GMRES-Accelerated ADMM for Quadratic Objectives

Richard Y. Zhang† Jacob K. White‡ April 3, 2017

Abstract

We consider the sequence acceleration problem for the alternating direction method-of-multipliers (ADMM) applied to a class of equality-constrained problems with strongly convex quadratic objectives, which frequently arise as the Newton subproblem of interior-point methods. Within this context, the ADMM update equations are linear, the iterates are confined within a Krylov subspace, and the General Minimum RESidual (GMRES) algorithm is optimal in its ability to accelerate convergence. The basic ADMM method solves a $\kappa$-conditioned problem in $O(\sqrt{\kappa})$ iterations. We give theoretical justification and numerical evidence that the GMRES-accelerated variant consistently solves the same problem in $O(\kappa^{1/4})$ iterations, for an order-of-magnitude reduction in iterations, despite a worst-case bound of $O(\sqrt{\kappa})$ iterations. The method is shown to be competitive against standard preconditioned Krylov subspace methods for saddle-point problems. The method is embedded within SeDuMi, and used to solve many large-scale semidefinite programs with error that decreases like $O(1/k^2)$, instead of $O(1/k)$, where $k$ is the iteration index.

1 Introduction

The alternating direction method-of-multipliers (ADMM) [25, 22] is a popular first-order optimization algorithm, used to establish consensus between many local subproblems and a global master problem. This sort of decomposable problem structure naturally arises over a wide range of applications, from statistics and machine learning, to the solution of semidefinite programs; see [9] for an overview. Part of the appeal of ADMM is that it is simple and easy to implement at a large scale, and that convergence is guaranteed under very mild assumptions. On the other hand, ADMM can converge very slowly, sometimes requiring thousands of iterations to compute solutions accurate to just 2-3 digits.

Sequence acceleration—the idea of extrapolating information collected in past iterates—can improve the convergence rates of first-order methods, thereby allowing them to compute more accurate solutions. Within this context, the technique known as momentum acceleration has been particularly successful. Originally developed by Nesterov for the basic gradient method [36], the technique has since been extended to a wide range of first-order methods [38, 2, 3]. In each case, the momentum-accelerated variant improves upon the basic first-order method by an order of magnitude, achieving a convergence rate that is provably optimal.

---

†Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 (ryz@alum.mit.edu). Current address: Department of Industrial Engineering and Operations Research, University of California, Berkeley, CA 94720 (ryz@berkeley.edu).
‡Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, MA 02139 (white@mit.edu).

*This work was supported in part by the Skolkovo-MIT initiative in Computational Mathematics, a DARPA YFA Award, an ONR YIP Award, and an AFOSR YIP Award.
Unfortunately, sequence acceleration has been less successful for ADMM. Momentum-accelerated ADMM schemes have been proposed [26, 45, 41, 30], but the reductions in iterations have been modest, and limited to special cases where the objectives are assumed to be strongly convex and/or quadratic. Part of the problem is that ADMM is a difficult method to analyze, even in the simple quadratic case. The most widely used technique to accelerate ADMM in practice is simple over-relaxation, which reliably reduces total iteration count by a small constant [58, 24, 23, 39]. It remains unclear whether an order-of-magnitude acceleration is even achievable for ADMM in the first place.

1.1 Sequence acceleration using GMRES

In this paper, we restrict our attention to the ADMM solution of the equality-constrained quadratic program

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} x^T D x + c^T x + p^T z \\
\text{subject to} & \quad Ax + Bz = d,
\end{align*}
\]

(ECQP)

which frequently arises as the Newton subproblem of interior-point methods. The Karush–Kuhn–Tucker (KKT) equations for (ECQP) are linear, so first-order methods applied to (ECQP) reduce to a matrix iteration of the form

\[
u^{k+1} = G(\beta)u^k + b(\beta),
\]

where \(u = [x; z; y]\) collects the primal-dual variables, and \(\beta > 0\) is the ADMM quadratic-penalty / step-size parameter.

Sequence acceleration for matrix iterations is a well-studied topic. While the optimal acceleration scheme is rarely available as an analytical expression, it is attained numerically by the Generalized Minimum RESidual (GMRES) algorithm [50]; we review this point in detail in Section 2. Using GMRES to accelerate a matrix iteration, the resulting iterates are guaranteed to converge as fast or faster (viewed under a particular metric) than any sequence acceleration scheme, based on over-relaxation, momentum, or otherwise. Hence, at a minimum, we may use GMRES to numerically bound the amount of acceleration available using ADMM or its variants. If convergence is sufficiently rapid, then the accelerated method (described as Algorithm 1 in Section 2, which we name ADMM-GMRES) may be also be used as a solution algorithm for (ECQP).

1.2 Convergence in \(O(\kappa^{1/4})\) iterations

Under the strong convexity assumptions described in [14], ADMM converges at a linear rate, with a convergence rate dependent on the parameter choice \(\beta\) and the problem condition number \(\kappa\). A number of previous authors have derived the optimal fixed parameter choice \(\beta^*\) that allows ADMM to converge to an \(\epsilon\)-accurate solution in \(O(\sqrt{\kappa} \log \epsilon^{-1})\) iterations [23, 24, 39].

Numerically accelerating this sequence using GMRES, we were surprised to find that an \(\epsilon\)-accurate solution is consistently computed in \(O(\kappa^{1/4} \log \epsilon^{-1})\) iterations, for a square-root factor reduction in the number of iterations. Figure 1 makes this comparison for 1000 instances of (ECQP) randomly generated using Algorithm 2 in Section 5.2 below. The same acceleration was also reliably observed for a wide collection of interior-point Newton subproblems in Section 8.

