An efficient active-set method with applications to sparse approximations and risk minimization

Spyridon Pougkakiotis[1], Jacek Gondzio[2], and Dionysis Kalogerias[3]

[1]School of Science and Engineering, University of Dundee, UK
[2]School of Mathematics, University of Edinburgh, UK
[3]Department of Electrical Engineering, Yale University, USA

May 8, 2024

Abstract

In this paper we present an efficient active-set method for the solution of convex quadratic programming problems with general piecewise-linear terms in the objective, with applications to sparse approximations and risk-minimization. The algorithm is derived by combining a proximal method of multipliers (PMM) with a standard semismooth Newton method (SSN), and is shown to be globally convergent under minimal assumptions. Further local linear (and potentially superlinear) convergence is shown under standard additional conditions. The major computational bottleneck of the proposed approach arises from the solution of the associated SSN linear systems. These are solved using a Krylov-subspace method, accelerated by certain novel general-purpose preconditioners which are shown to be optimal with respect to the proximal penalty parameters. The preconditioners are easy to store and invert, since they exploit the structure of the nonsmooth terms appearing in the problem's objective to significantly reduce their memory requirements. We showcase the efficiency, robustness, and scalability of the proposed solver on a variety of problems arising in risk-averse portfolio selection, L^1-regularized partial differential equation constrained optimization, quantile regression, and binary classification via linear support vector machines. We provide computational evidence, on real-world datasets, to demonstrate the ability of the solver to efficiently and competitively handle a diverse set of medium- and large-scale optimization instances.

1 Introduction

1.1 Problem formulation

In this paper we consider convex optimization problems of the following form:

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & \left\{ c^\top x + \frac{1}{2}x^\top Q x + \sum_{i=1}^{l} \left((Cx+d)_i\right)_+ + \delta_{\mathcal{K}}(x) \right\}, \\ \text{s.t.} \quad & Ax = b, \end{aligned} \tag{P}$$

where $c \in \mathbb{R}^n$, $Q \in \mathbb{R}^{n\times n}$ is a positive semi-definite matrix, $C \in \mathbb{R}^{l\times n}$, $d \in \mathbb{R}^l$, $A \in \mathbb{R}^{m\times n}$ is a linear constraint matrix with $b \in \mathbb{R}^m$ a given right-hand side, and $\mathcal{K}$ is a closed convex set $\mathcal{K} \triangleq \{x \in \mathbb{R}^n \mid x \in [a_l, a_u]\}$, with $a_l,\ a_u \in \mathbb{R}^n$, such that $(a_l)_i \in \mathbb{R}\cup\{-\infty\},\ (a_u)_i \in \mathbb{R}\cup\{+\infty\}$, for all $i = 1,\ldots,n$. Additionally, $(\cdot)_+ \equiv \max\{\cdot, 0\}$, while $\delta_{\mathcal{K}}(\cdot)$ is an indicator function for the set $\mathcal{K}$, that is, $\delta_{\mathcal{K}}(x) = 0$ if $x \in \mathcal{K}$, and $\delta_{\mathcal{K}}(x) = \infty$ otherwise.

For simplicity, we introduce some additional notation and equivalently write (P) as

$$\begin{aligned} \min_{x\in\mathbb{R}^n} & \left\{ f(x) + h(Cx+d) + \delta_{\mathcal{K}}(x) \right\}, \\ \text{s.t. } & Ax = b. \end{aligned}$$

where we define $f(x) \triangleq c^\top x + \frac{1}{2}x^\top Q x$, and $h(w) \triangleq \sum_{i=1}^{l} (w_i)_+$.

Remark 1. *Let us notice that the model in* (P) *can readily accommodate terms of the form of* $\|Cx+d\|_1$ *where* $C \in \mathbb{R}^{l\times n}$, *and* $d \in \mathbb{R}^l$. *Indeed, letting* $c = c - C^\top \mathbb{1}_l$ *and adding the constant term* $-\mathbb{1}_l^\top d$ *in the objective of* (P), *we notice that*

$$\|Cx+d\|_1 \equiv -\mathbb{1}_l^\top (Cx+d) + \sum_{i=1}^{l} \left(2(Cx+d)_i\right)_+,$$

where $\mathbb{1}_l \triangleq (1,\ldots,1)^\top \in \mathbb{R}^l$. *Similarly, any piecewise-linear term of the form*

$$\sum_{i=1}^{l} \max\left\{(C_1x+d_1)_i,(C_2x+d_2)_i\right\},$$

where $C_1,\ C_2 \in \mathbb{R}^{l\times n}$ *and* $d_1,\ d_2 \in \mathbb{R}^l$, *can also be readily modeled. Specifically, setting* $c = c + C_2^\top \mathbb{1}_l$ *and adding the constant term* $d_2^\top \mathbb{1}_l$ *in the objective yields*

$$\mathbb{1}_l^\top (C_2x+d_2) + \sum_{i=1}^{l} \left((C_1x+d_1-C_2x-d_2)_i\right)_+ \equiv \sum_{i=1}^{l} \max\left\{(C_1x+d_1)_i,(C_2x+d_2)_i\right\}.$$

Finally, it is important to note that model (P) *allows for multiple piecewise-linear terms of the form* $\max\{Cx+d,0_l\}$, $\|Cx+d\|_1$ *or* $\max\left\{C_1x+d_1,C_2x+d_2\right\}$, *since we can always adjust* l *to account for more than one terms. Hence, one can observe that* (P) *is very general and can be used to model a plethora of very important problems that arise in practice.*

Let $F \triangleq [A^\top \ \ -C^\top]^\top$ and $\hat{b} \triangleq [b^\top \ \ d^\top]^\top$. Using Fenchel duality, we can verify (see Appendix A.1) that the dual of (P) is

$$\begin{aligned} \max_{x\in\mathbb{R}^n, y\in\mathbb{R}^{m+l}, z\in\mathbb{R}^n} \quad & \hat{b}^\top y - \frac{1}{2}x^\top Q x - \delta^*_{\mathcal{K}}(z) - h^*(y_{m+1:m+l}), \\ \text{s.t.} \quad & c + Qx - F^\top y + z = 0_n. \end{aligned} \tag{D}$$

Throughout the paper we make use of the following blanket assumption.

Assumption 1. *Problems* (P) *and* (D) *are both feasible and* A *has full row rank.*

If $C = 0_{0,n}$, using [5, Proposition 2.3.4] we note that Assumption 1 implies that there exists a primal-dual triple (x^*, y^*, z^*) solving (P)–(D). If the primal-dual pair (P)–(D) is feasible, it must remain feasible for any $C \in \mathbb{R}^{l\times n}$, since in this case (P) can be written as a convex quadratic problem by appending appropriate (necessarily feasible) linear inequality constraints. Thus, Assumption 1 suffices to guarantee that the solution set of (P)–(D) is non-empty.

In light of the discussion in Remark 1, we observe that **problem** (P) **models a plethora of very important problems arising in several application domains** spanning, among others, operational research, machine learning, data science, and engineering. More specifically, (P) can model linear and convex quadratic programming instances, regularized (fused) lasso instances (often arising in signal or image processing and machine learning, e.g., see [12, 18, 69, 76]), or portfolio allocation problems [1] arising in finance. Furthermore, various optimal control problems can be tackled in the form of (P), such as those arising from L^1-regularized partial differential equation (PDE) constrained optimization, assuming that a *discretize-then-optimize* strategy is adopted (e.g., see [71]). Additionally, various risk-minimization problems with linear random cost functions can be modeled by (P) (e.g., see [37, 58, 62]). Nonetheless, even risk-minimization problems with nonlinear random cost functions, which are typically solved via Gauss-Newton schemes (e.g., see [9]), often require the solution of sub-problems of the form of (P). Finally, continuous relaxations of integer programming problems with applications in operational research (e.g., [35]) often take the form of (P). Given the multitude of problems requiring easy access to (usually accurate) solutions of (P), the derivation of efficient, robust, and scalable solution methods is of paramount importance and has naturally attracted a lot of attention in the literature.

1.2 Existing solution methods

There is a large variety of first-order methods capable of finding an approximate solution to (P). For example, one could employ proximal (sub-)gradient (e.g., see [3]) or splitting schemes (e.g., see [19]). While such solution methods are very general, easy to implement, and require very little memory, they are usually able to find only highly approximate solutions, not exceeding 2- or 3-digits of accuracy. If a more accurate solution is needed, then one has to resort to an approach that utilizes second-order information.

There are three major classes of second-order methods for problems of the form of (P). Those include globalized (smooth, semismooth, quasi or proximal) Newton methods (e.g., see [28, 43, 64]), variants of the proximal point method (e.g., see [17, 20, 30, 36, 39]), or interior point methods (IPMs) applied to a reformulation of (P) (e.g., see [18, 24, 27, 47]).

Most globalized Newton-like approaches or proximal point variants studied in the literature are developed for composite programming problems in which either $C = 0_{0,n}$ (e.g., see [11, 17, 26, 31, 39]) or $\mathcal{K} = \mathbb{R}^n$ and the general $\max\{\cdot, 0\}$ terms are substituted by separable ℓ_1 terms (e.g., see [20, 29, 38]). Nonetheless, there have been developed certain globalized Newton-like schemes, specialized to the case of L^1-regularized PDE-constrained optimization (see [41, 48]), in which both ℓ_1 term as well as box constraints are explicitly handled. We should notice, however, that globalized Newton-like schemes applied to (P) require additional assumptions (on top of Assumption 1) to guarantee convergence or stability of the related Newton linear systems arising as sub-problems. Typically, superlinear convergence of Newton-like schemes is observed "close to a solution". Under additional assumptions, global convergence can be achieved via appropriate line-search or trust-region strategies (e.g., see the developments in [13, 28, 31, 66] and the references therein).

The potential convergence and stability issues arising within Newton-like schemes can be alleviated by combining Newton-like methods with proximal point variants. Solvers based on the proximal point method can achieve superlinear convergence, assuming their penalty parameters increase indefinitely at a suitable rate (e.g., see [55, 56]). For problems (P)–(D), the sub-problems arising within proximal methods are (possibly nonsmooth) convex optimization instances, and are typically solved by means of semismooth Newton strategies. The associated SSN linear systems are better conditioned than their (possibly regularized) interior-point counterparts (e.g., see [4, 18, 24, 27, 72]); however, convergence is expected to be slower, as the method does not enjoy the polynomial worst-case complexity of interior-point methods. Nevertheless, these better conditioned linear systems can, in certain cases, allow one to achieve better computational and/or memory efficiency, especially if the nonsmooth terms are appropriately handled.

Various such solvers have been developed and analyzed in the literature. For example, the authors in [75] developed a dual augmented Lagrangian scheme combined with an SSN method for the solution of semidefinite programming problems and obtained very promising results. This scheme was then utilized for the solution of linear programming problems in [39], and for lasso-regularized problems in [38]. A similar dual augmented Lagrangian-based solver (ALM) was recently developed for square-root Lasso problems in [73]. Nonetheless, those aforementioned dual ALM solvers are brittle, in that slight alterations in the structure of the problem can significantly hinder their employability and efficiency. For example, a computationally attractive extension of [39] to convex quadratic (or, generally, non-separable) programming is by no means obvious and requires a different algorithmic strategy than the dualization approach followed by the authors. Furthermore, there is a **wide gap in the current literature of proximal-SSN methods concerning the efficient solution of the associated SSN linear systems**. While the convergence results appearing in the literature enable the use of inexact linear system solves (e.g., enabling the use of iterative linear algebra, such as Krylov subspace solvers), no general preconditioning strategies have been proposed or analyzed and the question of exploiting efficient linear algebra in this context is still open.

1.3 Contributions

In this paper, we develop an active-set method for (P)–(D) by employing an appropriate proximal method of multipliers (PMM) using a standard backtraking-linesearch semismooth Newton (SSN) strategy for solving the associated PMM sub-problems. The SSN linear systems are approximately solved by means of Krylov subspace methods, using certain novel general-purpose preconditioners. Unlike most proximal point methods given in the literature (e.g., see the primal approaches in [36, 46], the dual approaches in [38, 39] or the primal-dual approaches in [17, 26, 55]), the proposed method introduces proximal terms for each primal and dual variable of the problem, and this results in Newton linear systems which are easy to precondition and solve. We dualize each of the two nonsmooth terms of the objective in (P), which contributes to the simplification of the resulting

SSN linear systems, and paves the way for readily analyzing the convergence of the proposed algorithm. Our contributions are summarized as follows:

- We are able to directly apply the theory developed in [39] to show that the proposed outer proximal method of multipliers is globally convergent under mere feasibility assumptions, and locally superlinearly convergent under standard additional assumptions. Importantly, we should mention that the theory in [39] was developed to analyze the convergence of a proximal semismooth Newton scheme designed for linear programming. Nonetheless, in this work we showcase that the proposed active-set scheme can be analyzed using this theory, and without requiring any additional assumptions other than those employed in [39] in the context of linear programming, despite our model (P) being significantly more general. We also note the inherent difference of our approach compared to the dual augmented Lagrangian schemes developed in [38, 39, 73, 75], which enables the efficient solution of problems with non-separable (smooth or nonsmooth) terms in the objective function of (P).

  Additionally, by introducing an appropriate augmented Lagrangian penalty function (see Section 2), we obtain continuously differentiable PMM sub-problems. As a result, we are able to apply a standard globally convergent (at a local superlinear rate) backtracking-linesearch semismooth Newton scheme (e.g., see [75]) for their solution. Thus, our active-set scheme is readily theoretically supported, and covers a very wide range of problems not currently addressed by similar methods in the literature, unless (P) is reformulated without exploiting its inherent structure.

- The price to pay for the strong theoretical properties of the proposed approach is that the difficulty arising from the nonsmooth terms is hidden in the solution of the associated SSN linear systems. Nonetheless, we identify a principled way of exploiting the resulting active-set structure, that occurs by utilizing the Clarke subdifferential within SSN, to create novel, robust, and highly efficient preconditioners for the solution of these linear systems by means of Krylov subspace solvers. Specifically, we derive and analyze highly effective preconditioning strategies which we show to be optimal with respect to the penalty parameters of the PMM; that is, the conditioning of the preconditioned linear systems does not deteriorate as we increase the penalty parameters of the PMM (which, in turn, accelerates the convergence of the PMM). Additionally, the preconditioners are versatile and general (i.e., not problem-dependent), and are very manageable in terms of memory requirements, giving a strong computational edge of the proposed approach compared to alternative second-order methods. The developed linear algebra kernel of our method showcases the advantages of utilizing active-set schemes compared to alternative (state-of-the-art) interior point methods for similar problems; such a study was missing from the field of augmented Lagrangian methods, and we argue that it should be capitalized more heavily in similar augmented Lagrangian-type approaches appearing in the literature. To the best of our knowledge, aside from the work in [48], which is specialized to the case of L^1-regularized PDE constrained optimization, most proximal Newton-like schemes utilizing Krylov subspace methods do so without employing any (non-trivial) preconditioner (e.g., see [11, 38, 46, 73]).

- We provide extensive computational evidence to support that the proposed approach exhibits a significantly better practical convergence than that predicted by the theory, over a very wide range of applications. Specifically, we showcase the robustness and the efficiency of our approach over a plethora of instances arising from risk-averse portfolio selection problems, L^1-regularized partial differential equation constrained optimization, quantile regression as well as linear classification via support vector machines. In each of these cases, we compare our approach to a robust regularized interior point method (see [27, 49]) which is available on GitHub, and showcase its efficiency not only in terms of CPU time and memory requirements, but also in terms of total iterations required to obtain a sufficiently accurate solution. Indeed, we demonstrate that while our approach requires more iterations compared to the alternative polynomially convergent interior point method, the discrepancy in terms of iteration count is not particularly significant. This is reflected in the computational efficiency of our approach, in terms of CPU time. At the same time, we demonstrate that both second-order solvers are significantly more robust and efficient compared to a state-of-the-art alternating direction method of multipliers [65] (ADMM), since we are interested in finding sufficiently accurate solutions in our numerical experiments, which prevent first-order schemes from being competitive.

