inner minimization procedures rely on semismooth Newton directions and an exact line search which is available in closed form. The iterates, with sparse factorization update routines, allow for large updates in the active set and are more efficient than those of interior point methods and more effective than those of active-set methods. QPALM was shown to compare favorably against state-of-the-art QP solvers, both in runtime and in robustness against problem ill conditioning.

Future work can be focused on considering warm-starting aspects, considering extensions to nonconvex QPs and SOCPs, and on executing a more thorough set of benchmarking examples, focused on problems arising from real applications instead of randomly generated ones.

References

[1] MOSEK ApS. *Introducing the MOSEK Optimization Suite 8.1.0.80*, 2019.

[2] D. Axehill and A. Hansson. A dual gradient projection quadratic programming algorithm tailored for model predictive control. In *2008 47th IEEE Conference on Decision and Control*, pages 3057–3064. IEEE, 2008.

[3] G. Banjac, P. Goulart, B. Stellato, and S. Boyd. Infeasibility detection in the alternating direction method of multipliers for convex optimization. *optimization-online.org*, 2017.

[4] H. H. Bauschke and P. L. Combettes. *Convex analysis and monotone operator theory in Hilbert spaces*. CMS Books in Mathematics. Springer, 2017.

[5] D. P. Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Athena Scientific, 1999.

[6] E. G. Birgin and J. M. Martínez. *Practical augmented Lagrangian methods for constrained optimization*, volume 10. SIAM, 2014.

[7] Y. Chen, T. A. Davis, W. W. Hager, and S. Rajamanickam. Algorithm 887: CHOLMOD, supernodal sparse Cholesky factorization and update/downdate. *ACM Transactions on Mathematical Software (TOMS)*, 35(3):22, 2008.

[8] P. L. Combettes and B. C. Vũ. Variable metric quasi-Fejér monotonicity. *Nonlinear Analysis, Theory, Methods and Applications*, 78(1):17–31, 2013.

[9] N. K. Dhingra, S. Z. Khong, and M. R. Jovanović. A second order primal-dual algorithm for nonsmooth convex composite optimization. In *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*, pages 2868–2873, Dec 2017.

[10] F. Facchinei and JS. Pang. *Finite-dimensional variational inequalities and complementarity problems*. Springer Science & Business Media, 2007.

[11] H. J. Ferreau, H. G. Bock, and M. Diehl. An online active set strategy to overcome the limitations of explicit MPC. *International Journal of Robust and Nonlinear Control*, 18(8):816–830, 2008.

[12] H. J. Ferreau, C. Kirches, A. Potschka, H. G. Bock, and M. Diehl. qpOASES: A parametric active-set algorithm for quadratic programming. *Mathematical Programming Computation*, 6(4):327–363, 2014.

[13] J. C. Gilbert and É. Joannopoulos. OQLA/QPALM–Convex quadratic optimization solvers using the augmented Lagrangian approach, with an appropriate behavior on infeasible or unbounded problems. 2014.

[14] LLC Gurobi Optimization. Gurobi optimizer reference manual, 2018.

[15] J. Nocedal and S. Wright. *Numerical optimization*. Springer Science & Business Media, 2006.

[16] N. Parikh and S. Boyd. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3):127–239, 2014.

[17] P. Patrinos, P. Sopasakis, and H. Sarimveis. A global piecewise smooth Newton method for fast large-scale model predictive control. *Automatica*, 47(9):2016–2022, 2011.

[18] R. T. Rockafellar. Augmented Lagrangians and applications of the proximal point algorithm in convex programming. *Mathematics of operations research*, 1(2):97–116, 1976.

[19] R. T. Rockafellar and R. JB. Wets. *Variational analysis*, volume 317. Springer Science & Business Media, 2011.

[20] Daniel Ruiz. A scaling algorithm to equilibrate both rows and columns norms in matrices. Technical report, Rutherford Appleton Laboratorie, 2001.

[21] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd. OSQP: An operator splitting solver for quadratic programs. In *2018 UKACC 12th International Conference on Control (CONTROL)*, pages 339–339, Sep. 2018.

[22] J. Sun. On piecewise quadratic Newton and trust region problems. *Mathematical Programming*, 76(3):451–467, Mar 1997.

[23] A. Themelis, M. Ahookhosh, and P. Patrinos. On the acceleration of forward-backward splitting via an inexact Newton method. In R. Luke, H. Bauschke, and R. Burachik, editors, *Splitting Algorithms, Modern Operator Theory, and Applications*. Springer. To appear `https://arxiv.org/abs/1811.02935`.