The order-of-magnitude acceleration is surprising because the \(O(\sqrt{\kappa} \log \epsilon^{-1})\) iteration bound is sharp, when considered over all \(\kappa\)-conditioned (ECQP). Indeed, we give an explicit example in Section 4 and prove that ADMM-GMRES cannot converge for this example at a rate faster than
Figure 1: Given the same 1000 randomly-generated problems and using the same parameter choice, ADMM (circles) converges in $O(\sqrt{\kappa})$ iterations while GMRES-accelerated ADMM (crosses) converges in $O(\kappa^{1/4})$ iterations, where $\kappa$ is the condition number.

$(\sqrt{\kappa} - 1)/(\sqrt{\kappa} + 1)$. The same sort of $\sqrt{\kappa}$ lower complexity bounds are well-known for gradient methods and their proximal / projected variants; see [35] and also [37, 2].

Nevertheless, these numerical results immediately confirm the possibility for an order-of-magnitude acceleration for ADMM under practical settings, using over-relaxation, momentum, or some other scheme. In fact, an optimal over-relaxation scheme with $k$ parameters may be extracted from $k$ iterations of GMRES and explicitly applied; see [34] and the references therein. Also, GMRES-accelerated ADMM may be used as a direct solution method for (ECQP). In this case, the $O(k^2)$ time and $O(k)$ memory requirements of GMRES should be balanced against the cost of $k$ iterations of ADMM. For very ill-conditioned problems needing many iterations, it is possible to periodically restart ADMM-GMRES, although its optimality is no longer guaranteed. We discuss these practical considerations in Section 2.4.

1.3 Main Results

Our primary goal in this paper is to understand why GMRES so frequently achieves an order-of-magnitude acceleration for ADMM. Convergence analysis for Krylov subspace methods is typically formulated as a classic problem in approximation theory—how well can one approximate the eigenvalues of the matrix $I - G(\beta)$ as the roots of an order-$k$ polynomial? In Section 5 we show that the $\kappa^{1/4}$ factor arises because the real eigenvalues of the matrix $I - G(\beta^*)$ lie along (a rescaled version of) the interval $[1, \kappa^{1/4}]$, and the Chebyshev polynomial approximates this entire interval with convergence factor $(\kappa^{1/4} - 1)/(\kappa^{1/4} + 1)$. This is precisely the same mechanism that grants conjugate gradients a square-root factor acceleration over basic gradient descent; see [28, Ch.3] and [49, Ch.6.11].
Within the context of ADMM-GMRES, however, the square-root factor acceleration hinges on two additional assumptions. First, the nonsymmetric iteration matrix $G(\beta)$ should be close to normal, in order for its behavior to be accurately captured by its eigenvalues. Furthermore, $G(\beta)$ also contains complex outliers: eigenvalues with nonzero imaginary parts, which prevent the Chebyshev polynomial from being directly applicable. In Section 5.1, we prove that the $\kappa^{1/4}$ factor will persist if the complex outliers are better conditioned than the real eigenvalues.

Our extensive numerical trials found both assumptions to be generic properties of (ECQP), holding for almost all problem instances. As a consequence, convergence in $O(\kappa^{1/4})$ is observed to be generic property for the ADMM-GMRES solution of (ECQP). Although it is difficult to derive rigorous bounds, we make a number of heuristic arguments in support of this observation in Section 5.2 and Section 6.

It remains an open question whether the order-of-magnitude acceleration would persist for non-quadratic objectives. If the update equations are nonlinear, then ADMM is no longer a matrix iteration, and GMRES acceleration is no longer optimal. Nevertheless, the update equations may be locally well-approximated by their linearization, and a nonlinear version of GMRES like a Newton-Krylov method [11] or Anderson acceleration [57] may prove to be useful.

1.4 View as a preconditioner

Much of our analysis is based on the observation that ADMM reduces to a preconditioner for an augmented Lagrangian version of the Karush–Kuhn–Tucker (KKT) equations for (ECQP). Within this context, ADMM-GMRES is only one of numerous preconditioned Krylov subspace methods available; see [6] for a comprehensive survey. Indeed, it is closely related to block-triangular [10] [61] [51] and augmented Lagrangian / Uzawa preconditioners [21] [27].

In Section 7, we compare ADMM-GMRES against some classic preconditioners for saddle-point problems, including the block-diagonal preconditioner, variants of the constraint preconditioner, and the Hermetian / Skew-Hermitian splitting preconditioner. Restricting each preconditioner to the same operations used in ADMM, we find that each preconditioner regularly attains its worst-case iteration bound of $O(\sqrt{\kappa})$. By comparison, the ADMM preconditioner converges in $O(\kappa^{1/4})$ iterations for every problem considered. The order-of-magnitude reduction in the number of iterations was able to offset both the higher cost of the preconditioner, as well as the need to deploy an expensive Krylov method like GMRES.

1.5 Applications in large-scale semidefinite programming

Our original motivation for this paper is the solution of large-scale semidefinite programs

$$\min_X \{ \text{tr} CX : \text{tr} A_i X = b_i \forall i \in \{1, \ldots, m\}, \ X \succeq 0 \} \quad \text{(SDP)}$$

$$\max_{y, S} \{ b^T y : \sum_{i=1}^m y_i A_i + S = C, \ S \succeq 0 \} \quad \text{(SDD)}$$

where each matrix is $n \times n$ real symmetric, and $X \succeq 0$ indicates that $X$ is symmetric positive semidefinite. The standard approach is to use an interior-point method, which transforms (SDD) into a sequence of (ECQP), each written

$$\max b^T y - \frac{1}{2} \| W^{1/2} (S - Z) W^{1/2} \|_F^2 \quad \text{(NEWT)}$$

subject to

$$\sum_{i=1}^m y_i A_i + S = C.$$
The method usually converges in a few tens of iterations. Each iteration solves \(\text{NEWT}\) via its dense KKT equations in \(O(N^3)\) time and \(O(N^2)\) memory, where \(N = n(n+1) + m\) is the total number of primal-dual variables.