- The method is extremely versatile and can solve a plethora of very important applications (as we showcase in Section 5), by explicitly exploiting the structure of the nonsmooth terms appearing in the objective of (P). Additionally, it deals with general box constraints, without requiring the introduction of auxiliary variables to deal with upper and lower bounds separately, something that is required when utilizing conic-based solvers. As a result, the associated linear systems solved within SSN have significantly smaller dimensions, compared to linear systems arising within interior point methods suitable for the solution of problems like (P) (e.g., see [18, 27, 47]), potentially making the proposed approach a more attractive alternative for very large-scale instances.

Before closing this section, we should mention that the algorithm developed in this paper arose in an attempt to consolidate two of our previously developed methodologies laid out in two technical reports ([50, 51]); the former report studies (P) by assuming that the nonsmooth terms are separable, while the latter studies (P) in its full generality. However, these previous approaches dealt with certain non-smooth terms of (P) explicitly. As a result, the associated PMM sub-problems of the developed algorithms were nonsmooth, and that resulted in limited theoretical support of the inner SSN scheme utilized to tackle these sub-problems. This issue was overcome in this consolidated work, by transferring the difficulty to the SSN linear systems, and then utilizing our versatile iterative linear algebra kernel (and associated preconditioning strategy) to ensure that this new and improved version of the active-set schemes developed in [50, 51] is equally scalable but also more robust and better theoretically supported.

To summarize, in Section 2 we derive a proximal method of multipliers and briefly discuss its convergence properties. Then, in Section 3 we present a well-studied globally convergent backtracking-linesearch semismooth Newton scheme used to approximately solve the PMM sub-problems, noticing its local superlinear convergence properties under no additional assumptions. Next, in Section 4, we propose *novel* general-purpose preconditioners for the associated SSN linear systems and analyze their effectiveness. In Section 5, our approach is extensively tested on a wide variety of optimization instances. Finally, we derive some conclusions in Section 6.

Notation. Given a vector x in $\mathbb{R}^n$, $\|x\|$ denotes the Euclidean norm. Letting $R \succ 0_{n,n}$ be a symmetric positive definite matrix, we denote $\|x\|_R^2 = x^\top Rx$. Given a closed set $\mathcal{K} \subset \mathbb{R}^n$, we write $\Pi_{\mathcal{K}}(x) \triangleq \arg\min\{\|x-z\| \mid z \in \mathcal{K}\}$, while for any $R \succ 0_{n,n}$ we write $\text{dist}_R(z,\mathcal{K}) \triangleq \inf_{z' \in \mathcal{K}} \|z - z'\|_R$ (with $\text{dist}_I(z,\mathcal{K}) \equiv \text{dist}(z,\mathcal{K})$). Given a proper and closed convex function $p \colon \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$, we define $\mathbf{prox}_p(u) \triangleq \arg\min_x \left\{p(x) + \frac{1}{2}\|u - x\|^2\right\}$. Given an arbitrary rectangular matrix A, $\sigma_{\max}(A)$ denotes its maximum singular value. For an arbitrary square matrix B, $\lambda(B)$ is the set of eigenvalues of B while $\lambda_{\max}(B)$ (resp., $\lambda_{\min}(B)$) denotes its maximum (resp., minimum) eigenvalue. Given an index set $\mathcal{D}$, $|\mathcal{D}|$ denotes its cardinality. Given a rectangular matrix $A \in \mathbb{R}^{m \times n}$ and an index set $\mathcal{B} \subseteq \{1,\ldots,n\}$, we denote the columns of A, the indices of which belong to $\mathcal{B}$, as $A_{\mathcal{B}}$. Given a square matrix $Q \in \mathbb{R}^{n \times n}$, we denote the subset of columns and rows of Q, the indices of which belong to $\mathcal{B}$, as $(Q)_{\mathcal{B},\mathcal{B}}$. We denote by $\text{Diag}(Q)$ the diagonal matrix with diagonal elements equal to those of Q. Given a matrix $A \in \mathbb{R}^{m \times n}$, and two index sets $\mathcal{B}_1 \subseteq \{1,\ldots,m\}$ and $\mathcal{B}_2 \subseteq \{1,\ldots,n\}$, we denote the subset of rows and columns of A, the indices of which belong to $\mathcal{B}_1$, $\mathcal{B}_2$ respectively, as $(A)_{\mathcal{B}_1,\mathcal{B}_2}$. Finally, given an arbitrary vector d with n components as well as some indices $1 \leq i_1 \leq i_2 \leq n$, we denote by $d_{i_1:i_2}$ the vector $(d_{i_1}, d_{i_1+1}, \ldots, d_{i_2})$. To avoid confusion, indices i are always used to denote entries of vectors or matrices, while k and j are reserved to denote iteration counters (outer (PMM) and inner (SSN), respectively).

2 A primal-dual proximal method of multipliers

In what follows, we derive the proximal augmented Lagrangian penalty function corresponding to the primal problem (P). Using the latter, we derive a primal-dual PMM for solving the pair (P)–(D). The convergence of this PMM scheme is subsequently analyzed, assuming that we are able to find sufficiently accurate solutions to its associated sub-problems. In the next section, we briefly present a standard semismooth Newton scheme for the solution of these sub-problems, with global and local superlinear convergence guarantees.

2.1 Derivation of the PMM

We begin by deriving the Lagrangian associated to (P). First, we define the function

$$\varphi(x) \triangleq f(x) + h(Cx+d) + \delta_{\mathcal{K}}(x) + \delta_{\{0_m\}}(b - Ax).$$

Following the dualization strategy proposed in [60, Chapter 11], we let

$$\hat{\varphi}(x, u', w', v') \triangleq f(x) + h(Cx + d + w') + \delta_{\mathcal{K}}(x + v') + \delta_{\{0_m\}}(b - Ax + u'),$$

for which it holds that $\varphi(x) = \hat{\varphi}(x, 0_m, 0_l, 0_n)$. Then, by letting $y = [y_1^\top \; y_2^\top]^\top$, the Lagrangian associated to (P) reads:

$$\begin{aligned}
\ell(x,y,z) \triangleq\; & \inf_{u',w',v'} \left\{\hat{\varphi}(x,u',w',v') - y_1^\top u' - y_2^\top w' - z^\top v'\right\} \\
=\; & f(x) - \sup_{u'}\left\{y_1^\top u' - \delta_{\{0_m\}}(b - Ax + u')\right\} \\
& - \sup_{w'}\left\{y_2^\top w' - h(Cx + d + w')\right\} - \sup_{v'}\left\{z^\top v' - \delta_{\mathcal{K}}(x + v')\right\} \\
=\; & f(x) - y_1^\top (Ax - b) + y_2^\top (Cx + d) - h^*(y_2) + z^\top x - \delta_{\mathcal{K}}^*(z),
\end{aligned}$$

where we used the definition of the Fenchel conjugate. Before we proceed with the derivation of the augmented Lagrangian associated to $\ell(\cdot)$, let us observe that the Fenchel conjugate of $h(\cdot)$ is $h^*(u) = \delta_{\mathcal{C}}(u)$, where $\mathcal{C} \triangleq \left\{u \in \mathbb{R}^l \middle| v_i \in [0,1],\ i \in \{1,\ldots,l\}\right\}$. Given a penalty parameter $\beta > 0$, the augmented Lagrangian corresponding to (P) reads:

$$\begin{aligned}
\mathcal{L}_\beta(x;y,z) \triangleq\; & f(x) - y_1^\top (Ax - b) + \frac{\beta}{2}\|Ax - b\|^2 \\
& + (Cx+d)^\top \Pi_{\mathcal{C}}\left(y_2 + \beta(Cx+d)\right) - \frac{1}{2\beta}\left\|y_2 - \Pi_{\mathcal{C}}\left(y_2 + \beta(Cx+d)\right)\right\|^2 \\
& - \frac{1}{2\beta}\|z\|^2 + \frac{1}{2\beta}\left\|z + \beta x - \beta \Pi_{\mathcal{K}}\left(\beta^{-1}z + x\right)\right\|^2.
\end{aligned} \tag{2.1}$$

A detailed derivation of the augmented Lagrangian penalty function given in (2.1) can be found in Appendix A.2.

Assume that at iteration $k \geq 0$ we have the estimates (x_k, y_k, z_k) as well as the penalty parameters $\beta_k,\ \rho_k$, such that $\rho_k \triangleq \frac{\beta_k}{\tau_k}$, where $\{\tau_k\}_{k=0}^\infty$ is a non-increasing positive sequence, i.e. $\tau_k > 0$ for all $k \geq 0$. We define the continuously differentiable function

$$\phi(x) \equiv \phi_{\rho_k,\beta_k}(x; x_k, y_k, z_k) \triangleq \mathcal{L}_{\beta_k}(x; y_k, z_k) + \frac{1}{2\rho_k}\|x - x_k\|^2. \tag{2.2}$$

Using the previous notation, we need to find x^* such that

$$\left(\nabla \phi(x^*)\right)^\top (x - x^*) \geq 0, \qquad \forall\, x \in \mathbb{R}^n,$$

where it can be shown (see Appendix A.3) that

$$\begin{aligned}
\nabla \phi(x) = \nabla f(x) - A^\top y_{1_k} + \beta_k A^\top (Ax - b) + C^\top \Pi_{\mathcal{C}}\left(y_{2_k} + \beta_k (Cx + d)\right) \\
+ (z_k + \beta_k x) - \beta_k \Pi_{\mathcal{K}}(\beta_k^{-1} z_k + x) + \rho_k^{-1}(x - x_k).
\end{aligned}$$

We now describe the primal-dual PMM in Algorithm PD-PMM.

Notice that we allow step (2.3) to be computed inexactly. In Section 2.2 we will provide precise conditions on the error sequence guaranteeing that Algorithm PD-PMM achieves global convergence, and additional conditions for achieving a local linear or superlinear rate (where the local superlinear convergence requires that $\beta_k \to \infty$). Further conditions on the starting point and on the starting penalty parameter β_0, required to guarantee a global linear convergence rate, are also discussed.

Algorithm PD-PMM Primal-dual proximal method of mutlipliers

```
Input: $(x_0, y_0, z_0) \in \mathbb{R}^n \times \mathbb{R}^{m+l} \times \mathbb{R}^n$, $\beta_0,\ \beta_\infty,\ \tau_\infty > 0$, $\{\tau_k\}_{k=0}^{\infty}$ such that $\tau_k \searrow \tau_\infty > 0$.
  Choose a sequence of positive numbers $\{\epsilon_k\}$ such that $\epsilon_k \to 0$.
  for $(k = 0, 1, 2, \ldots)$ do
    Find $x_{k+1}$ such that:
        $\|\nabla \phi(x_{k+1})\| \leq \epsilon_k.$    (2.3)
    $y_{k+1} = \begin{bmatrix} y_{1_k} - \beta_k(Ax_{k+1} - b) \\ \Pi_{\mathcal{C}}\left(y_{2_k} + \beta_k(Cx_{k+1} + d)\right) \end{bmatrix}.$    (2.4)
    $z_{k+1} = (z_k + \beta_k x_{k+1}) - \beta_k \Pi_{\mathcal{K}}\big(\beta_k^{-1} z_k + x_{k+1}\big).$    (2.5)
    $\beta_{k+1} \nearrow \beta_\infty \leq \infty, \quad \rho_{k+1} = \frac{\beta_{k+1}}{\tau_{k+1}}.$    (2.6)
  end for
  return $(x_k, y_k, z_k)$.
```

2.2 Convergence analysis

In this section we provide conditions on the error sequence $\{\epsilon_k\}$ in (2.3) that guarantee the convergence of Algorithm PD-PMM, potentially at a global linear or local superlinear rate. The analysis is derived by applying the results shown in [39, Section 2] or [57] (or by an extension of the analyses in [55, 56]) after connecting Algorithm PD-PMM to an appropriate proximal point iteration. As before, we let $F = [A^\top \;\; -C^\top]^\top$ and $\hat{b} = [b^\top \; d^\top]^\top$ and define the maximal monotone operator $T_\ell \colon \mathbb{R}^{2n+m+l} \rightrightarrows \mathbb{R}^{2n+m+l}$, associated to (P)–(D):

$$T_\ell(x, y, z) \triangleq \left\{ (u', v', w') \;\middle|\; u' \in \nabla f(x) - F^\top y + z,\ v' \in Fx - \hat{b} + \begin{bmatrix} 0_m \\ \partial h^*(y_2) \end{bmatrix},\ w' + x \in \partial \delta_{\mathcal{K}}^*(z) \right\}.$$

The inverse of this operator reads

$$T_\ell^{-1}(u', v', w') \triangleq \arg\max_{y,z} \min_{x} \left\{ \ell(x, y, z) + u'^\top x - v'^\top y - w'^\top z \right\}.$$

Notice that Assumption 1 implies that $T_\ell^{-1}(0) \neq \emptyset$. Following the result in [39], and since f is convex quadratic while h is piecewise linear, we note that T_ℓ is in fact a polyhedral multifunction (see [54] for a detailed discussion on the properties of such multifunctions). In light of this property of T_ℓ we note that the following specialized *metric subregularity* condition (see [21] for a definition) holds automatically without any additional assumptions.

Lemma 2.1. *For any $r > 0$, there exists $\kappa > 0$ such that*

$$\operatorname{dist}\big(p, T_\ell^{-1}(0)\big) \leq \kappa \operatorname{dist}\big(0, T_\ell(p)\big), \quad \forall\, p \in \mathbb{R}^{2n+m}, \text{ with } \operatorname{dist}\big(p, T_\ell^{-1}(0)\big) \leq r. \tag{2.7}$$

Proof. The reader is referred to [39, Lemma 2.4] as well as [54]. □

Next, let some sequence of positive definite matrices $\{R_k\}_{k=0}^{\infty}$ be given with $R_k \triangleq \tau_k I_n \oplus I_{m+l} \oplus I_n$, for all $k \geq 0$, where τ_k is defined in Algorithm PD-PMM and $\oplus$ denotes the direct sum of two matrices. We define the single-valued proximal operator $P_k \colon \mathbb{R}^{2n+m+l} \mapsto \mathbb{R}^{2n+m+l}$, associated with T_ℓ:

$$P_k \triangleq \big(R_k + \beta_k T_\ell\big)^{-1} R_k. \tag{2.8}$$

In particular, under our assumptions on the matrices R_k, we have that (e.g. see [56]) for all $(u_1, v_1, w_1),\ (u_2, v_2, w_2) \in \mathbb{R}^{2n+m+l}$, the following inequality (non-expansiveness) holds

$$\|(u_1, v_1, w_1) - P_k(u_2, v_2, w_2)\|_{R_k} \leq \|(u_1, v_1, w_1) - (u_2, v_2, w_2)\|_{R_k}. \tag{2.9}$$

Obviously, we can observe that if $(x^*, y^*, z^*) \in T_\ell^{-1}(0)$, then $P_k(x^*, y^*, z^*) = (x^*, y^*, z^*)$. We are now able to connect Algorithm PD-PMM with the proximal point iteration produced by (2.8).