ADMM can be applied to \(\text{SDP} - \text{SDD}\), and each iteration can be performed in just \(O(N^{3/2})\) time and \(O(N)\) memory, assuming a certain sparsity structure in the data \[58, 40\]. For many large-scale SDPs, ADMM converges at a linear rate—like an interior-point method—to 6+ digits of precision in just a few hundred iterations \[58\]. For many others, particularly those that arise from the chordal decomposition of sparse SDPs, the method converges in feasibility and duality gap at the sublinear rate of \(O(1/k)\) at the \(k\)-th iteration; thousands of iterations are required to obtain solutions of only modest accuracy \[31, 33, 40, 60\].

Alternatively, ADMM can be applied to \(\text{NEWT}\), as a set of inner iterations within an interior-point solution of \(\text{SDP} - \text{SDD}\) \[42, 43\]. Indeed, each inner ADMM iteration has roughly the same cost as before: \(O(N^{3/2})\) flops and \(O(N)\) memory, assuming the same sparsity structure in the data. However, the worst-case convergence rate is not improved. The condition number associated with \(\text{NEWT}\) scales as \(\kappa = \Theta(1/\epsilon^2)\) where \(\epsilon\) is the current duality gap. Amortizing \(O(\sqrt{\kappa})\) ADMM iterations over each outer interior-point iteration again yields a sublinear rate of \(O(1/k)\) at the \(k\)-th ADMM iteration.

Now, suppose that ADMM-GMRES is able to converge in \(O(\kappa^{1/4})\) iterations. Then, applying the method to \(\text{NEWT}\) yields an interior-point method with an improved sublinear rate of \(O(1/k^2)\) at the \(k\)-th ADMM-GMRES iteration. In Section 8 we embed ADMM-GMRES within SeDuMi \[53\], and use it to solve problems from SDPLIB \[8\] and the Seventh DIMACS Implementation Challenge \[44\]. ADMM-GMRES solves every instance of \(\text{NEWT}\) in \(O(\kappa^{1/4})\) iterations, although the costs associated with GMRES become too high past \(\approx 30\) iterations. Restarting ADMM-GMRES every 25 iterations yielded a modified interior-point method with an effective error rate of \(O(1/k^2)\) for 8 of the 10 DIMACS problems considered.

1.6 Definitions and notation

Throughout this paper, we will frequently refer to the condition number \(\kappa\), the gradient Lipschitz constant \(L\), the strong convexity parameter \(\mu\), which are defined in terms of a rescaled objective matrix \(\tilde{D}\) in \[7\]. The variable \(k\) is reserved for the iteration index.

From Section 2 to Section 7 the positive integers \(n, \ell, m\) refer to the dimensions of the primal-dual variables \(x, z, y\), and \(N = n + \ell + m\) refers to the total number of primal-dual variables. However, in discussing interior-point methods for SDPs in Section 8 (and previously in Section 1.5), we redefine \(n\) and \(m\) as the order of the semidefinite cone and the dimension of the dual vector, in order to concur with standard notation in this area. We again use \(N = n(n+1) + m\) to refer to the total number of primal-dual variables.

Our notation is otherwise standard, with the following exceptions. We use \(\Lambda\{A\}\) to refer to the eigenvalues of \(A\). The set \(\mathbb{P}_k\) denotes the space of order-\(k\) polynomials. Given a polynomial \(p(z)\), we denote its maximum modulus over a compact subset of the complex plane \(S \subset \mathbb{C}\) as \(\|p(z)\|_S = \max_{z \in S} |p(z)|\).
2 ADMM for Quadratic Problems

Beginning with a choice of the quadratic-penalty / step-size parameter $\beta > 0$ and initial points $u^0 = \{x^0, z^0, y^0\}$, ADMM generates iterates

$$x^{k+1} = \arg \min_x \frac{1}{2} x^T D x + c^T x + \frac{\beta}{2} \| Ax + B z - c + y^k \|^2, \quad (2a)$$

$$z^{k+1} = \arg \min_z p^T z + \frac{\beta}{2} \| Ax^{k+1} + B z - c + y^k \|^2, \quad (2b)$$

$$y^{k+1} = y^k + (Ax^{k+1} + Bz^{k+1} - c). \quad (2c)$$

Some basic algebraic manipulations reveal these to be a matrix-splitting iteration for the (rescaled) augmented Lagrangian KKT system (see e.g. [21, 27])

$$\begin{bmatrix}
\beta^{-1} D + A^T A & A^T B & A^T \\
B^T A & B^T B & B^T \\
A & B & 0
\end{bmatrix}
\begin{bmatrix}
x \\
z \\
y_k
\end{bmatrix}
= \begin{bmatrix}
A^T d - \beta^{-1} c \\
B^T d - \beta^{-1} p \\
d
\end{bmatrix}, \quad (3)$$

using a Gauss-Seidel–like splitting

$$H(\beta) = \begin{bmatrix}
\beta^{-1} D + A^T A & 0 & 0 \\
B^T A & B^T B & 0 \\
A & B & -I
\end{bmatrix} - \begin{bmatrix}
0 & -A^T B & -A^T \\
0 & 0 & -B^T \\
0 & 0 & -I
\end{bmatrix} = M(\beta) - N(\beta). \quad (4)$$

Indeed, fixing the parameter $\beta$, (2) is precisely the linear fixed-point iterations

$$u^{k+1} = M(\beta)^{-1} N(\beta) u^k + M(\beta)^{-1} v(\beta). \quad (5)$$

The dominant cost at each ADMM iteration is a single matrix-vector product with each of $(\beta^{-1} D + A^T A)^{-1}, (B^T B)^{-1}, A, B, A^T$ and $B^T$; the method can be effective only if these operations can be performed at reduced cost. In practice, efficient matrix-vector products usually arise from problem-specific structure. For example, when the matrices $A$, $B$, and $D$ are sparse, the matrices $\beta^{-1} D + A^T A$ and $B^T B$ often admit sparse Cholesky factorizations; see the discussion in [21, Sec.4.2]. Alternatively, the matrices may admit efficient representations, such as a Kronecker factorization (e.g. $D = W \otimes W$; see our discussion in Section 8 for the SDP Newton subproblem) that allow data-sparse representations of their inverses to be constructed.

2.1 Sequence acceleration

Let us view ADMM as a black box $T_\beta(\cdot)$ that maps a given test point $u$ to its image $T_\beta(u)$, in order to rewrite the basic ADMM method as the iterated map

$$u^{k+1} = T_\beta(u^k). \quad (6)$$

Under strong convexity assumptions, (6) converges linearly to a unique fixed-point [14].

Assumption 1. The matrix $D$ is symmetric positive definite, the matrix $A$ has full row-rank (i.e. $AA^T$ is invertible), and the matrix $B$ has full column-rank (i.e. $B^T B$ is invertible).
The convergence rate of \[ (6) \] depends on parameter choice $\beta$ and the problem condition number $\kappa$, defined as the ratio of a gradient Lipschitz parameter $L$ and a strong convexity parameter $\mu$,

$$
\bar{D} \triangleq (AD^{-1}A^T)^{-1}, \quad \mu \triangleq \lambda_{\min}(\bar{D}), \quad L \triangleq \lambda_{\max}(\bar{D}), \quad \kappa \triangleq L/\mu.
$$

In particular, the parameter choice $\beta = \sqrt{\mu L}$ allows an $\epsilon$-accurate solution to be computed in no more than $O(\sqrt{\kappa} \log \epsilon^{-1})$ iterations \[23, 24, 39\].

Sequence acceleration seeks to find an $\epsilon$-accurate approximation of the fixed point $u^* = T_\beta(u^*)$ while making as few calls to the black-box oracle as possible. Two popular approaches are over-relaxation, which linearly extrapolates the current step,

$$
u^{k+1} = u^k + \omega_k[T_\beta(u^k) - u^k],
$$

and momentum, which linearly extrapolates the previous step,

$$
\bar{u} = u^k + \theta_k(u^k - u^{k-1}), \quad \nu^{k+1} = T_\beta(\bar{u}).
$$

In each case, the $(k+1)$-th iterate is selected from the plane that crosses the initial point and the $k$ images

$$
u^{k+1} \in \text{Aff}\{u^0, T_\beta(u^0), T_\beta(u^1), \ldots, T_\beta(u^k)\}.
$$

The affine hull linearly extrapolates the information collected from $k$ evaluations of the black-box oracle; the parameters $\theta_1, \ldots, \theta_k$ and $\omega_1, \ldots, \omega_k$ may be viewed as its coordinates. By carefully tuning these parameters, it is possible to select better candidates than the default choice produced by the iterated map \[6\], thereby yielding a sequence $u^0, u^1, \ldots, u^k$ with an accelerated convergence rate.

##### 2.2 The optimality of GMRES

Let the black-box oracle $T_\beta(\cdot)$ be affine, meaning that there exists some matrix $G(\beta)$ and vector $b(\beta)$ such that $T_\beta(u) = G(\beta)u + b(\beta)$. Furthermore, let us measure the accuracy of a test point $u$ using an implicit but easily computable Euclidean metric\[4\]

$$
\|u - u^*\|_M \triangleq \|u - [G(\beta)u + b(\beta)]\| = \|(I - G(\beta))(u - u^*)\|.
$$

Then, the affine search space \[10\] reduces to a Krylov subspace

$$
u^{k+1} \in u^0 + \text{span}\{r, Gr, \ldots, G^k r\}
$$

where $G \equiv G(\beta)$ and $r = u^0 - [G(\beta)u^0 + b(\beta)]$, and the problem of selecting the best candidate from the Krylov subspace \[12\] is numerically solved by GMRES. We defer to standard texts for its implementation details, e.g. \[49, \text{Alg. 6.9}\] or \[49, \text{Alg. 6.10}\], and only note that the algorithm can be viewed as a “black-box” that solves the following projected least-squares problem at the $k$-th iteration:

$$
\text{GMRES}_k(A, b) = \arg \min \left\{ \|b - Ax\|_2 : x \in \text{span}\{b, Ab, \ldots, A^{k-1}b\} \right\},
$$

in $\Theta(k^2 N)$ flops, $\Theta(k N)$ memory, and $k$ matrix-vector products with $A$. Consider the following algorithm.

\[\text{Indeed, } \kappa_M^{-1} \leq \|u - u^*\|_M/\|u - u^*\| \leq \kappa_M, \text{ where } \kappa_M = \text{cond}(I - G(\beta)) \text{ is finite because the ADMM iterations converge to a unique fixed-point.}\]
Algorithm 1 (ADMM-GMRES). Input: The update operator $T_\beta$ that implements (2) as $u^{k+1} = T_\beta(u^k)$; Initial point $u^0$; Number of iterations $k$.