Proposition 1. *Let $\{(x_k, y_k, z_k)\}_{k=0}^{\infty}$ be a sequence of iterates produced by Algorithm PD-PMM. Then, for every $k \geq 0$ we have that*

$$\|(x_{k+1}, y_{k+1}, z_{k+1}) - P_k(x_k, y_k, z_k)\|_{R_k} \leq \frac{\beta_k}{\min\{\sqrt{\tau_k}, 1\}} \|\nabla \phi(x_{k+1})\|. \tag{2.10}$$

Proof. From (2.2), we observe that

$$\nabla \phi(x_{k+1}) = \nabla \mathcal{L}_{\beta_k}(x_{k+1}; y_k, z_k) + \frac{1}{2\rho_k}(x_{k+1} - x_k).$$

Thus, by utilizing [55, Proposition 7], we deduce

$$(\nabla \phi(x_{k+1}), 0_{m+l}, 0_n) + \beta_k^{-1}(\tau_k(x_k - x_{k+1}), y_k - y_{k+1}, z_k - z_{k+1}) \in T_\ell(x_{k+1}, y_{k+1}, z_{k+1}).$$

Subtracting $P_k(x_k, y_k, z_k)$ from both sides, taking norms, and using the non-expansiveness of P_k (see (2.9)), yields (2.10) and concludes the proof. □

Now that we have established the connection of Algorithm PD-PMM with the proximal point iteration governed by the operator P_k defined in (2.8), we can directly provide conditions on the error sequence in (2.3), to guarantee global (possibly linear) and local linear (potentially superlinear) convergence of Algorithm PD-PMM. To that end, we will make use of certain results, as reported in [39, Section 2]. Firstly, we provide the global convergence result for the algorithm.

Theorem 2.2. *Let Assumption 1 hold. Let $\{(x_k, y_k, z_k)\}_{k=0}^{\infty}$ be generated by Algorithm PD-PMM. Furthermore, assume that we choose a sequence $\{\epsilon_k\}_{k=0}^{\infty}$ in (2.3), such that*

$$\epsilon_k \leq \frac{\min\{\sqrt{\tau_k}, 1\}}{\beta_k} \delta_k, \quad 0 \leq \delta_k, \quad \sum_{k=0}^{\infty} \delta_k < \infty. \tag{2.11}$$

Then, $\{(x_k, y_k, z_k)\}_{k=0}^{\infty}$ is bounded and converges to a primal-dual solution of (P)–(D).

Proof. The proof is omitted since it is a direct application of [39, Theorem 2.3]. □

Next, we discuss local linear (and potentially superlinear) convergence of Algorithm PD-PMM. To that end, let $r > \sum_{k=0}^{\infty} \delta_k$, where δ_k is defined in (2.11). Then, from Lemma 2.1 we know that there exists $\kappa > 0$ associated with r such that

$$\text{dist}\big((x, y, z), T_\ell^{-1}(0)\big) \leq \kappa \ \text{dist}\big(0, T_\ell(x, y, z)\big), \tag{2.12}$$

for all $(x, y, z) \in \mathbb{R}^{2n+m}$ such that $\text{dist}\big((x, y, z), T_\ell^{-1}(0)\big) \leq r$.

Theorem 2.3. *Let Assumption 1 hold, and assume that (x_0, y_0, z_0) is chosen to satisfy*

$$\text{dist}_{R_0}\big((x_0, y_0, z_0), T_\ell^{-1}(0)\big) \leq r - \sum_{k=0}^{\infty} \delta_k,$$

where $\{\delta_k\}_{k=0}^{\infty}$ is given in (2.11). Let also κ be given as in (2.12) and assume that we choose a sequence $\{\epsilon_k\}_{k=0}^{\infty}$ in (2.3) such that

$$\epsilon_k \leq \frac{\min\{\sqrt{\tau_k}, 1\}}{\beta_k} \min\left\{\delta_k, \delta_k' \|(x_{k+1}, y_{k+1}, z_{k+1}) - (x_k, y_k, z_k)\|_{R_k}\right\}, \tag{2.13}$$

where $0 \leq \delta_k$, $\sum_{k=0}^{\infty} \delta_k < \infty$, and $0 \leq \delta_k' < 1$, $\sum_{k=0}^{\infty} \delta_k' < \infty$. Then, for all $k \geq 0$ we have that

$$\text{dist}_{R_{k+1}}\left((x_{k+1}, y_{k+1}, z_{k+1}, T_\ell^{-1}(0)\right) \leq \mu_k \text{dist}_{R_k}\left(x_k, y_k, z_k, T_\ell^{-1}(0)\right),$$

$$\mu_k \triangleq (1 - \delta_k')^{-1}\left(\delta_k' + (1 + \delta_k')\frac{\kappa \gamma_k}{\sqrt{\beta_k^2 + \kappa^2 \gamma_k^2}}\right), \qquad \lim_{k \to \infty} \mu_k = \mu_\infty \triangleq \frac{\kappa \gamma_\infty}{\sqrt{\beta_\infty^2 + \kappa^2 \gamma_\infty^2}},$$

where $\gamma_k \triangleq \max\{\tau_k, 1\}$ and $\gamma_\infty = \max\{\tau_\infty, 1\}$ (noting that $\mu_\infty = 0$, if $\beta_\infty = \infty$).

Proof. The proof is omitted since it follows by direct application of [39, Theorem 2.5] (see also [56, Theorem 2]). □

Remark 2. *Following [39, Remarks 2, 3], we can choose a non-increasing sequence $\{\delta_k'\}_{k=0}^{\infty}$ and a large enough β_0 such that $\mu_0 < 1$, which in turn implies that $\mu_k \leq \mu_0 < 1$, yielding a global linear convergence of both $\text{dist}\big((x_k, y_k, z_k), T_\ell^{-1}(0)\big)$ as well as $\text{dist}_{R_k}\big((x_k, y_k, z_k), T_\ell^{-1}(0)\big)$, assuming that the starting point of the algorithm satisfies the assumption stated in Theorem 2.3. On the other hand, as is implicitly mentioned in Theorem 2.3, if β_k is forced to increase indefinitely, we obtain a local superlinear convergence rate (notice that $\mu_\infty = 0$ if $\beta_\infty = \infty$).*

3 Semismooth Newton method

In this section we briefly present a standard semismooth Newton (SSN) scheme suitable for the solution of problem (2.3), appearing in Algorithm PD-PMM. To that end, we fix the estimates (x_k, y_k, z_k) as well as the penalty parameters $\beta_k,\ \rho_k$. We set $x_{k_0} = x_k$, and at every iteration j of SSN, we approximately solve a system of the following form:

$$J_{k_j} d_x = -\nabla \phi(x_{k_j}), \tag{3.1}$$

where $J_{k_j} \in \partial_x^C \left(\nabla \phi(x_{k_j})\right)$. The symbol $\partial_x^C(\cdot)$ denotes the *Clarke subdifferential* of a function (see [14]) with respect to x, which can be obtained as the convex hull of the *Bouligand subdifferential* ([14]). Any element of the Clarke subdifferential is a *Newton derivative* (see [15, Chapter 13]), since $\nabla \phi(x_{k_j})$ is a *piecewise continuously differentiable* and *regular function.* Using [15, Theorem 14.8], we obtain that for any interval $[l, u]$ on the real line (i.e., $l < u$):

$$\partial_w^C \left(\Pi_{[l,u]}(w_i)\right) = \begin{cases} \{1\}, & \text{if} \quad w \in (l, u), \\ \{0\}, & \text{if} \quad w \notin [l, u], \\ [0, 1], & \text{if} \quad w \in \{l, u\}. \end{cases}$$

For computational reasons, we always choose matrices M_{k_j} from the Bouligand subdifferential. This choice will become apparent in the Section 4. On the other hand, it is well-known (see [43, Theorem 4]) that an inexact semismooth Newton scheme using the Bouligand subdifferential converges at a local linear rate (assuming that the linear systems are solved up to an appropriate accuracy), if the equation $\nabla \phi(x) = 0_n$ is *BD-regular* at the optimum (x_k^*) (that is, each element of the Bouligand subdifferential of $\nabla \phi(x_k^*)$ is nonsingular). Note, however, that since we employ the semismooth Newton scheme to solve the sub-problems arising from Algorithm PD-PMM, we obtain that the resulting nonsmooth equations are indeed BD-regular for every outer iteration $k \geq 0$. Thus, assuming that the associated linear systems are solved up to a sufficient accuracy (see [43, Theorem 4]), we obtain local linear convergence rate of the resulting inexact semismooth Newton scheme. If, additionally, the solution accuracy of the associated linear systems is increased at a suitable rate, the resulting local rate can be superlinear.

We complete the derivation of the SSN by introducing backtracking line-search to ensure global convergence. Then, under no additional assumptions, one can show that SSN is globally convergent. Algorithm SSN outlines a semismooth Newton method for the solution of (2.3). We assume that the associated linear systems are approximately solved by means of a Krylov subspace method. An analysis of the effect of errors arising from the use of Krylov methods within SSN applied to nonsmooth equations can be found in [11].

Algorithm SSN Semismooth Newton method

```
Input: $\epsilon_k > 0$, $\mu \in \left(0, \frac{1}{2}\right)$, $\delta \in (0,1)$, $\gamma \in (0,1]$, $\bar{\eta} \in (0,1)$, $x_{k_0} = x_k$, $y_{k_0} = y_k$.
  for ($j = 0, 1, 2, \ldots$) do
      Choose $J_{k_j} \in \partial_x^C \left(\nabla \phi(x_{k_j})\right)$, and solve
          $$J_{k_j} d_x = -\nabla \phi(x_{k_j}),$$
      such that $\left\|J_{k_j} d_x + \nabla \phi(x_{k_j})\right\| \leq \min\left\{\bar{\eta}, \left\|\nabla \phi(x_{k_j})\right\|^{1+\gamma}\right\}$.
      (Line-search) Set $\alpha_j = \delta^{m_j}$, where $m_j$ is the first non-negative integer for which:
          $$\phi\left(x_{k_j} + \delta^{m_j} d_x\right) \leq \phi(x_{k_j}) + \mu \delta^{m_j} \left(\nabla \phi(x_{k_j})\right)^\top d_x$$
      $x_{k_{j+1}} = x_{k_j} + \alpha_j d_x$.
      if $\left(\left\|\nabla \phi(x_{k_{j+1}})\right\| \leq \epsilon_k\right)$ then
          return $x_{k_{j+1}}$.
      end if
  end for
```

Below we provide a theorem characterizing the convergence of Algorithm SSN.

Theorem 3.1. *Let $\{x_{k_j}\}_{j=0}^{\infty}$ be the sequence of iterates generated by Algorithm SSN. Then, this sequence converges to the unique optimal solution x_k^* of $\min_x \phi(x)$, and $\|x_{k_{j+1}} - x_k^*\| = \mathcal{O}\left(\|x_{k_j} - x_k^*\|^{1+\gamma}\right)$.*

Proof. The proof is omitted since it follows by a trivial extension of the results given in [75, Proposition 3.3, Theorems 3.4-3.5]. □

4 The SSN linear systems

The major bottleneck of the proposed active-set scheme is the approximate solution of the linear systems given in (3.1). Since Algorithm SSN does not require an exact solution, we can utilize preconditioned Krylov subspace solvers for the efficient solution of such systems. In what follows we will derive an equivalent reformulation of (3.1), and we will subsequently propose and analyze a robust preconditioner for the acceleration of the associated Krylov subspace method employed for its solution.

4.1 Derivation of the linear systems

Before we proceed with the derivation of the linear systems, we first need to note that our implementation treats separable nonsmooth terms differently for improved memory and computational efficiency. Specifically, if the problem given in (P) contains a term $\sum_{i=1}^{n} \left((Dx+d)_i\right)_+$, where D is a diagonal ("weight") matrix, we will discuss why and how this is treated differently. To that end, we assume that the matrix C appearing in (P) has the form $C = [\hat{C}^\top \ D]^\top$, where $\hat{C} \in \mathbb{R}^{s\times n}$ is some rectangular matrix, and $D \in \mathbb{R}^{n\times n}$ is a diagonal square matrix, with $s+n = l$. While this structure might seem rather specific, it is very common as it appears when dealing with sparse approximation problems, in which the objective function contains a term of the form $\|Dx\|_1$.

Let $k,\ j \geq 0$ be some arbitrary iterations of Algorithm PD-PMM, and SSN, respectively. Given the assumed structure of matrix C, let us re-write the gradient of (2.2):

$$\begin{aligned}\nabla\phi(x) = \nabla f(x) - A^\top y_{1_k} + \beta_k A^\top(Ax-b) + \hat{C}^\top \Pi_{\mathcal{C}_s}\left((y_{2_k})_{1:s} + \beta_k\left(\hat{C}x + (d)_{1:s}\right)\right) \\ + D\Pi_{\mathcal{C}_n}\left((y_{2_k})_{s+1:l} + \beta_k\left(Dx + (d)_{s+1:l}\right)\right) \\ + (z_k + \beta_k x) - \beta_k \Pi_{\mathcal{K}}(\beta_k^{-1} z_k + x) + \rho_k^{-1}(x - x_k),\end{aligned}$$

where $\mathcal{C}_q = \left\{u \in \mathbb{R}^q \middle| u_i \in [0,1],\ i \in \{1,\ldots,q\}\right\}$. We notice that any element

$$B_{\delta,k_j} \in \partial_w^C\left(\Pi_{\mathcal{K}}(w)\right)\big|_{w=\beta_k^{-1}z_k + x_{k_j}}$$

yields a Newton derivative (see [15, Theorem 14.8]). The same applies for any

$$B_{h,k_j} \in \partial_w^C\left(\Pi_{\mathcal{C}}(w)\right)\big|_{w=y_{2_k}+\beta_k C(x_{k_j}+d)}.$$

We choose B_{δ,k_j}, B_{h,k_j} from the Bouligand subdifferential to improve computational efficiency. We set $B_{\delta,k_j} \in \mathbb{R}^{n\times n}$ as the diagonal matrix with

$$\left(B_{\delta,k_j}\right)_{i,i} \triangleq \begin{cases} 1, & \text{if } \beta_k^{-1} z_{k,i} + x_{k_j,i} \in (a_{l_i}, a_{u_i}), \\ 0, & \text{otherwise,}\end{cases} \tag{4.1}$$

for $i \in \{1,\ldots,n\}$. On the other hand, we separate two cases for $B_{h,k_j} \in \mathbb{R}^{l\times l}$. Specifically, we set

$$\begin{aligned}\left(B_{h,k_j}\right)_{i,i} &\triangleq \begin{cases} 1, & \text{if } (Cx+d)_i \in (0,1), \\ 0, & \text{otherwise,}\end{cases}, \quad \text{for } i \in \{1,\ldots,s\},\ \text{and} \\ \left(B_{h,k_j}\right)_{i,i} &\triangleq \begin{cases} 1, & \text{if } (Cx+d)_i \in [0,1], \\ 0, & \text{otherwise,}\end{cases}, \quad \text{for } i \in \{s+1,\ldots,l\}.\end{aligned} \tag{4.2}$$

Having defined these, we can now formulate the chosen Newton derivative of $\nabla\phi(x_{k_j})$, as

$$M_{k_j} = Q - \beta_k A^\top A + \beta_k C^\top B_{h,k_j} C + \beta_k I_n - \beta_k B_{\delta,k_j} + \rho_k^{-1} I_n.$$