1. Precompute and store $T_\beta(u^0) = G(\beta)u^0 + b(\beta)$ and $r = u^0 - T_\beta(u^0)$;
2. Call $\Delta u = \text{GMRES}_k(I - G(\beta), r)$, while evaluating each matrix-vector product as $[I - G(\beta)]h = h - [T_\beta(u_0 + h) - T_\beta(u^0)]$;
3. Output $u^k = u^0 - \Delta u$.

The optimality of GMRES in (13) guarantees $\|u^k - u^*\|_M$ to be smaller than that of regular ADMM, as well as any accelerated variant that selects its $k$-th iterate $u^k$ from (10). In other words, no linearly extrapolating sequence acceleration scheme, based on momentum, over-relaxation, or otherwise, can converge faster than GMRES when viewed under this metric.

2.3 ADMM as a preconditioner

The fixed-point equation associated with the ADMM iterations (5)

$$u^* - G(\beta)u^* = b(\beta),$$

is a linear system of equations when $\beta$ is held fixed, which can be solved using GMRES. In the previous subsection, we named the resulting method ADMM-GMRES, and viewed it as an optimally accelerated version of ADMM. Equivalently, (14) is also the left-preconditioned system of equations

$$M^{-1}(\beta)[H(\beta)u^* - b(\beta)] = 0 \iff (14),$$

where $H$ and $v$ comprise the augmented Lagrangian KKT system in (3), and $M$ is the preconditioner matrix defined in (4). Note that the ADMM iteration matrix satisfies $G(\beta) = I - M^{-1}(\beta)H(\beta)$ by definition. In turn, ADMM-GMRES is equivalent to a preconditioned GMRES solution of the augmented KKT system $H(\beta)u = v(\beta)$ using preconditioner $M(\beta)$.

2.4 Reducing the cost of GMRES

A significant shortcoming of GMRES is its need to store and manipulate an $N \times k$ dense matrix at the $k$-th iteration. Its $\Theta(k^2N)$ time and $\Theta(kN)$ memory requirements become unsustainable once $k$ grows large. It was proved by Faber and Manteuffel that these complexity figures cannot be substantially reduced without destroying the optimal property of GMRES [19]. Hence, if many GMRES iterations are desired, then a limited-memory approach must be used.

One limited-memory approach is to periodically restart GMRES: after $p$ iterations, the final iterate is extracted, and used as the initial point for a new set of $p$ iterations. However, a significant issue with restarting ADMM-GMRES is that it may stall, meaning that it would fail to make further progress after a certain number of iterations. Alternatively, we may use a method based on Lanczos biorthogonalization, like BiCG, QMR and their transpose-free variants; see [49] for their implementation details. These methods are entirely heuristic, but tend to work well when GMRES converges quickly, and do not stall as easily as restarted GMRES. QMR is often preferred over BiCG for being more stable and for having a convergence analysis somewhat related to GMRES.

Figure 2 compares GMRES with restarted GMRES and QMR for an “easy” and “difficult” problems of comparable conditioning. In the “easy” example, all these methods outperform basic ADMM, with QMR converging almost as rapidly as GMRES. But in the “difficult” example, all of these limited-memory methods stall, or get close to stalling. Only the full GMRES is able to converge at the $(\kappa^{1/4} - 1)/(\kappa^{1/4} + 1)$ rate.

8
Figure 2: Comparison of GMRES with its limited-memory variants: (a) an “easy” example; (b) a “difficult” example.

3 GMRES Convergence Analysis

Krylov subspace methods like GMRES are closely associated with the idea of approximating a matrix inverse using a low-order matrix polynomial. To explain, consider substituting the constraints of the GMRES least-squares problem \((13)\) into its objective, to yield

\[
\|b - Ax^k\| = \min_{\alpha_1, \ldots, \alpha_k} \left\| b - A \left( \sum_{j=1}^{k} \alpha_j A^{j-1} b \right) \right\| = \min_{p \in \mathbb{P}_{k-1}} \| [I - Ap(A)]b \|. \tag{16}
\]

If there exists an accurate low-order matrix polynomial approximation \(p(A)\) for the matrix inverse \(A^{-1}\), then \(Ap(A) \approx I\) and \(\|[I - Ap(A)]b\| \approx 0\). GMRES must converge rapidly, since it optimizes over all polynomials in \((16)\).

Equivalently, we may solve the residual minimization problem

\[
\min_{p \in \mathbb{P}_{k-1}} \{ \|q(A)b\| : q(z) = 1 - zp(z) \} = \min_{q \in \mathbb{P}_{k-1}} \|q(A)b\|, \tag{17}
\]

and recover \(p(A)\) via \(p(z) = z^{-1}(q(z) - 1)\). Equation \((17)\) is the standard tool for analyzing the convergence of optimal Krylov subspace methods like GMRES; see [28, 16, 49]. The typical proof constructs a family of heuristic polynomials \(h_0(z), h_1(z), \ldots, h_k(z)\), and establishes a desired geometric convergence rate \(0 \leq \varrho < 1\) for the associated matrix induced 2-norm, as in \(\|h_k(A)\| \leq \alpha \varrho^k\).

Then, since GMRES optimizes over all polynomials in \((17)\), it must converge at least as quickly, i.e. \(\min_{p} \|p(A)\| \leq \|h_k(A)\|\).

In this section, we will simplify \((17)\) associated with \(A \leftarrow I - G(\beta)\) to an easier one associated with the \(m \times m\) nonsymmetric matrix

\[
K(\beta) \triangleq \begin{bmatrix} Q^T & \tilde{D}^2 + I \end{bmatrix} \begin{bmatrix} \beta^{-1} \tilde{D}^{-1} & - (\beta \tilde{D}^{-1} + I)^{-1} \end{bmatrix} \begin{bmatrix} \beta^{-1} \tilde{D}^{-1} + I \end{bmatrix}, \tag{18}
\]

in which \(\tilde{D} = (AD^{-1}A^T)^{-1}\) and \(B = [Q \ P] \begin{bmatrix} R & 0 \end{bmatrix}\) is the QR decomposition of \(B\). We begin by showing that only \(m\) eigenvalues of the ADMM iteration matrix are nonzero, and these coincide with the eigenvalues of \(G_{22}(\beta) = \frac{1}{2}I + \frac{1}{2}K(\beta)\).
Lemma 1. Let $B = [Q \, P] \begin{bmatrix} R & 0 \end{bmatrix}$ be the QR decomposition of $B$. Then the ADMM iteration matrix $G(\beta)$ defined in (7) has a block decomposition

$$G(\beta) = \begin{bmatrix} I_n & 0 & 0 \\ 0 & R^{-1} & 0 \\ 0 & 0 & P \end{bmatrix} \begin{bmatrix} 0_n & G_{12}(\beta) & G_{13}(\beta) \\ 0 & 0 & G_{22}(\beta) \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} I_n & 0 & 0 \\ 0 & R & 0 \\ 0 & 0 & P^T \end{bmatrix}, \quad (19)$$

where

$$G_{12}(\beta) = -\beta D^{-1} A^T (\beta^{-1} \tilde{D} + I)^{-1} [Q \, P],$$

$$G_{13}(\beta) = -\beta D^{-1} A^T (\beta^{-1} \tilde{D} + I)^{-1} Q,$$

$$G_{22}(\beta) = \frac{1}{2} I + \frac{1}{2} K(\beta), \quad G_{23}(\beta) = \begin{bmatrix} Q^T \\ -P^T \end{bmatrix} (\beta \tilde{D}^{-1} + I)^{-1} Q,$$

$K(\beta)$ is defined in (18), and $\tilde{D} = (AD^{-1}A^T)^{-1}$.

Proof. Follows from direct computation and applications of the Sherman–Morrison–Woodbury identity.

Remark 2. The special structure of $K(\beta)$ allows its singular values to be explicitly stated. In particular,

$$\|K(\beta)\| = \frac{\gamma - 1}{\gamma + 1} \text{ where } \gamma = \max \left\{ \frac{L}{\beta}, \frac{\beta}{\mu} \right\}, \quad (20)$$

so the eigenvalues of $K(\beta)$ are contained within the disk on the complex plane centered at at the origin, with radius $(\gamma - 1)/(\gamma + 1)$. This is an important characterization of $K(\beta)$ that we will use extensively in Sections 4 & 5.

Furthermore, the block pattern in the Schur decomposition (19) suggests that the Jordan block associated with each zero eigenvalue of $G(\beta)$ is at most size $2 \times 2$. After two iterations, ADMM becomes entirely dependent upon the inner iteration matrix $G_{22}(\beta)$.

Corollary 3. Given $\beta > 0$ and any polynomial $p(\cdot)$, we have

$$\|p(G(\beta)) G^2(\beta)\| \leq c_0 \left( 1 + \beta \|D^{-1} A^T\| \right) \|p(G_{22}(\beta))\|,$$

where $c_0 = 2 \max\{\sigma_{\max}(B), 1/\sigma_{\min}(B), \sigma_{\max}(B)/\sigma_{\min}(B)\}$.

Proof. Let us write $G \equiv G(\beta)$. For each matrix monomial, we substitute Lemma 1 and note that the following holds for all $j \geq 0$

$$G^j = U \begin{bmatrix} 0 & G_{12} & G_{13} \\ 0 & G_{22} & G_{23} \\ 0 & 0 & 0 \end{bmatrix}^{j+2} U^{-1} = U \begin{bmatrix} G_{12} \\ G_{22} \\ 0 \end{bmatrix} G_{22}^j \begin{bmatrix} 0 & G_{22} & G_{23} \end{bmatrix} U^{-1}.$$

Repeating this argument for each monomial in $p(\cdot)$, we find that

$$\|p(G) G^2\| \leq \kappa_U \|G_{12} \, G_{22}\| \|G_{22} \, G_{23}\| \|p(G_{22})\|,$$

$$\leq \kappa_U (\beta \|D^{-1} A^T\| + 1) (1 + 1) \|p(G_{22})\|,$$

where $\kappa_U = \|U\|\|U^{-1}\| = \max\{\sigma_{\max}(B), 1/\sigma_{\min}(B), \sigma_{\max}(B)/\sigma_{\min}(B)\}$. \qed
Accordingly, the residual minimization problem (17) posed over $G(\beta)$ is reduced to a simpler problem over $K(\beta)$ after two iterations.

**Lemma 4.** Fix $\beta > 0$ and initial point $u^0$. Let $u^k$ be the iterate generated at the $k$-th iteration of ADMM-GMRES (Algorithm 7). Then the following holds for all $k \geq 2$

$$\frac{\|u^k - u^*\|_M}{\|u^0 - u^*\|_M} \leq (c_0 + c_1 \beta) \min_{p \in P_{k-2}} \|p(K(\beta))\|,$$

where $K(\beta)$ is defined in (18), $P_k$ is the space of order-$k$ polynomials, and $c_0, c_1$ are constants.