However, by letting $B^1_{h,k_j} \triangleq (B_{h,k_j})_{1:s,1:s}$ and $B^2_{h,k_j} \triangleq (B_{h,k_j})_{s+1:l,s+1:l}$, and by introducing two auxiliary variables $w_1,\ w_2$, we observe that the semismoooth Newton system in (3.1) can equivalently be written as

$$\begin{bmatrix} -(Q+\rho_k^{-1}I_n+\beta_k(I_n-B_{\delta,k_j})+\beta_k D B^2_{h,k_j} D) & A^\top & -\hat{C}^\top \\ A & \beta_k^{-1}I_m & 0_{m,s} \\ -B^1_{h,k_j}\hat{C} & 0_{s,m} & \beta_k^{-1}I_s \end{bmatrix} \begin{bmatrix} d_x \\ w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix}, \tag{4.3}$$

where

$$\begin{bmatrix} \xi_1 \\ \xi_2 \\ \xi_3 \end{bmatrix} \triangleq \begin{bmatrix} c+Qx_{k_j}+D\Pi_{\mathcal{C}_n}\left((y_{2_k})_{s+1:l}+\beta_k(Dx_{k_j}+(d)_{s+1:l})\right)+\tilde{z}_{k_j}+\rho_k^{-1}(x_{k_j}-x_k) \\ \beta_k^{-1}y_{1_k}-(Ax_{k_j}-b) \\ \beta_k^{-1}\Pi_{\mathcal{C}_s}\left(\left(y_{2_k}+\beta_k(Cx_{k_j}+d)\right)_{1:s}\right) \end{bmatrix},$$

with $\tilde{z}_{k_j} \triangleq (z_k+\beta_k x_{k_j})-\beta_k\Pi_{\mathcal{K}}\left(\beta_k^{-1}z_k+x_{k_j}\right)$. Derived from (4.2), we define four index sets:

$$\mathcal{B}^1_{h,j} \triangleq \left\{ i \in \{1,\ldots,s\} \,\middle|\, \left(B^1_{h,k_j}\right)_{i,i} = 1 \right\}, \qquad \mathcal{N}^1_{h,j} \triangleq \{1,\ldots,s\} \setminus \mathcal{B}^1_{h,j},$$

and

$$\mathcal{B}^2_{h,j} \triangleq \left\{ i \in \{1,\ldots,n\} \,\middle|\, \left(B^2_{h,k_j}\right)_{i,i} = 1 \right\}, \qquad \mathcal{N}^2_{h,j} \triangleq \{1,\ldots,s\} \setminus \mathcal{B}^2_{h,j}.$$

Observe that

$$B^1_{h,k_j}\hat{C} = \mathscr{P}\begin{bmatrix} \left(\hat{C}\right)_{\mathcal{B}^1_{h,k_j},1:n} \\ 0_{\left|\mathcal{N}^1_{h,k_j}\right|,n} \end{bmatrix},$$

where $\mathscr{P}$ is an appropriate permutation matrix. Then, from (4.3) we obtain

$$(w_2)_{\mathcal{N}^1_{h,k_j}} = \left(\Pi_{\mathcal{C}_s}\left((y_{2_k}+\beta_k(Cx+d))_{1:s}\right)\right)_{\mathcal{N}^1_{h,k_j}},$$

and (4.3) reduces to

$$\underbrace{\begin{bmatrix} -(Q+\rho_k^{-1}I_n+\beta_k(I_n-B_{\delta,k_j})+\beta_k D B^2_{h,k_j} D) & A^\top & -\left(\hat{C}^\top\right)_{\mathcal{B}^1_{h,k_j}} \\ A & \beta_k^{-1}I_m & 0_{m,|\mathcal{B}^1_{h,k_j}|} \\ -\left(\hat{C}\right)_{\mathcal{B}^1_{h,k_j},1:n} & 0_{|\mathcal{B}^1_{h,k_j}|,m} & \beta_k^{-1}I_{|\mathcal{B}^1_{h,k_j}|} \end{bmatrix}}_{M_{k_j}} \begin{bmatrix} d_x \\ w_1 \\ (w_2)_{\mathcal{B}^1_{h,k_j}} \end{bmatrix}$$
$$= \begin{bmatrix} \xi_1 \\ \xi_2 \\ (\xi_3)_{\mathcal{B}^1_{h,k_j}} \end{bmatrix} + \begin{bmatrix} \left(\hat{C}^\top\right)_{\mathcal{N}^1_{h,k_j}} (w_2)_{\mathcal{N}^1_{h,k_j}} \\ 0_m \\ 0_{|\mathcal{B}^1_{h,k_j}|} \end{bmatrix}, \tag{4.4}$$

the coefficient matrix of which is *quasi-definite* (see [68]), invertible, and its conditioning can be directly controlled by the regularization parameters (β_k, ρ_k) of Algorithm PD-PMM.

4.2 Preconditioning and iterative solution via Krylov subspace methods

Next, we would like to construct an effective preconditioner for M_{k_j} (i.e., the coefficient matrix of (4.4)). Before delving into this, let us observe that in the derivation of this linear system, we utilized the active-set $\mathcal{B}^1_{h,k_j}$ to reduce the size of the resulting coefficient matrix. As before, let us define

$$\mathcal{B}_{\delta,k_j} \triangleq \left\{ i \in \{1,\ldots,n\} \,\middle|\, \left(B_{\delta,k_j}\right)_{i,i} = 1 \right\}, \qquad \mathcal{N}_{\delta,k_j} \triangleq \{1,\ldots,n\} \setminus \mathcal{B}_{\delta,k_j}.$$

In what follows, we will make use of the active sets $\mathcal{B}^2_{h,k_j}$ and $\mathcal{B}_{\delta,k_j}$ to derive cheap and effective preconditioners for the iterative solution of (4.4) by means of Krylov subspace solvers. For simplicity of notation, let us define

$$H_{k_j} \triangleq Q+\rho_k^{-1}I_n+\beta_k(I_n-B_{\delta,k_j})+\beta_k D B^2_{h,k_j} D,$$

and consider an approximation of this matrix that reads

$$\widetilde{H}_{k_j} \triangleq \mathrm{Diag}(Q) + \rho_k^{-1} I_n + \beta_k (I_n - B_{\delta,k_j}) + \beta_k D B_{h,k_j}^2 D.$$

Let us also define the compound matrix

$$G \triangleq \begin{bmatrix} A \\ -\left(\hat{C}\right)_{\mathcal{B}_{h,k_j}^1, 1:n} \end{bmatrix},$$

where we drop the iteration dependence for simplicity, and two sets of interest

$$\mathcal{B}_{k_j} \triangleq \mathcal{B}_{\delta,k_j} \cap \mathcal{N}_{h,k_j}^2, \qquad \mathcal{N}_{k_j} \triangleq \{1, \ldots, n\} \setminus \mathcal{B}_{k_j}.$$

We are now ready to define a positive definite preconditioner for the coefficient matrix in (4.4), as follows:

$$P_{k_j} \triangleq \begin{bmatrix} \widetilde{H}_{k_j} & 0_{n, m+|\mathcal{B}_{h,k_j}^1|} \\ 0_{m+|\mathcal{B}_{h,k_j}^1|, n} & \left(G E_{k_j} G^\top + \beta_k^{-1} I_{m+|\mathcal{B}_{h,k_j}^1|}\right) \end{bmatrix}, \tag{4.5}$$

with $E_{k_j} \in \mathbb{R}^{n \times n}$ the diagonal matrix defined as

$$\left(E_{k_j}\right)_{(i,i)} \triangleq \begin{cases} \frac{1}{\left(\widetilde{H}_{k_j}\right)_{i,i}}, & \text{if} \quad i \in \mathcal{B}_{k_j}, \\ 0, & \text{otherwise.} \end{cases} \tag{4.6}$$

Thus, we utilize a block diagonal preconditioner in which we drop all columns of G that belong to $\mathcal{N}_{k_j}$. This provides a clear motivation behind our choice of matrices in (4.1) and (4.2).

The preconditioner in (4.5) is an extension of the preconditioner proposed in [4] for the solution of linear systems arising from the application of a regularized interior point method to convex quadratic programming. Being a diagonal matrix, E_{k_j} yields a sparse approximation of the Schur complement of the saddle-point matrix in (4.5). This approximation is then used to construct a positive definite block-diagonal preconditioner (i.e. P_{k_j}), which can be used within a symmetric Krylov solver, like the minimum residual (MINRES) method (see [45]). The $(2,2)$ block of P_{k_j} can be inverted via a Cholesky decomposition.

Next, we analyze the spectral properties of the preconditioned matrix $(P_{k_j})^{-1} M_{k_j}$. Let

$$S_{P_{k_j}} \triangleq \left(G \widetilde{H}_{k_j}^{-1} G^\top + \beta_k^{-1} I_m\right), \qquad S_{k_j} \triangleq \left(G E_{k_j} G^\top + \beta_k^{-1} I_m\right).$$

In the following lemma, we bound the eigenvalues of the preconditioned matrix $S_{P_{k_j}}^{-1} S_{k_j}$. This is subsequently used to analyze the spectrum of $P_{k_j}^{-1} M_{k_j}$.

Lemma 4.1. *For any iterates k and j of Algorithms PD-PMM and SSN, respectively, we have*

$$1 \leq \lambda \leq 1 + \sigma_{\max}^2(\hat{F}) \left(\frac{1}{2 + \beta_\infty^{-2} \tau_\infty}\right),$$

where $\lambda \in \lambda\left(S_{P_{k_j}}^{-1} S_{k_j}\right)$, $\hat{F} = [A^\top \ \ -\hat{C}^\top]^\top$, and β_∞, τ_∞ are defined in Algorithm PD-PMM.

Proof. Consider the preconditioned matrix $S_{P_{k_j}}^{-1} S_{k_j}$, and let (λ, u) be its eigenpair. Then, λ must satisfy the following equation:

$$\lambda = \frac{u^\top \left(G_{\mathcal{B}_{k_j}} D_{\mathcal{B}_{k_j}} G_{\mathcal{B}_{k_j}}^\top + \beta_k^{-1} I_m + G_{\mathcal{N}_{k_j}} D_{\mathcal{N}_{k_j}} G_{\mathcal{N}_{k_j}}^\top\right) u}{u^\top \left(G_{\mathcal{B}_{k_j}} D_{\mathcal{B}_{k_j}} G_{\mathcal{B}_{k_j}}^\top + \beta_k^{-1} I_m\right) u},$$

where $D_{\mathcal{B}_{k_j}} = \left(\widetilde{H}_{k_j}^{-1}\right)_{(\mathcal{B}_{k_j}, \mathcal{B}_{k_j})}$, $D_{\mathcal{N}} = \left(\widetilde{H}_{k_j}^{-1}\right)_{(\mathcal{N}_{k_j}, \mathcal{N}_{k_j})}$. The above equality holds since $\widetilde{H}_{k_j}$ is a diagonal matrix, and $\left(E_{k_j}\right)_{(i,i)} = 0$ for every $i \in \mathcal{N}_{k_j}$ (indeed, see the definition in (4.6)). Hence, from positive semi-definiteness of Q, we obtain

$$\begin{aligned} 1 \ \leq \lambda = 1 + \frac{u^\top \left(G_{\mathcal{N}_{k_j}} D_{\mathcal{N}_{k_j}} G_{\mathcal{N}_{k_j}}\right) u}{u^\top \left(G_{\mathcal{B}_{k_j}} D_{\mathcal{B}_{k_j}} G_{\mathcal{B}_{k_j}}^\top + \beta_k^{-1} I_m\right) u} &\leq 1 + \beta_k \sigma_{\max}^2(G_{\mathcal{N}_{k_j}}) \left(2\beta_k + \rho_k^{-1}\right)^{-1} \\ \leq 1 + \sigma_{\max}^2(\hat{F}) \left(\frac{1}{2 + \beta_k^{-2} \tau_k}\right) \leq 1 + \sigma_{\max}^2(\hat{F}) \left(\frac{1}{2 + \beta_\infty^{-2} \tau_\infty}\right), \end{aligned}$$

where we used that $\rho_k = \beta_k/\tau_k$, $\beta_k \leq \beta_\infty$, and $\tau_k \geq \tau_\infty$. □

Given Lemma 4.1, we are now able to invoke [4, Theorem 3] to characterize the spectral properties of the preconditioned matrix $P_{k_j}^{-1}M_{k_j}$. Let

$$\bar{S}_{k_j} \triangleq \left(S_{P_{k_j}}\right)^{-\frac{1}{2}} S_{k_j} \left(S_{P_{k_j}}\right)^{-\frac{1}{2}}, \qquad \bar{H}_{k_j} \triangleq \widetilde{H}_{k_j}^{-1/2} H_{k_j} \widetilde{H}_{k_j}^{-1/2},$$
$$\alpha_{NE} \triangleq \lambda_{\min}(\bar{S}_{k_j}), \quad \beta_{NE} \triangleq \lambda_{\max}(\bar{S}_{k_j}), \quad \alpha_H \triangleq \lambda_{\min}(\bar{H}_{k_j}), \quad \beta_H \triangleq \lambda_{\max}(\bar{H}_{k_j}).$$

Notice that Lemma 4.1 yields upper and lower bounds for α_{NE} and β_{NE}. From the definition of $\bar{H}_{k_j}$ we can also obtain that $\alpha_H \leq 1 \leq \beta_H$ (see [4]). We are now ready to state the spectral properties of the preconditioned matrix $P_{k_j}^{-1}M_{k_j}$.

Theorem 4.2. *Let k and j be some arbitrary iterates of Algorithms* PD-PMM and SSN, *respectively. Then, the eigenvalues of $P_{k_j}^{-1}M_{k_j}$ lie in the union of the following intervals:*

$$I_- \triangleq \left[-\beta_H - \sqrt{\beta_{NE}}, -\alpha_H\right], \qquad I_+ \triangleq \left[\frac{1}{1+\beta_H}, 1 + \sqrt{\beta_{NE} - 1}\right].$$

Proof. The proof follows by direct application of [4, Theorem 3]. □

Remark 3. *By combining Lemma 4.1 with Theorem 4.2, we can observe that the eigenvalues of the preconditioned matrix $P_{k_j}^{-1}M_{k_j}$ are not deteriorating as $\beta_k \to \infty$. In other words, the preconditioner is robust with respect to the penalty parameters β_k, ρ_k of Algorithm* PD-PMM. *Furthermore, our choices of B_{h,k_j}, B_{δ,k_j} in* (4.1) *and* (4.2) *serve the purpose of further sparsifying the preconditioner in* (4.5)*, thus potentially further sparsifying its Cholesky decomposition.*

We note that the preconditioner in (4.5) *is a very efficient choice if $n \geq m+s$, which is the case in several of the experiments considered in Section* 5*. However, any of the preconditioners given in* [27] *can be directly applied for systems appearing in the proposed inner-outer scheme. For example, in the case where $m+s \geq n$ one could adapt the preconditioner given in* [27, Section 3.2] *(using the developments of this section), which would (potentially) be a more efficient choice when compared to that given in* (4.5)*. Additionally, we should mention that for problems considered within this work, a diagonal approximation of the Hessian (within the preconditioner) seems sufficient to deliver very good performance. Indeed, this is the case for a wide range of problems. However, in certain instances, one might consider non-diagonal approximations of the Hessian. In that case, the preconditioner in* (4.5) *can be readily generalized and analyzed, following the developments in* [27, Section 3.1]*. The aforementioned extensions are omitted here for brevity of exposition, but should be considered in a production implementation of the proposed method.*

5 Applications and numerical results

In this section we present various applications that can be modeled by problem (P). Specifically, we consider portfolio optimization, L^1-regularized partial differential equation (PDE) constrained optimization, quantile regression, and binary classification problems. We first discuss (and numerically demonstrate) the effectiveness of the approach for the solution of single-period mean-risk portfolio optimization problems, where risk is measured via the *conditional value at risk* or the *mean absolute semi-deviation*. Subsequently, we apply the proposed scheme to solve L^1-regularized Poisson optimal control problems. Then, we test the proposed approach on quantile regression problems, and finally on binary classification problems via linear support vector machines.

Implementation details Before proceeding to the numerical results, we mention certain implementation details of the proposed algorithm. Following the discussion in Section 4, our implementation takes problems of the following form as input:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & \left\{ c^\top x + \frac{1}{2}x^\top Q x + \sum_{i=1}^{l} \left((Cx+d)_i\right)_+ + \|Dx\|_1 + \delta_{\mathcal{K}}(x) \right\}, \\ \text{s.t.} \quad & Ax = b, \end{aligned}$$

where $D \in \mathbb{R}^{n\times n}$ is a diagonal weight matrix. As discussed in Section 4, the separable nonsmooth term $\|Dx\|_1$ is treated separately as shown in the derivation of the SSN linear systems. The

MATLAB implementation can be found on GitHub[1]. The experiments are run on a PC with a 2.3GHz Intel core i7-12700H processor, 32GB of RAM, using the Windows 11 operating system.

We initialize the PMM penalty parameters as $\beta_0 = 50$, and $\rho_0 = 100$. These are increased in subsequent iterations, at a rate that depends on the residual reduction achieved in that particular PMM iteration. When solving the PMM sub-problems using Algorithm SSN, we use a predictor-corrector-like heuristic in which the first iteration is accepted without line-search and then line-search is activated for subsequent iterations. Algorithm PD-PMM is allowed to run for at most 200 iterations, while in each outer iteration Algorithm SSN is run for at most 40 steps. We utilize the minimum residual (MINRES) method [45] for the solution of the associated SSN linear systems, which is allowed to run for at most 150 iterations. At the beginning of the optimization process the proposed scheme does not utilize the preconditioner given in (4.5), unless MINRES reaches 100 iterations (without a preconditioner). Unless stated otherwise, a primal-dual iterate is accepted as optimal if the conditions given in (A.3) are satisfied for the tolerance specified by the user. All other implementation details follow directly from the developments in Sections 2–4. We refer the reader to the implementation on GitHub for additional details.

In the presented experiments, we compare the proposed approach against the robust regularized interior point solver given in [27, 49] (known as IP-PMM; *interior point-proximal method of multipliers*), the exact and inexact implementations of which can be found on GitHub[2,3]. There are several reasons why the comparison against IP-PMM is meaningful. Firstly, both methods can be classified as second-order solvers. Secondly, both solvers scale well with the problem size, since the inexact variant of IP-PMM also utilizes iterative linear algebra for the solution of its associated linear systems. In that case, both methods rely on a similar preconditioning strategy (we note that the preconditioner analyzed in (4.5) is a variant inspired from the preconditioner proposed in [27, Section 3.1], which is used in the benchmark inexact IP-PMM code). Finally, both methods are implemented in MATLAB. To ensure a fair comparison, we run both the exact and the inexact variants of IP-PMM and report the statistics of the best performing variant. To showcase the efficiency of both second-order solvers (active-set and IP-PMM) against potential first-order alternatives, in the first application (i.e., on the portfolio optimization instances) we will also report the results obtained by applying the well-known ADMM-based OSQP solver given in [65], the implementation of which can also be found on GitHub[4].

5.1 Portfolio optimization

We first consider the mean-risk portfolio selection problem (originally proposed in [42]), where we minimize some convex risk measure, while keeping the expected return of the portfolio above some desirable level. A variety of models for this problem have been extensively analyzed and solved in the literature (e.g., see [1, 34, 40, 58]). The departure from the variance as a measure of risk often allows for great flexibility in the decision making of investors, enabling them to follow potential regulations as well as to better control the risk associated with an "optimal" portfolio. Optimality conditions and existence of solutions for several general deviation measures have been characterized in [59]. Additionally, the comparison of portfolios obtained by minimizing different risk measures has also been considered multiple times (e.g. see [25, 52, 53, 62]).

The method presented in this paper is general and not related to a specific model choice. Indeed, we would like to showcase the efficiency of our approach for obtaining accurate solutions to portfolio optimization problems with two distinct risk measures of practical interest. To that end, we focus on the solution of a standard portfolio selection model that has been used in the literature. All numerical results are obtained on real-world datasets. As a result the problems are of medium-scale. Nonetheless, even for such medium-scale problems, we will be able to demonstrate the efficiency of the proposed approach, when compared to a robust interior point method employed in the literature [18] for similar problems. We also note that both second-order solvers (i.e., IPM and active-set) significantly outperform OSQP on these instances, however the latter was included in the comparison for completeness. Some large-scale instances will be tackled in the following subsections.

We consider the case where a decision vector $x \in \mathbb{R}^n$ represents a portfolio of n financial

[1] https://github.com/spougkakiotis/Krylov_SSN_PMM

[2] https://github.com/spougkakiotis/IP_PMM

[3] https://github.com/spougkakiotis/IP-PMM_QP_Solver

[4] https://github.com/osqp/osqp-matlab

instruments, such that

$$x_i \in [a_{l_i}, a_{u_i}], \quad \text{with } a_{l_i} = 0,\ a_{u_i} \leq 1,\ \text{for all } i = 1, \ldots, n, \quad \text{and } \sum_{i=1}^{n} x_i = 1.$$

This requirement indicates that no short positions are allowed, and no more than $a_{u_i}\%$ of the total wealth can be invested in instrument i. Let $\boldsymbol{\xi} \in \mathbb{R}^n$ denote a random vector, the i-th entry of which represents the random return of the i-th instrument. Then, the negative of the random return of a given portfolio x is given by $g(x, \boldsymbol{\xi}) \triangleq -x^\top \boldsymbol{\xi}$. In this paper we assume that $\boldsymbol{\xi}$ follows some continuous distribution $p(\boldsymbol{\xi})$, as well as that there is a one-to-one correspondence between percentage return and monetary value (as in [58, Section 3]). Additionally, given some expected benchmark return r (e.g., the *market index*), we only consider portfolios that yield an expected return above a certain threshold, i.e. $\mathbb{E}[-g(x, \boldsymbol{\xi})] \geq r$.

Finally, given the previously stated constraints, we would like to minimize some convex risk measure $\varrho(\cdot)$ of interest. By putting everything together, the model reads as

$$\begin{aligned} \min_{x}\ & \varrho\left(g(x, \boldsymbol{\xi})\right), \\ \text{s.t. } & \sum_{i=1}^{n} x_i = 1, \\ & \mathbb{E}\left[-g(x, \boldsymbol{\xi})\right] \geq r, \\ & x_i \in [a_{l_i}, a_{u_i}], \qquad i = 1, \ldots, n. \end{aligned} \tag{5.1}$$

There are several methods for solving such stochastic problems; let us mention two important variants. There is the *parametric approach* (e.g., as in [1, 58]), where one assumes that the returns follow some known distribution which is subsequently sampled to yield finite-dimensional optimization problems, and the *sampling approach* (e.g., as in [34]), where one obtains a finite number of samples (without assuming a specific distribution). Such samples are often obtained by historical observations, and this approach is also followed in this paper. It is well-known that historical data cannot fully predict the future (see [34]), however it is a widely-used practice. The reader is referred to [32] for an extensive study on probabilistic models for portfolio selection problems, with practical considerations.

Additional soft or hard constraints can be included when solving a portfolio selection problem. Such constraints can either be incorporated directly via the use of a model (e.g., see [6, Section 2]) as hard constraints or by including appropriate penalty terms in the objective (soft constraints). It is important to note that the model given in (P) is quite general and as a result has great expressive power, allowing one to incorporate various important risk measures (and their combinations), as well as modeling constraints of interest.

Real-world datasets: In what follows, we solve two different instances of problem (5.1). In particular, we consider two risk measures; the conditional value at risk (e.g., see [58]), as well as the mean absolute semi-deviation (e.g., see [61, Section 6.2.2.], noting that this is in fact equivalent to the mean absolute deviation originally proposed in [33]).

We showcase the effectiveness of the proposed approach on 6 real datasets taken from [8]. Each dataset contains time series for weekly asset returns and market indexes for different major stock markets, namely, DowJones, NASDAQ100, FTSE100, SP500, NASDAQComp, and FF49Industries. In the first 5 markets the authors in [8] provide the market indexes, while for the last dataset the uniform allocation strategy is considered as a benchmark. Additional information on the datasets is collected in Table 1. We note that the authors in [8] disregard stocks with less than 10 years of observations.

5.1.1 Conditional value at risk

First, we consider portfolio optimization problems that seek a solution minimizing the conditional value at risk; a measure which is known to be *coherent* (see [2] for a definition of coherent risk measures). In particular using the notation introduced earlier, we consider the following optimization problem

$$\min_{x \in \mathbb{R}^n} \left\{ \mathrm{CVaR}_\alpha\left(g(x, \boldsymbol{\xi})\right) + \delta_{\mathcal{K}}(x) \right\}, \qquad \text{s.t. } Ax = b, \tag{5.2}$$

Table 1: Portfolio optimization datasets.

Name	# of assets	# of data points	Timeline
DowJones [DJ]	28	1363	Feb. 1990–Apr. 2016
NASDAQ100 [NDQ]	82	596	Nov. 2004–Apr. 2016
FTSE100 [FT]	83	717	Jul. 2002–Apr. 2016
FF49Industries [FF]	49	2325	Jul. 1969–Jul. 2015
SP500 [SP]	442	595	Nov. 2004–Apr. 2016
NASDAQComp [NDC]	1203	685	Feb. 2003–Apr. 2016

where $g(x, \boldsymbol{\xi})$ is the random cost function, $A \in \mathbb{R}^{m\times n}$ models the linear constraint matrix of problem (5.1) (where an auxiliary variable has been introduced to transform the inequality constraint involving the discretized expectation into an equality), and $\mathcal{K} \triangleq [a_l, a_u]$. In the above $1-\alpha \in (0,1)$ is the confidence level. It is well-known ([58]) that given a continuous random variable X, the conditional value at risk can be computed as an infimal convolution

$$\mathrm{CVaR}_\alpha\left(X\right) \triangleq \min_{t\in\mathbb{R}} \left\{t + \alpha^{-1}\mathbb{E}\left[\left(X-t\right)_+\right]\right\}.$$

Thus, we can write problem (5.2) in the following equivalent form:

$$\min_{(x,t)\in\mathbb{R}^n\times\mathbb{R}} \left\{ t + \frac{1}{l\alpha}\sum_{i=1}^{l}\left(-\xi_i^\top x - t\right)_+ + \delta_{\mathcal{K}}(x)\right\}, \qquad \text{s.t. } Ax = b,$$

where all expectations have been substituted by summations since we assume the availability of a dataset $\{\xi_1, \ldots, \xi_l\}$.

We solve problem (5.2) using the proposed active-set method (AS), the interior point-proximal method of multipliers (IP-PMM) given in [49], and OSQP ([65]). We set `tol` $= 10^{-5}$, and run the three methods for each of the datasets described in Table 1 for varying confidence level. We report the confidence parameter α, the number of PMM, SSN, and Krylov iterations required by the active-set scheme, the number IP-PMM and Krylov iterations required by the interior point scheme (with Krylov iterations reported only if the inexact IP-PMM variant performs better), and the number of OSQP iterations, the CPU time needed by each of the three schemes, as well as the number of factorizations used within the active set (PMM-SSN) scheme. As already mentioned earlier, the AS scheme does not utilize a preconditioner until it is deemed necessary. Additionally, it often happens that the active-set is not altered from one iteration to the next, and the factorization of the preconditioner does not need to be re-computed. The results are collected in Table 2. In all the numerical results that follow, the lowest running time exhibited by a solver, assuming it successfully converged, is presented in bold.

From Table 2 we observe that for the smaller instances (DowJones, NASDAQ100, FTSE100, FF49Industries) both second-order methods perform well and are comparable in terms of CPU time. Nonetheless, even in this case, the proposed active-set solver requires less computations per iteration. Indeed, this can be seen by the fact that the two methods achieve similar times but the active-set scheme is performing significantly more SSN iterations. On the other hand, for the larger instances (SP500, NASDAQComp) the proposed active-set scheme outperforms the interior point method significantly. This is mostly due to the efficiency gained from the lower memory requirements and cheaper factorizations of the preconditioner associated with the active-set solver. We additionally observe that both second-order methods are robust with respect to the confidence level and consistently outperform OSQP, which struggles to find a 5-digit accurate solution. We note that OSQP could potentially be competitive for smaller tolerances (e.g., for finding a 3-digit accurate solution); however, the application under consideration dictates that an accurate solution is needed, since a small improvement in the portfolio output can translate into considerable profits in practice. Concerning IP-PMM, we note that the inexact variant is only elected for some of the largest instances (i.e., SP500 and NASDAQComp). This is expected, since these problems are small- to medium-scale. Nonetheless, large-scale instances will be tackled in the following subsections.

Table 2: CVaR portfolio selection: varying confidence level (`tol` $= 10^{-5}$).

Dataset	α	Iterations			Time (s)		
		PMM(SSN)[Fact.]{Krylov}	IP–PMM(Krylov)	OSQP	AS	IP–PMM	OSQP
DJ	0.05	29(78)[15]{3841}	22(–)[1]	32,575	**0.23**	0.91	10.94
	0.10	33(88)[24]{4116}	22(–)	42,625	**0.40**	0.84	13.61
	0.15	36(103)[35]{3954}	19(–)	43,100	**0.54**	0.75	14.45
NDQ	0.05	26(80)[20]{3722}	17(–)	30,575	**0.28**	0.51	7.92
	0.10	29(104)[27]{4781}	18(–)	34,700	**0.46**	0.55	8.81
	0.15	32(100)[33]{4352}	17(–)	40,500	**0.45**	0.53	10.46
FE	0.05	30(108)[73]{2208}	21(–)	21,825	**0.82**	1.02	10.96
	0.10	31(101)[46]{3266}	16(–)	35,825	**0.52**	0.72	17.70
	0.15	34(99)[54]{2798}	18(–)	36,550	**0.65**	0.93	18.24
FF	0.05	38(118)[31]{3491}	27(–)	50,000‡[2]	**0.60**	3.80	59.16‡
	0.10	37(132)[20]{2777}	23(–)	50,000‡	**0.76**	3.21	57.92‡
	0.15	38(108)[19]{4169}	20(–)	50,000‡	**0.88**	2.82	51.25‡
SP	0.05	29(117)[93]{2753}	24(699)	21,075	**1.87**	6.57	65.70
	0.10	32(119)[70]{2861}	27(877)	32,375	**1.70**	7.79	99.39
	0.15	31(106)[64]{3069}	23(–)	39,225	**1.76**	13.49	119.95
NDC	0.05	32(225)[154]{4538}	26(593)	16,100	**6.89**	23.65	143.82
	0.10	31(168)[123]{3441}	27(830)	31,275	**6.91**	27.54	291.97
	0.15	34(179)[124]{3538}	28(925)	16,100	**8.65**	31.43	403.75

[1] – in the Krylov iterations for IP-PMM indicates that the exact variant of IP-PMM was used.
[2] ‡ indicates that the solver reached the maximum number of iterations.

5.1.2 Mean absolute semi-deviation

Next, we consider portfolio optimization problems that seek a solution minimizing the mean absolute semi-deviation, which is also coherent. We consider the following optimization problem

$$\min_{x\in\mathbb{R}^n} \left\{ \mathrm{MAsD}\left(g(x,\boldsymbol{\xi})\right) + \delta_{\mathcal{K}}(x) \right\}, \qquad \text{s.t. } Ax = b, \tag{5.3}$$

where, given a continuous random variable X, the associated risk is defined as

$$\mathrm{MAsD}\left(X\right) \triangleq \mathbb{E}\left[\left(X - \mathbb{E}[X]\right)_+\right] \equiv \frac{1}{2}\mathbb{E}\left|X - \mathbb{E}[X]\right|,$$

where the equivalence follows from [61, Prospotion 6.1]. Given a dataset $\{\xi_1,\ldots,\xi_l\}$, problem (5.3) can be written as

$$\begin{aligned} &\min_{x\in\mathbb{R}^n} \left\{ \frac{1}{l}\sum_{i=1}^{l}\left(-\xi_i^\top x + \boldsymbol{\mu}^\top x\right)_+ + \|Dx\|_1 + \delta_{\mathcal{K}}(x) \right\}, \\ &\text{s.t.} \qquad Ax = b, \end{aligned}$$

where $\boldsymbol{\mu} \triangleq \frac{1}{l}\sum_{i=1}^{l}\xi_i^\top x$. Note that this model is in the form of (P). We fix `tol` $= 10^{-5}$ and run the three methods on the 6 datasets. The results are collected in Table 3.

Table 3: MAsD portfolio selection (`tol` $= 10^{-5}$).