**Proof.** Substituting $A \leftarrow (I - G(\beta))$ and $b \leftarrow r$ into (17) yields

$$\min_{p \in P_k, p(0) = 1} \|p(I - G(\beta)r)\| = \min_{p \in P_k, p(1) = 1} \|p(G(\beta)r)\| \leq \|r\| c(\beta) \min_{p \in P_{k-2}, p(1) = 1} \|p(G_{22}(\beta))\| \leq \|r\| c(\beta) \min_{p \in P_{k-2}, p(1) = 1} \|p(K(\beta))\|.$$

Equality (a) shifts the polynomials $p(1 - z) \leftrightarrow p(z')$, which also shifts the constraint point from $z = 0$ to $z' = 1$. Inequality (b) takes the heuristic choice of $p(z) = z^2 q(z)$ with an order $k - 2$ polynomial $q$, and substitutes Corollary 3. Equality (c) then shifts and scales the polynomials $p(z) \leftrightarrow p(\frac{1}{2} + 1 \frac{1}{2} z')$, keeping the constraint point $z = 1$ at $z' = 1$. \hfill \Box

### 4 Worst-Case Behavior

When applied to (ECQP), ADMM converges at the rate of $1 - 1/\sqrt{\kappa}$ with the parameter choice of $\beta = \sqrt{\mu L}$; a number of previous authors have established versions of the following statement [14, 24, 23, 39].

**Proposition 5.** The $k$-th iterate of ADMM with $k \geq 2$ satisfies

$$\frac{\|u^k - u^*\|_M}{\|u^0 - u^*\|_M} \leq (c_0 + c_1 \beta) \left( \frac{\gamma}{\gamma + 1} \right)^{k-2}$$

where $\gamma = \max\{L/\beta, \beta/\mu\}$, and $c_0, c_1$ are constants. The bound is sharp up to a multiplicative constant.

**Proof.** The residuals satisfy $r^k = G^k(\beta)r^0$, so $\|r^k\|/\|r^0\| \leq \|G^k(\beta)\|$. To establish the inequality, we substitute $\|G_{22}^k\| \leq (\frac{1}{2} + \frac{1}{2} \|K\|^k)$ into Lemma 3. To prove sharpness, take $m$ to be even and set $A = I_m$, $B = [I_m, 0, m]_T$, and $D$ diagonal. Then both $G_{22}$ and $K$ are also diagonal, so $\|G_{22}^k\| = (\frac{1}{2} + \frac{1}{2} \|K\|^k)$ trivially holds. \hfill \Box

Let us use Lemma 4 to prove a similar statement for ADMM-GMRES.

**Theorem 6.** The $k$-th iteration of ADMM-GMRES satisfies

$$\frac{\|u^k - u^*\|_M}{\|u^0 - u^*\|_M} \leq (c_0 + c_1 \beta) \left( \frac{\gamma - 1}{\gamma + 1} \right)^{k-2}$$

where $\gamma = \max\{L/\beta, \beta/\mu\}$, and $c_0, c_1$ are constants. The bound is sharp up to a multiplicative constant.
Proof. To establish the inequality, set \( p(z) \) in Lemma 4 to be the monomial \( p(z) = z^{k-2} \) and take \( \|K^k\| \leq \|K\|^k \). To prove sharpness, we give a problem construction satisfying \( \|K^k\| = \|K\|^k \) whose optimal polynomial is precisely \( p^*(z) = z^{k-2} \). Let us take \( m \) to be even and consider
\[
A = I_m, \quad D = \begin{bmatrix} 1 & 0 \\ \sqrt{\kappa} & 0 \\ 0 & \sqrt{\kappa} \end{bmatrix},
\]
\[
B = \begin{bmatrix} \cos \Theta & \sin \Theta \\ \sin \Theta & -\cos \Theta \end{bmatrix}
\]
where \( \Theta = \frac{\pi}{2m} \) diag \((1, 3, 5, \ldots, m - 1)\).

(Note that \( B \) is orthogonal with complement \( P = \begin{bmatrix} \sin \Theta & \cos \Theta \end{bmatrix}^T \).) By inspection, we have \( \mu = 1/\sqrt{\kappa}, L = \sqrt{\kappa}, \) and \( L/\mu = \kappa \). The choice of \( \beta = \sqrt{\mu L} \) minimizes \( \gamma \), and \( K(\sqrt{\mu L}) \) is the scaled orthogonal matrix
\[
K(\sqrt{\mu L}) = \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \begin{bmatrix} \cos 2\Theta & -\sin 2\Theta \\ \sin 2\Theta & \cos 2\Theta \end{bmatrix},
\]
whose eigenvalues lie evenly spaced along the circumference of a circle: \( \lambda_i = a\omega^i \), where \( a = \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \) and \( \omega = \exp(\sqrt{-1} \frac{2\pi}{m}) \). We have \( \|K^k\| = \|K\|^k \); let us show that the optimal polynomial is \( p^*(z) = z^{k-2} \) in the limit \( m \to \infty \). Consider
\[
\min_{\substack{p \in \mathbb{P}_{k-2} \\ p(1)=1}} \|p(K(\sqrt{\mu L}))\| = \min_{\substack{p \in \mathbb{P}_{k-2} \\ p(1)=1}} \max_{i \in \{1, \ldots, m\}} |p(\lambda)| \leq \min_{\substack{p \in \mathbb{P}_{k-2} \\ p(1)=1}} \max_{|\lambda| = a} |p(\lambda)| \equiv a^{k-2}.
\]

Step (a) makes a unitary eigendecomposition for the normal matrix \( K(\sqrt{\mu L}) = XSX^* \), where \( XX^* = X^*X = I \), and notes that \( \|p(K(\sqrt{\mu L}))\| = \|Xp(S)X^*\| = \|p(S)\| \). Step (b) encompasses the roots of unity \( \omega^j \) within the unit circle \( \{z : |z| = 1\} \). Step (c) then solves the approximation problem over the circle using the closed-form solution \( p(z) = z^{k-2} \) due to Zarantonello (see [16] for a proof, or [13] for a more intuitive explanation). The inequality (b) is sharp in the limit \( m \to \infty \), i.e. when the \( m \)-th roots of unity converge towards the unit circle.