Dataset	Iterations			Time (s)		
	PMM(SSN)[Fact.]{Krylov}	IP–PMM(Krylov)	OSQP	AS	IP–PMM	OSQP
DJ	48(105)[31]{2686}	14(–)	14,050	0.79	**0.54**	4.60
NDQ	41(119)[37]{3436}	13(–)	40,550	0.44	**0.28**	10.72
FE	42(133)[53]{3020}	12(–)	47,125	0.69	**0.30**	23.29
FF	52(100)[0]{4289}	18(–)	12,050	**0.77**	2.70	14.24
SP	42(180)[65]{4313}	19(–)	41,275	**2.06**	4.09	128.09
NDC	46(176)[75]{4569}	14(–)	41,000	**6.74**	10.57	367.32

From Table 3 we observe that both second-order schemes are, once again, robust and comparable for all instances. In this case, the larger instances (SP500, NASDAQComp) were solved in comparable time by both solvers. Nevertheless, it is important to note that the active-set scheme is the better choice, since it has significantly less memory requirements (this is due to solving a smaller problem formulation while also only utilizing information from the active sets when constructing its associated preconditioners). Notably the proposed method is robust (i.e., converges reliably to an accurate solution). Finally, as in the case of the CVaR instances, OSQP struggled to find accurate solutions at a reasonable number of iterations, making it a less efficient choice for such problems. In subsequent experiments, we will drop OSQP from our benchmarks since it performed significantly worse compared to both second-order solvers.

5.2 PDE-constrained optimization

Let us now test the proposed methodology on some optimization problems with partial differential equation constraints. We consider optimal control problems of the following form:

$$\begin{aligned} \min_{\mathrm{y},\mathrm{u}} \ & \frac{1}{2}\|\mathrm{y}-\bar{\mathrm{y}}\|_{\mathrm{L}^2(\Omega)}^2 + \frac{\alpha_1}{2}\|\mathrm{u}\|_{\mathrm{L}^1(\Omega)}^2 + \frac{\alpha_2}{2}\|\mathrm{u}\|_{\mathrm{L}^2(\Omega)}^2, \\ \text{s.t.} \ & \mathrm{Dy}(\boldsymbol{x}) + \mathrm{u}(\boldsymbol{x}) = \mathrm{g}(\boldsymbol{x}), \quad \mathrm{u_a}(\boldsymbol{x}) \le \mathrm{u}(\boldsymbol{x}) \le \mathrm{u_b}(\boldsymbol{x}), \end{aligned} \tag{5.4}$$

where $(\mathrm{y},\mathrm{u}) \in \mathrm{H}^1(\Omega) \times \mathrm{L}^2(\Omega)$, D is some linear differential operator, $\boldsymbol{x}$ is a 2-dimensional spatial variable, and $\alpha_1,\ \alpha_2 \ge 0$ are the regularization parameters of the control variable. The problem is considered on a given compact spatial domain Ω, where $\Omega \subset \mathbb{R}^2$ has boundary $\partial\Omega$, and is equipped with Dirichlet boundary conditions. The algebraic inequality constraints are assumed to hold a.e. on Ω, while $\mathrm{u_a}$ and $\mathrm{u_b}$ may take the form of constants or functions of the spatial variables.

We solve problem (5.4) via a discretize-then-optimize strategy. We employ the Q1 finite element discretization implemented in IFISS[5] (see [22, 23]) which yields a sequence of ℓ_1-regularized convex quadratic programming problems in the form of (P). We note that the discretization of the smooth parts of problem (5.4) follows a standarad Galerkin approach (e.g., see [67]), while the L^1 term is discretized by the *nodal quadrature rule* as in [63] (an approximation that achieves a first-order convergence–see [71]).

In what follows, we consider two-dimensional L^1/L^2-regularized Poisson optimal control problems. The problem is posed on $\Omega = (0,1)^2$. Following [47, Section 5.1], we set constant control bounds $\mathrm{u}_a = -2$, $\mathrm{u}_b = 1.5$, and the desired state as $\bar{\mathrm{y}} = \sin(\pi x_1)\sin(\pi x_2)$. In order to ensure that both solvers exhibit a consistent behaviour over PDE-constrained optimization problems of arbitrary size, we enforce stopping conditions with *absolute* tolerance `tol` $= 10^{-4}$. This is important in this case, since IP-PMM solves a smooth reformulation of the problem, and its relative tolerance criteria differ significantly from those of the active-set method when the discretized PDE size increases. In Table 4, we fix the L^2 regularization parameter to the value $\alpha_2 = 10^{-2}$, and present the runs of the two methods (i.e. active-set, IP-PMM) for varying L^1 regularization (i.e. α_1) as well as grid size. We report the size of the resulting discretized problems (before any reformulation), the value of α_1, the number iterations required by each solver, as well as the total time to convergence.

From Table 4 we observe that both the active-set and IP-PMM are competitive, with the active-set method achieving faster convergence in terms of CPU time. Let us note that the number of Krylov iterations required to obtain an absolute 4-digit solution by both solvers is comparable. Thus, the active-set approach is more efficient, since it requires less memory (as it solves smaller and better conditioned linear systems, and thus the inversion of the associated preconditioners is less resource-consuming). On the other hand, we can observe that both AS and IP-PMM have a very consistent behaviour for a wide range of values of the associated problem parameters.

Next, we fix $\alpha_1 = 10^{-4}$ and the problem size, and run three solvers on PDE-constrained optimization instances with varying L^2 regularization. The results are collected in Table 5, and showcase the robustness of the proposed active-set solver with respect to the problem parameters. Indeed, one can observe that the active-set scheme exhibits more consistent convergence behaviour compared to IP-PMM as we alter the problem parameters.

5.3 Penalized quantile regression

Let us now consider linear regression models of the following form

$$y_i = \beta_0 + \xi_i^\top \beta + \epsilon_i, \qquad i \in \{1,\ldots,l\}$$

[5] https://personalpages.manchester.ac.uk/staff/david.silvester/ifiss/default.htm

Table 4: Poisson control: varying grid size and L^1 regularization ($\alpha_2 = 10^{-2}$).

n	α_1	Iterations		Time (s)	
		PMM(SSN)[Fact.]{Krylov}	IP-PMM(Krylov)	AS	IP-PMM
$8.45 \cdot 10^3$	10^{-2}	14(39)[28]{1691}	13(1096)	**3.14**	4.36
	10^{-4}	14(39)[23]{1687}	14(1253)	**2.67**	4.50
	10^{-6}	14(39)[23]{1686}	14(1256)	**2.49**	4.11
$3.32 \cdot 10^4$	10^{-2}	16(34)[28]{1295}	15(1147)	**9.98**	18.97
	10^{-4}	16(34)[24]{1291}	15(1379)	**9.98**	22.55
	10^{-6}	16(34)[24]{1291}	15(1381)	**9.82**	23.40
$1.32 \cdot 10^5$	10^{-2}	18(31)[29]{1149}	14(876)	**43.98**	67.10
	10^{-4}	18(31)[26]{1143}	15(1111)	**42.56**	83.88
	10^{-6}	18(31)[26]{1143}	15(1116)	**42.69**	82.23
$5.26 \cdot 10^5$	10^{-2}	18(32)[30]{1251}	15(1166)	**246.08**	450.25
	10^{-4}	18(32)[30]{1250}	15(1171)	**218.24**	743.47
	10^{-6}	18(32)[30]{1250}	15(1174)	**259.12**	728.32
$2.11 \cdot 10^6$	10^{-2}	21(38)[35]{1535}	15(909)	**1199.34**	2676.53
	10^{-4}	21(38)[36]{1534}	16(1068)	**1342.57**	1878.91
	10^{-6}	21(38)[36]{1534}	16(1068)	**1452.42**	1612.56

[1] ‡ indicates that the solver reached the maximum number of iterations.
[2] † indicates that the solver ran out of memory.

Table 5: Poisson control: varying L^2 regularization $(n, \alpha_1) = (1.32 \cdot 10^5, 10^{-4})$.

α_2	Iterations		Time (s)	
	PMM(SSN)[Fact.]{Krylov}	IP-PMM(Krylov)	AS	IP-PMM
10^{-1}	18(31)[26]{1155}	18(3979)	**44.06**	423.02
10^{-4}	18(31)[26]{1142}	15(661)	**77.29**	106.84
10^{-6}	18(31)[26]{1141}	15(607)	**60.52**	103.41
0	18(31)[26]{1144}	15(499)	**70.45**	73.72

[1] ‡ indicates that the solver reached the maximum number of iterations.

where ξ_i is a d-dimensional vector of covariates, (β_0, β) are the regression coefficients and ϵ_i is some random error. A very popular problem in statistics is the estimation of the optimal coefficients, in the sense of minimizing a model of the following form:

$$\min_{(\beta_0,\beta)\,\in\,\mathbb{R}\times\mathbb{R}^d} \left\{ \frac{1}{l}\sum_{i=1}^{l} \ell\left(y_i - \beta_0 - \xi_i^\top \beta\right) + \lambda p(\beta) \right\}, \tag{5.5}$$

where $\ell(\cdot)$ is some loss function and $p(\cdot)$ is a penalty function with an associated regularization parameter $\lambda \geq 0$. Following [76], we consider the elastic-net penalty,

$$p(\beta) \equiv p_\tau(\beta) \triangleq \tau\|\beta\|_1 + \frac{1-\tau}{2}\|\beta\|_2^2, \qquad 0 \leq \tau \leq 1.$$

For the loss function, we employ the quantile loss

$$\ell(w) \equiv \rho_\alpha(w) \triangleq (1-\alpha)\, w_- + \alpha w_+ = \frac{1}{2}\left(|w| + (2\alpha - 1)w\right), \qquad 0 < \alpha < 1, \tag{5.6}$$

where $w \in \mathbb{R}$. Notice that the case $\alpha = \frac{1}{2}$ yields the absolute loss. Letting $x = \begin{bmatrix}\beta_0 & \beta^\top\end{bmatrix}^\top$, and using Remark 1, we can re-write problem (5.5) in the form of (P), as

$$\min_{x\,\in\,\mathbb{R}^{1+d}} \left\{ (\alpha - 1)\mathbb{1}_l^\top (Cx + d) + \frac{1}{2}x^\top Q x + \sum_{i=1}^{l}((Cx+d)_i)_+ + \|Dx\|_1 \right\},$$

where

$$C_{:,i} = -\frac{1}{l}[1 \quad \xi_i^\top], \; d_i = \frac{1}{l}y_i, \quad i = 1,\ldots,l, \qquad Q = \begin{bmatrix} 0 & 0_{1,d} \\ 0_{d,1} & \lambda(1-\tau)I_d \end{bmatrix}, \qquad D = \begin{bmatrix} 0 & 0_{1,d} \\ 0_{d,1} & \lambda\tau I_d \end{bmatrix}.$$

Real-world datasets: In what follows, we solve several instances of problem (5.5). We showcase the effectiveness of the proposed approach on 5 regression problems taken from the LIBSVM library (see [10]). Additional information on the datasets is collected in Table 6.

Table 6: Quantile regression datasets.

Name	# of data points	# of features
space_ga	3107	6
abalone	4177	8
cpusmall	8192	12
cadata	20,640	8
E2006-tfidf	16,087	150,360

We fix `tol` $= 10^{-4}$, $\lambda = 10^{-2}$, and $\tau = 0.5$, and run the two methods (active-set (AS), IP-PMM) on the 5 instances for varying quantile level α. The results are collected in Table 7.

Table 7: Quantile regression: varying confidence level (`tol` $= 10^{-4}$, $\lambda = 10^{-2}$, $\tau = 0.5$).

Problem	α	Iterations		Time (s)	
		PMM(SSN)[Fact.]{Krylov}	IP–PMM(Krylov)	AS	IP–PMM
space_ga	0.50	32(36)[0]{623}	18(–)	**0.15**	0.31
	0.65	32(37)[0]{636}	20(–)	**0.13**	0.38
	0.80	30(34)[0]{583}	16(–)	**0.12**	0.25
	0.90	29(31)[0]{536}	22(–)	**0.11**	0.35
abalone	0.50	27(35)[0]{603}	34(–)	**0.17**	1.39
	0.65	27(31)[0]{534}	31(–)	**0.15**	1.13
	0.80	25(30)[0]{492}	22(–)	**0.12**	0.85
	0.90	24(24)[0]{367}	20(–)	**0.09**	0.74
cpusmall	0.50	28(32)[0]{538}	25(–)	**0.26**	2.02
	0.65	28(33)[0]{577}	25(–)	**0.29**	2.01
	0.80	29(30)[0]{518}	22(–)	**0.24**	1.81
	0.90	29(31)[0]{487}	25(–)	**0.23**	2.13
cadata	0.50	4(46)[0]{860}	49(–)	**0.27**	11.40
	0.65	5(49)[0]{933}	42(–)	**0.31**	10.51
	0.80	5(50)[0]{861}	45(–)	**0.30**	10.60
	0.90	29(1405)[214]{5722}	$\doublebarwedge$[1]	**8.95**	$\doublebarwedge$
E2006-tfidf	0.50	37(37)[0]{363}	†[2]	**60.91**	†
	0.65	36(39)[0]{405}	†	**111.66**	†
	0.80	36(39)[0]{399}	†	**118.28**	†
	0.95	35(37)[0]{388}	†	**103.19**	†

[1] $\doublebarwedge$ indicates that the solver incorrectly identified the problem as infeasible.
[2] † indicates that the solver ran out of memory.

From Table 7 we observe that the two approaches are comparable for the smaller instances (space_ga, abalone, and cpusmall). The active-set scheme significantly outperforms IP-PMM on the cadata problem, mainly due to its superior numerical stability and better conditioning of its linear systems. Indeed, this problem is highly ill-conditioned; this is reflected in the increased number of IP-PMM iterations needed to obtain a 4-digit accurate solution. The better conditioning of the linear systems associated with the active-set (compared to those appearing within IP-PMM) can also be observed from the fact that it did not utilize the preconditioner in most instances, enabling it to successfully converge matrix-free. Finally, we observe that for the large-scale instance (E2006-tfidf), both the exact and the inexact variants of IP-PMM crashed due to memory requirements.

Next, we fix `tol` $= 10^{-4}$, and $\alpha = 0.8$, and run the two methods for varying regularization parameters τ and λ. The results are collected in Table 8. We observe that both methods are robust for a wide range of parameter values. As before, the active-set scheme consistently solved the large-scale instance (E2006-tfidf) for a wide range of parameter values, without running into any memory issues.

Table 8: Quantile regression: varying regularization (`tol` $= 10^{-4}$, $\alpha = 0.8$).

Problem	$(\boldsymbol{\tau}, \boldsymbol{\lambda})$	**Iterations**		**Time (s)**	
		PMM(SSN)[Fact.]{Krylov}	IP–PMM(Krylov)	AS	IP–PMM
space_ga	$(0.2, 5 \cdot 10^{-2})$	30(36)[0]{585}	26(–)	**0.13**	0.40
	$(0.4, 1 \cdot 10^{-2})$	30(33)[0]{568}	20(–)	**0.12**	0.32
	$(0.6, 5 \cdot 10^{-3})$	30(33)[0]{654}	18(–)	**0.14**	0.29
	$(0.8, 1 \cdot 10^{-3})$	30(32)[0]{695}	22(–)	**0.12**	0.37
abalone	$(0.2, 5 \cdot 10^{-2})$	26(31)[0]{422}	20(–)	**0.12**	0.76
	$(0.4, 1 \cdot 10^{-2})$	25(30)[0]{485}	27(–)	**0.14**	0.90
	$(0.6, 5 \cdot 10^{-3})$	26(26)[0]{474}	34(–)	**0.12**	1.27
	$(0.8, 1 \cdot 10^{-3})$	26(26)[0]{622}	25(–)	**0.13**	0.74
cpusmall	$(0.2, 5 \cdot 10^{-2})$	28(35)[0]{440}	19(–)	**0.26**	1.58
	$(0.4, 1 \cdot 10^{-2})$	29(29)[0]{487}	17(–)	**0.24**	1.37
	$(0.6, 5 \cdot 10^{-3})$	29(30)[0]{652}	17(–)	**0.28**	1.47
	$(0.8, 1 \cdot 10^{-3})$	29(29)[0]{871}	19(–)	**0.30**	1.54
cadata	$(0.2, 5 \cdot 10^{-2})$	34(940)[265]{4436}	50(–)	**5.93**	11.34
	$(0.4, 1 \cdot 10^{-2})$	6(56)[0]{928}	50(–)	**0.34**	11.55
	$(0.6, 5 \cdot 10^{-3})$	6(69)[0]{1274}	45(–)	**0.40**	10.18
	$(0.8, 1 \cdot 10^{-3})$	12(187)[0]{3621}	33(–)	**1.05**	7.34
E2006-tfidf	$(0.2, 5 \cdot 10^{-2})$	34(43)[0]{406}	†[1]	**61.64**	†
	$(0.4, 1 \cdot 10^{-2})$	36(39)[0]{395}	†	**57.64**	†
	$(0.6, 5 \cdot 10^{-3})$	36(40)[0]{430}	†	**60.10**	†
	$(0.8, 1 \cdot 10^{-3})$	35(36)[0]{447}	†	**84.41**	†

[1] † indicates that the solver ran out of memory.

5.4 Binary classification via linear support vector machines

Finally, we are interested in training a binary linear classifier using regularized soft-margin linear support vector machines (SVMs), [70]. More specifically, given a training dataset $\{(y_i, \xi_i)\}_{i=1}^{l}$, where $y_i \in \{-1, 1\}$ are the *labels* and $\xi_i \in \mathbb{R}^d$ are the *feature vectors* (with d the number of features), we would like to solve the following optimization problem

$$\min_{(\beta_0, \beta) \in \mathbb{R} \times \mathbb{R}^d} \left\{ \frac{1}{l} \sum_{i=1}^{l} \left(1 - y_i \left(\xi_i^\top \beta - \beta_0\right)\right)_+ + \lambda \left(\tau_1 \|\beta\|_1 + \frac{\tau_2}{2} \|\beta\|_2^2\right) \right\}, \tag{5.7}$$

where $\lambda > 0$ is a regularization parameter, and τ_1, $\tau_2 > 0$ are the weights for the ℓ_1 and ℓ_2 regularizers, respectively. The standard Euclidean-norm regularization is traditionally used as a trade-off between the margin of the classifier (the larger the better) and the correct classification of ξ_i, for all $i \in \{1, \ldots, l\}$ (e.g., see [16]). However, this often leads to a dense estimate for β, which has led researchers in the machine learning community into considering the ℓ_1 regularizer instead, in order to encourage sparsity (e.g., see [7]). It is well-known that both regularizers can be combined to deliver the benefits of each of them, using the elastic-net penalty (see, for example, [74]), assuming that τ_1 and τ_2 are appropriately tuned.

Real-world datasets: In what follows we consider elastic-net SVM instances of the form of (5.7). We showcase the effectiveness of the proposed active-set scheme on 3 large-scale binary classification datasets taken from the LIBSVM library ([10]). The problem names, as well as the number of features and training points are collected in Table 9.

Table 9: Binary classification datasets.

Name	# of training points	# of features
rcv1	20,242	47,236
real-sim	72,309	20,958
news20	19,996	1,355,191

In the experiments to follow, we only consider large-scale problems that neither the exact nor the inexact version of IP-PMM can solve on a personal computer, due to excessive memory requirements. Nonetheless, we should note that, by following the developments in [18], IP-PMM can be specialized to problems of this form in order to be able to tackle such large-scale instances. However, the same applies to the proposed active-set method, and thus specialization is avoided for the sake of generality. We fix $\texttt{tol} = 10^{-5}$, $\lambda = 10^{-2}$, and run the active-set solver for all datasets given in Table 9 for varying values of the regularization parameters τ_1, and τ_2. The results are collected in Table 10.

Table 10: Elastic-net SVMs: varying regularization ($\texttt{tol} = 10^{-5}$, $\lambda = 10^{-2}$).

Problem	τ_1	τ_2	Iterations	Time (s)
			PMM(SSN)[Fact.]{Krylov}	AS
rcv1	0.2	0.2	46(64)[0]{1497}	9.28
	0.8	0.2	42(50)[0]{642}	6.90
	0.2	0.8	44(68)[0]{1228}	8.98
	5	5	30(20)[0]{138}	3.74
real-sim	0.2	0.2	57(88)[0]{2382}	34.99
	0.8	0.2	38(22)[0]{180}	10.42
	0.2	0.8	55(84)[0]{1740}	31.21
	5	5	37(26)[0]{168}	10.66
news20	0.2	0.2	36(43)[0]{629}	70.22
	0.8	0.2	29(26)[0]{206}	40.29
	0.2	0.8	31(110)[31]{857}	142.18
	5	5	33(26)[0]{154}	39.94

We observe that the solver was consistently able to solve these instances, without running into memory issues. In this case, the behaviour of the active-set solver was affected by the parameters τ_1 and τ_2 (indeed, see the number of SSN iterations for different regularization values), however, we consistently obtain convergence in a very reasonable amount of time. We observe that no preconditioning was required for most of these problems, except from one instance originating from the news20 dataset (with $\tau_1 = 0.2$ and $\tau_2 = 0.8$). On the one hand, even when the preconditioner was required, the method did not run into any memory issues. On the other hand, we observe that the associated linear systems of the proposed method are relatively well-conditioned, allowing us to avoid the inversion of the preconditioner is many instances, which enables its matrix-free utilization in several settings.

Overall, we observe that the proposed algorithm is very general and can be applied in a plethora of very important applications arising in practice. We were able to showcase that the active-set nature of the method allows one to solve large-scale instances on a personal computer. The proposed method strikes a good balance between first-order methods, which are fast but unreliable, and second-order interior point methods, which are extremely robust but can struggle with the problem size, ill-conditioning, and memory requirements. We have demonstrated that convex quadratic problems with piecewise-linear terms can be solved very efficiently using the proposed active-set scheme, and we conjecture that the algorithm can be readily extended to deal with significantly more general problems, either by appropriately extending its associated theory, or by adjusting its versatile numerical linear algebra kernel.

6 Conclusions

We presented an active-set method for the solution of convex quadratic optimization problems with piecewise-linear terms in the objective, with applications to sparse approximations and risk minimization. After showing the convergence of the proposed algorithm under minimal assumptions, we derived and analyzed a robust numerical linear algebra kernel for the solution of its associated linear systems. We derive preconditioners suitable for Krylov subspace methods that are robust with respect to the penalty parameters of our method and can, in general, pave the way for the widespread utilization of iterative linear algebra in the context of active-set schemes. We were able to numerically demonstrate the efficiency, scalability and robustness of the proposed approach over a wide range of instances arising from risk-averse portfolio selection, L^1-regularized partial differential equation constrained optimization, quantile regression and linear classification via support vector machines, showcasing its competitiveness against alternative state-of-the-art methods. We have demonstrated the generality and versatility of the proposed solver and its associated linear algebra kernel, which make it a very competitive alternative to first- or second-order schemes appearing in the literature.

A Appendix

A.1 Derivation of the dual problem

Assume that notation introduced in Section 2.1. Let $y = [y_1^\top \ y_2^\top]^\top \in \mathbb{R}^{m+l}$. The dual of (P) is $\sup_{y,z} \inf_x \{\ell(x,y,z)\}$. We have

$$\begin{aligned}\inf_x \{\ell(x,y,z)\} &= -\sup_x \left\{\left(F^\top y - z\right)^\top x - f(x)\right\} + y^\top \hat{b} - \delta^*_{\mathcal{K}}(z) - h^*(y_2)\\ &= -f^*(F^\top y - z) + y^\top \hat{b} - \delta^*_{\mathcal{K}}(z) - h^*(y_2)\\ &= -\frac{1}{2}x^\top Q x - \delta_{\{0\}}\left(c + Qx - F^\top y + z\right) + y^\top \hat{b} - \delta^*_{\mathcal{K}}(z) - h^*(y_2),\end{aligned}$$

where we used the definition of the convex conjugate and the definition of $f(\cdot)$, which yields $f^*(F^\top y - z) = \frac{1}{2}x^\top Q x + \delta_{\{0\}}\left(c + Qx - F^\top y + z\right)$. In turn, we observe that this is exactly the dual problem (D).

A.2 Derivation of the augmented Lagrangian

Let $\beta > 0$ be some penalty parameter. The augmented Lagrangian corresponding to (P) reads:

$$\begin{aligned}\mathcal{L}_\beta(x;y,z) \triangleq\ & \sup_{u',v'} \left\{\ell(x,u',v') - \frac{1}{2\beta}\|u'-y\|^2 - \frac{1}{2\beta}\|v'-z\|^2\right\}\\ =\ & f(x) - \inf_{u'}\left\{u'^\top\left(Fx - \hat{b}\right) + h^*(u_{m+1:m+l}) + \frac{1}{2\beta}\|u'-y\|^2\right\}\\ & - \inf_{v'}\left\{-v'^\top x + \delta^*_{\mathcal{K}}(v') + \frac{1}{2\beta}\|v'-z\|^2\right\}\\ =\ & f(x) - y_1^\top(Ax - b) + \frac{\beta}{2}\|Ax - b\|^2\\ & - \inf_{w'}\left\{-w'^\top(Cx+d) + \delta_{\mathcal{C}}(w') + \frac{1}{2\beta}\|w'-y_2\|^2\right\}\\ & - \inf_{v'}\left\{-v'^\top x + \delta^*_{\mathcal{K}}(v') + \frac{1}{2\beta}\|v'-z\|^2\right\}.\end{aligned}$$

where $\mathcal{C} \triangleq \left\{w \in \mathbb{R}^l \middle| w_i \in [0,1],\ j = 1,\ldots,l\right\}$. Let us now solve each of the two minimization problems appearing in the last expression. For the first expression, we use the fact that if $p_1(x) = p_2(x) + r^\top x$, where $p_1,\ p_2$ are two proper closed convex functions, then $\mathbf{prox}_{p_1}(x) = \mathbf{prox}_{p_2}(x - r)$. Thus, we obtain

$$\arg\min_{w'}\left\{-w'^\top(Cx+d) + \delta_{\mathcal{C}}(w') + \frac{1}{2\beta}\|w'-y_2\|^2\right\} = \Pi_{\mathcal{C}}\left(y_2 + \beta(Cx+d)\right).$$

For the second minimization problem, we proceed as before to obtain that

$$\arg\min_{w'} \left\{ -v'^\top x + \delta^*_{\mathcal{K}}(v') + \frac{1}{2\beta}\|v' - z\|^2 \right\} = \mathbf{prox}_{\beta\delta^*_{\mathcal{K}}}\left(z + \beta x\right).$$

Then, by using the *Moreau Identity* (see [44]):

$$\mathbf{prox}_{\beta p}(u') + \beta \mathbf{prox}_{\beta^{-1}p^*}(\beta^{-1}u') = u', \tag{A.1}$$

and by performing some manipulations, we arrive at the final expression for the augmented Lagrangian given in (2.1).

A.3 Differentiability of the augmented Lagrangian

In order to find the gradient of the augmented Lagrangian, we need to use the gradient expression of the Moreau envelope of a given proper closed convex function p. Specifically, given some $\beta > 0$, the Moreau envelope of p reads

$$e_{\beta p}(x) = \inf_{w} \left\{ p(w) + \frac{1}{2\beta}\|w - x\|^2 \right\},$$

and it can be shown that $\nabla e_{\beta p}(x) = \frac{1}{\beta}\left(x - \mathbf{prox}_{\beta p}(x)\right)$. We proceed by computing the gradient of

$$\inf_{w'} \left\{ -w'^\top (Cx + d) + \delta_{\mathcal{C}}(w') + \frac{1}{2\beta}\|w' - y_2\|^2 \right\},$$

noting that the gradient of the second Moreau envelope (implicitly) appearing in the augmented Lagrangian can be computed similarly (using, additionally, the Moreau identity). We notice that

$$\begin{aligned}
&\inf_{w'} \left\{ -w'^\top (Cx + d) + \delta_{\mathcal{C}}(w') + \frac{1}{2\beta}\|w' - y_2\|^2 \right\} \\
&\quad = \inf_{w'} \left\{ \delta_{\mathcal{C}}(w') + \frac{1}{2\beta}\left\|w' - (y_2 + \beta(Cx + d))\right\|^2 \right\} - \frac{\beta}{2}\|Cx + d\|^2 - y_2^\top (Cx + d).
\end{aligned}$$

Thus, using the expression of the gradient of the second (shifted) Moreau envelope alongside the chain rule, we obtain that

$$\begin{aligned}
&\nabla_x \left(\inf_{w'} \left\{ \delta_{\mathcal{C}}(w') + \frac{1}{2\beta}\left\|w' - (y_2 + \beta(Cx + d))\right\|^2 \right\} - \frac{\beta}{2}\|Cx + d\|^2 - y_2^\top (Cx + d) \right) \\
&\quad = -C^\top \Pi_{\mathcal{C}}\left(y_2 + \beta(Cx + d)\right).
\end{aligned}$$

Proceeding in a similar manner for the second Moreau envelope, we arrive at the expression for the gradient of the proximal augmented Lagrangian penalty function given in Section 2.1.

A.4 Termination criteria

Let $\mathcal{C} = \left\{x \in \mathbb{R}^l \vert\ x_i \in [0,1],\ i \in \{1,\ldots,l\}\right\}$. We write the optimality conditions for (P)–(D) as

$$c + Qx - A^\top y_1 + C^\top y_2 + z = 0_n, \qquad Ax = b, \qquad y_2 = \Pi_{\mathcal{C}}\left(y_2 + Cx + d\right), \qquad x = \Pi_{\mathcal{K}}(x + z), \tag{A.2}$$

and the termination criteria for Algorithm PD-PMM (given a tolerance $\epsilon > 0$) are set as

$$\max\left\{ \frac{\|c + Qx - A^\top y_1 + C^\top y_2 + z\|}{1 + \|c\|}, \frac{\left\| \begin{matrix} Ax - b \\ y_2 - \Pi_{\mathcal{C}}(y_2 + Cx + d) \end{matrix} \right\|}{1 + \left\| \begin{matrix} b \\ d \end{matrix} \right\|}, \|x - \Pi_{\mathcal{K}}(x + z)\| \right\} \leq \epsilon. \tag{A.3}$$

References

[1] S. Alexander, T. F. Coleman, and Y. Li, *Minimizing CVaR and VaR for a portfolio of derivatives*, Journal of Banking & Finance, 30 (2006), pp. 583–605, https://doi.org/10.1016/j.jbankfin.2005.04.012.

[2] P. Artzner, F. Delbaen, J.-M. Eber, and D. Heath, *Coherent measures of risk*, Mathematical Finance, 9 (1999), pp. 203–228, https://doi.org/10.1111/1467-9965.00068.

[3] A. Beck, *First-Order Methods in Optimization*, MOS-SIAM Series on Optimization, SIAM & Mathematical Optimization Society, Philadelphia, 2017, https://doi.org/10.1137/1.9781611974997.

[4] L. Bergamaschi, J. Gondzio, A. Martínez, J. W. Pearson, and S. Pougkakiotis, *A new preconditioning approach for an interior point-proximal method of multipliers for linear and convex quadratic programming*, Numerical Linear Algebra with Applications, 28 (2020), p. e2361, https://doi.org/10.1002/nla.2361.

[5] D. P. Bertsekas, A. Nedic, and E. Ozdaglar, *Convex Analysis and Optimization*, Athena Scientific, 2003.

[6] S. Boyd, E. Busseti, S. Diamond, K. R. N., K. Koh, P. Nystrup, and J. Speth, *Multi-period trading via convex optimization*, Foundations and Trends® in Optimization, 3 (2017), pp. 1–76, https://doi.org/10.1561/2400000023.

[7] P. S. Bradley and O. L. Mangasarian, *Feature selection via concave minimization and support vector machines*, in Proceedings of the Fifteenth International Conference on Machine Learning, ICML '98, San Francisco, CA, USA, 1998, Morgan Kaufmann Publishers Inc., pp. 82—90.

[8] R. Bruni, F. Cesarone, A. Scozzari, and F. Tardella, *Real-world datasets for portfolio selection and solutions of some stochastic dominance portfolio models*, Data in Brief, 8 (2016), pp. 858–862, https://doi.org/10.1016/j.dib.2016.06.031.

[9] J. V. Burke and M. C. Ferris, *A Gauss-Newton method for convex composite optimization*, Mathematical Programming, 71 (1995), pp. 179–194, https://doi.org/10.1007/BF01585997.

[10] C.-C. Chang and C.-J. Lin, *LIBSVM: A library for support vector machines*, ACM Transactions on Intelligent Systems and Technology, 2 (2011), pp. 27:1–27:27. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[11] J. Chen and L. Qi, *Globally and superlinearly convergent inexact Newton-Krylov algorithms for solving nonsmooth equations*, Numerical Linear Algebra with Applications, 17 (2010), pp. 155–174, https://doi.org/10.1002/nla.673.

[12] S. S. Chen, D. L. Donoho, and M. A. Saunders, *Atomic decomposition by basis pursuit*, SIAM Review, 43 (2001), pp. 129–159, https://doi.org/10.1137/S003614450037906X.

[13] C. Christof, H. C. De Los Reyes, and C. Meyer, *A nonsmooth trust-region method for locally Lipschitz functions with applications to optimization problems constrained by variational inequalities*, SIAM Journal on Optimization, 30 (2020), pp. 2163–2196, https://doi.org/10.1137/18M1164925.

[14] F. Clarke, *Optimization and Nonsmooth Analysis*, Classics in Applied Mathematics, John Wiley and Sons, New York, 1990, https://doi.org/10.1137/1.9781611971309.

[15] C. Clason and T. Valkonen, *Introduction to Nonsmooth Analysis and Optimization*, arXiv:2001.00216v4, (2023).

[16] C. Cortes and V. N. Vapnik, *Support-vector networks*, Machine Learning, 20 (1995), pp. 273–297, https://doi.org/10.1007/BF00994018.

[17] A. De Marchi, *On a primal-dual Newton proximal method for convex quadratic programs*, Computational Optimization and Applications, (2022), https://doi.org/10.1007/s10589-021-00342-y.

[18] V. De Simone, D. di Serafino, J. Gondzio, S. Pougkakiotis, and M. Viola, *Sparse approximations with interior point methods*, SIAM Review, 64 (2022), pp. 954–988, https://doi.org/10.1137/21M1401103.

[19] W. Deng and W. Yin, *On the global and linear convergence of the generalized alternating direction method of multipliers*, Journal of Scientific Computing, 66 (2016), pp. 889–916, https://doi.org/10.1007/s10915-015-0048-x.

[20] 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), 2017, pp. 2868–2873, https://doi.org/10.1109/CDC.2017.8264075.

[21] A. L. Dontchev and R. T. Rockafellar, *Implicit Functions and Solution Mappings*, Springer Series in Operations Research and Financial Engineering, Springer, New York, NY, 2014, https://doi.org/10.1007/978-1-4939-1037-3.

[22] H. C. Elman, A. Ramage, and D. J. Silvester, *Algorithm 866: IFISS, a Matlab toolbox for modelling incompressible flow*, ACM Transactions on Mathematical Software, 33 (2007), p. 14, https://doi.org/10.1145/1236463.1236469.

[23] H. C. Elman, A. Ramage, and D. J. Silvester, *IFISS: A computational laboratory for investigating incompressible flow problems*, SIAM Review, 52 (2014), pp. 261–273, https://doi.org/10.1137/120891393.

[24] K. Fountoulakis, J. Gondzio, and P. Zhlobich, *Matrix-free interior point method for compressed sensing problems*, Mathematical Programming Computation, 6 (2014), pp. 1–31, https://doi.org/10.1007/s12532-013-0063-6.

[25] E. Gilbert and L. Meiklejohn, *A comparative analysis of risk measures: A portfolio optimisation approach*, Investment Analysts Journal, 48 (2019), https://doi.org/10.1080/10293523.2019.1643128.

[26] P. E. Gill and D. P. Robinson, *A primal–dual augmented Lagrangian*, Computational Optimization and Applications, 15 (2012), pp. 1–25, https://doi.org/10.1007/s10589-010-9339-1.

[27] J. Gondzio, S. Pougkakiotis, and J. W. Pearson, *General-purpose preconditioning for regularized interior point methods*, Computational Optimization and Applications, 83 (2022), pp. 727–757, https://doi.org/10.1007/s10589-022-00424-5.

[28] S.-P. Han, J.-S. Pang, and N. Rangaraj, *Globally convergent Newton methods for nonsmooth equations*, Mathematics of Operations Research, 17 (1992), pp. 586–607, https://doi.org/10.1287/moor.17.3.586.

[29] E. Hans and T. Raasch, *Global convergence of damped semismooth Newton methods for ℓ_1 Tikhonov regularization*, Inverse Problems, 31 (2015), p. 025005, https://doi.org/10.1088/0266-5611/31/2/025005.

[30] B. Hermans, A. Themelis, and P. Patrinos, *QPALM: a proximal augmented Lagrangian method for nonconvex quadratic programs*, Mathematical Programming Computation, 14 (2022), pp. 497–541, https://doi.org/10.1007/s12532-022-00218-0.

[31] K. Ito and K. Kunnisch, *On a semi-smooth Newton method and its globalization*, Mathematical Programming, 118 (2009), pp. 347–370, https://doi.org/10.1007/s10107-007-0196-3.

[32] P. Jorion, *Risk management lessons from long-term capital management*, European Financial Management, 6 (2000), pp. 277–300, https://doi.org/10.1111/1468-036X.00125.

[33] H. Konno and H. Yamazaki, *Mean-absolute deviation portfolio optimization model and its applications to Tokyo stock market*, Management Science, 37 (1991), pp. 519–531, https://doi.org/10.1287/mnsc.37.5.519.

[34] P. Krokhmal, S. Uryasev, and J. Palmquist, *Portfolio optimization with conditional value-at-risk objective and constraints*, Journal of Risk, 4 (2002), pp. 43–68, https://doi.org/10.21314/JOR.2002.057.

[35] R. Lazimy, *Mixed-integer quadratic programming*, Mathematical Programming, 22 (1982), pp. 332–349, https://doi.org/10.1007/BF01581047.

[36] J. D. Lee, Y. Sun, and M. A. Saunders, *Proximal Newton-type methods for minimizing composite functions*, SIAM Journal on Optimization, 24 (2014), pp. 1420–1443, https://doi.org/10.1137/130921428.

[37] D. Li and W.-L. Ng, *Optimal dynamic portfolio selection: Multiperiod mean-variance formulation*, Mathematical Finance, 10 (2000), pp. 387–406, https://doi.org/10.1111/1467-9965.00100.

[38] X. Li, D. Sun, and K. C. Toh, *A highly efficient semismooth Newton augmented Lagrangian method for solving Lasso problems*, SIAM Journal on Optimization, 28 (2018), pp. 433–458, https://doi.org/10.1137/16M1097572.

[39] X. Li, D. Sun, and K. C. Toh, *An asymptotically superilinearly convergent semismooth Newton augmented Lagrangian method for linear programming*, SIAM Journal on Optimization, 30 (2020), pp. 2410–2440, https://doi.org/10.1137/19M1251795.

[40] K. T. Lwin, R. Qu, and B. L. MacCarthy, *Mean-VaR portfolio optimization: A nonparametric approach*, European Journal of Operational Research, 260 (2017), pp. 751–766, https://doi.org/10.1016/j.ejor.2017.01.005.

[41] F. Mannel and A. Rund, *A hybrid semismooth quasi-Newton method for nonsmooth optimal control with PDEs*, Optimization and Engineering, 22 (2021), pp. 2087–2125, https://doi.org/10.1007/s11081-020-09523-w.

[42] H. M. Markowitz, *Portfolio selection*, The Journal of Finance, 7 (1952), pp. 77–91, https://doi.org/10.2307/2975974.

[43] J. Martínez and L. Qi, *Inexact Newton methods for solving nonsmooth equations*, Journal of Computational and Applied Mathematics, 60 (1995), pp. 127–145, https://doi.org/10.1016/0377-0427(94)00088-I.

[44] J. J. Moreau, *Proximité et dualité dans un espace Hilbertien*, Bulletin de la Société Mathématique de France, 93 (1965), pp. 273–299, https://doi.org/10.24033/bsmf.1625.

[45] C. C. Paige and M. A. Saunders, *Solution of sparse indefinite systems of linear equations*, SIAM Journal on Numerical Analysis, 12 (1975), pp. 617–629, https://doi.org/10.1137/0712047.

[46] P. Patrinos and A. Bemporad, *Proximal Newton methods for convex composite optimization*, in 52nd IEEE Conference on Decision and Control, 2013, pp. 2358–2363, https://doi.org/10.1109/CDC.2013.6760233.

[47] J. W. Pearson, M. Porcelli, and M. Stoll, *Interior-point methods and preconditioning for PDE-constrained optimization problems involving sparsity terms*, Numerical Linear Algebra with Applications, 27 (2019), p. e2276, https://doi.org/10.1002/nla.2276.

[48] M. Porcelli, V. Simoncini, and M. Stoll, *Preconditioning PDE-constrained optimization with l^1-sparsity and control constraints*, Computers & Mathematics with Applications, 74 (2017), pp. 1059–1075, https://doi.org/10.1016/j.camwa.2017.04.033.

[49] S. Pougkakiotis and J. Gondzio, *An interior point-proximal method of multipliers for convex quadratic programming*, Computational Optimization and Applications, 78 (2021), pp. 307–351, https://doi.org/10.1007/s10589-020-00240-9.

[50] S. Pougkakiotis, J. Gondzio, and D. Kalogerias, *An active-set method for sparse approximations. Part I: Separable ℓ_1 terms*, arXiv:2201.10211v2, (2023).

[51] S. Pougkakiotis, J. Gondzio, and D. Kalogerias, *An active-set method for sparse approximations. Part II: General piecewise-linear terms*, arXiv:2302.14497v1, (2023).

[52] H. Ramos, M. Righi, C. Guedes, and F. Müller, *A comparison of risk measures for portfolio optimization with cardinality constraints*, 2022, http://dx.doi.org/10.2139/ssrn.4141301.

[53] M. B. Righi and D. Borenstein, *A simulation comparison of risk measures for portfolio optimization*, Finance Research Letters, 24 (2018), pp. 105–112, https://doi.org/10.1016/j.frl.2017.07.013.

[54] S. M. Robinson, *Some continuity properties of polyhedral multifunctions*, in Mathematical Programming at Oberwolfach, H. König, B. Korte, and K. Ritter, eds., vol. 14 of Mathematical Programming Studies, Springer, Berlin, Heidelberg, 1981, pp. 206–214, https://doi.org/10.1007/BFb0120929.

[55] R. T. Rockafellar, *Augmented Lagrangians and applications of the proximal point algorithm in convex programming*, Mathematics of Operations Research, 1 (1976), pp. 97–116, https://doi.org/doi.org/10.1287/moor.1.2.97.

[56] R. T. Rockafellar, *Monotone operators and the proximal point algorithm*, SIAM Journal on Control and Optimization, 14 (1976), pp. 877–898, https://doi.org/10.1137/0314056.

[57] R. T. Rockafellar, *Generalizations of the proximal method of multipliers in convex optimization*, Computational Optimization and Applications, 87 (2024), pp. 219–247.

[58] R. T. Rockafellar and S. Uryasev, *Optimization of conditional value-at-risk*, Journal of Risk, 2 (2000), pp. 21–41, https://doi.org/10.21314/JOR.2000.038.

[59] R. T. Rockafellar, S. Uryasev, and M. Zabarankin, *Optimality conditions in portfolio analysis with general deviation measures*, Mathematical Programming, 108 (2006), pp. 515–540, https://doi.org/10.1007/s10107-006-0721-9.

[60] R. T. Rockafellar and R. J. B. Wets, *Variational Analysis*, vol. 317 of Grundlehren der mathematischen Wissenschaften, Springer-Verlag Berlin Heidelberg, 1998, https://doi.org/10.1007/978-3-642-02431-3.

[61] A. Shapiro, D. Dentcheva, and A. Ruszczyński, *Lectures on Stochastic Programming: Modeling and Theory*, MOS-SIAM Series on Optimization, SIAM & Mathematical Optimization Society, Philadelphia, 2009, https://doi.org/10.1137/1.9780898718751.

[62] L. P. Silva, D. Alem, and F. L. Carvalho, *Portfolio optimization using mean absolute deviation (MAD) and conditional value-at-risk (CVaR)*, Production, 27 (2017), https://doi.org/10.1590/0103-6513.208816.

[63] X. Song, B. Chen, and B. Yu, *An efficient duality-based approach for PDE-constrained sparse optimization*, Computational Optimization and Applications, 69 (2018), pp. 461–500, https://doi.org/10.1007/s10589-017-9951-4.

[64] L. Stella, A. Themelis, and P. Patrinos, *Forward-backward quasi-Newton methods for nonsmooth optimization problems*, Computational Optimization and Applications, 67 (2017), pp. 443–487, https://doi.org/10.1007/s10589-017-9912-y.

[65] B. Stellato, G. Banjac, P. Goulart, A. Bemporad, and S. Boyd, *OSQP: an operator splitting solver for quadratic programs*, Mathematical Programming Computation, 12 (2020), pp. 637–672, https://doi.org/10.1007/s12532-020-00179-2, https://doi.org/10.1007/s12532-020-00179-2.

[66] A. Themelis, L. Stella, and P. Patrinos, *Forward-backward envelope for the sum of two nonconvex functions: Further properties and nonmonotone linesearch algorithms*, SIAM Journal on Optimization, 28 (2018), pp. 2274–2303, https://doi.org/10.1137/16M1080240.

[67] F. Tröltzsch, *Optimal Control of Partial Differential Equations: Theory, Methods and Applications*, vol. 112 of Graduate Studies in Mathematics, American Mathematical Society, 2010, https://doi.org/10.1090/gsm/112.

[68] R. J. Vanderbei, *Symmetric quasidefinite matrices*, SIAM Journal on Optimization, 5 (1993), pp. 100–113, https://doi.org/10.1137/0805005.

[69] V. N. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, New York, 1998.

[70] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Information Science and Statistics, Springer, New York, NY, 2000, https://doi.org/10.1007/978-1-4757-3264-1.

[71] G. Wachsmuth and D. Wachsmuth, *Convergence and regularization results for optimal control problems with sparsity functional*, ESAIM: Control, Optimisation and Calculus of Variations, 17 (2011), pp. 858–886, https://doi.org/10.1051/cocv/2010027.

[72] R. A. Waltz, J. L. Morales, J. Nocedal, and D. Orban, *An interior algorithm for nonlinear optimization that combines line search and trust region steps*, Mathematical Programming, 107 (2006), pp. 391–408, https://doi.org/10.1007/s10107-004-0560-5.

[73] C. Wang and P. Tang, *A dual semismooth Newton based augmented Lagrangian method for large-scale linearly constrained sparse group square-root Lasso problems*, Journal of Scientific Computing, 96 (2023), https://doi.org/10.1007/s10915-023-02271-w.

[74] L. Wang, J. Zhu, and H. Zou, *The doubly regularized support vector machine*, Statistica Sinica, 16 (2006), pp. 589–615, http://www.jstor.org/stable/24307560.

[75] X. Y. Zhao, D. F. Sun, and K. C. Toh, *A Newton-CG augmented Lagrangian method for semidefinite programming*, SIAM Journal on Optimization, 20 (2010), pp. 1737–1765, https://doi.org/10.1137/080718206.

[76] H. Zou and T. Hastie, *Regularization and variable selection via the elastic net*, Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67 (2005), pp. 301–320, https://doi.org/10.1111/j.1467-9868.2005.00503.x.