Setting \( \beta = \sqrt{\mu L} \) minimizes value of \( \gamma = \max\{L/\beta, \beta/\mu\} \). This parameter choice allows ADMM to converge to an \( \epsilon \)-accurate solution in
\[
\left( (\sqrt{\kappa} + 1) \log \left( \frac{c_0 + c_1 \beta}{\epsilon} \right) \right) \text{ iterations,}
\]
and ADMM-GMRES to do the same in
\[
\left( \frac{1}{2} (\sqrt{\kappa} + 1) \log \left( \frac{c_0 + c_1 \beta}{\epsilon} \right) \right) \text{ iterations.}
\]
We see that ADMM-GMRES is only a factor of two better than basic ADMM. In the worst case, GMRES will not be able to yield a substantial acceleration over the basic ADMM method. This is easily verified numerically; see Figure 3.

Indeed, the optimal polynomial used to prove Theorem 6 may be extracted and explicitly applied as a successive over-relaxation scheme.

**Corollary 7.** Consider the successive over-relaxation iterations
\[
\begin{align*}
u^1 &= G(\beta)u^0 + v(\beta), \\
u^2 &= G(\beta)u^1 + v(\beta), \\
u^{i+1} &= (1 - \omega)u^i + \omega[G(\beta)u^i + v(\beta)] \quad \forall j \in \{3, \ldots, k\}
\end{align*}
\]
with \( \beta = \sqrt{\mu L} \) and \( \omega = 2 \). Then the \( k \)-th iterate \( u^k \) satisfies the bound in Theorem 6.
Figure 3: The problem construction in the proof of Theorem 6 places the eigenvalues of \( K(\beta) \) in a circle, so ADMM-GMRES converges at the same asymptotic rate as over-relaxed ADMM with \( \omega = 2 \).

Proof. The SOR residuals satisfy \( r^k = q(G)r^0 \) where \( q(z) = \prod_{i=1}^{k}[(1 - \omega_j) + \omega_j z] \). Collocating its roots with those of the optimal polynomial \( p^*(z) \) in the proof of Theorem 6 yields the desired iterates.

This is precisely over-relaxed ADMM using the parameter choice of \( \omega = 2 \), which was shown to be optimal by several previous authors \[12, 23, 39\].

5 Explaining Convergence in \( O(\kappa^{1/4}) \) Iterations

In order to understand the circumstances that allow the ADMM-GMRES converge an order-of-magnitude faster than basic ADMM, we make the following assumption.

Assumption 2 (\( \kappa_X \) is bounded). The matrix \( K(\beta) \), defined in (18), is diagonalizable for every choice of \( \beta > 0 \). Fix \( \beta > 0 \) and write \( K(\beta) = XSX^{-1}, X = \text{diag}(\lambda_1, \ldots, \lambda_m) \) as its eigendecomposition. Then the condition number for the matrix-of-eigenvectors, \( \kappa_X = \|X\|\|X^{-1}\| \), is bounded by an absolute constant.

Intuitively, this is to assume that the behavior of \( K(\beta) \) is well-described by its eigenvalues; we will return to this point later in Section 6. Substituting \( \|p(K)\| = \|Xp(S)X^{-1}\| \leq \|X\|\|p(S)\|\|X^{-1}\| \) reduces the residual minimization problem in Lemma 4 to an eigenvalue approximation problem (see e.g. [50])

\[
\frac{\|u^k - u^*\|_M}{\|u^0 - u^*\|_M} \leq (c_0 + c_1 \beta)\kappa_X \min_{p \in P_{k-2}} \max_{p(1)=1} \|p(z)\|_\Lambda, \tag{24}
\]

where we have used the maximum modulus notation

\[
\|p(z)\|_\Lambda \stackrel{\Delta}{=} \max_{z \in \Lambda} |p(z)|, \quad \Lambda \stackrel{\Delta}{=} \{\lambda_1, \ldots, \lambda_m\}.
\]
Figure 4: The enclosure $\mathcal{I} \cup \mathcal{C}$ bounds the outlier eigenvalues away from the right-side of the disk by the distance $\delta \cdot a$: (a) graphical illustration; (b) numerical example, for a problem with $n = 500$, $m = 400$, $\ell = 50$.

In this new problem, our objective is to construct a low-order polynomial whose zeros are approximately the eigenvalues of $K(\beta)$.

Problem (24) is made easier by enclosing the eigenvalues $\Lambda$ within the disk $D = \{z \in \mathbb{C} : |z| \leq a\}$ mentioned earlier in Remark 2 with radius

$$a \equiv \frac{\gamma - 1}{\gamma + 1} \text{ where } \gamma = \max \left\{ \frac{L}{\beta}, \frac{\beta}{\mu} \right\}.$$

(25)

In view of Theorem 6, this enclosure $\Lambda \subset D$ is sharp: there exists a choice of problem data $A, B, D$ to place $\Lambda$ right along the boundary $\partial D$. The associated optimal polynomial is simply $p^*(z) = z^k$, but this causes ADMM-GMRES to converge at the same rate as regular ADMM.

In order to improve upon the $O(\sqrt{\kappa})$ iteration estimate from Theorem 6, we must introduce additional information about the distribution of eigenvalues within the interior of the disk. Suppose, in particular, that all of our eigenvalues were also real, i.e. $\Lambda \subset \mathbb{R}$. Then the corresponding approximation problem over the real interval $I = \mathbb{R} \cap D$ has a closed-form solution attributed to Chebyshev (see [28, Ch.3] or [49, Sec.6.11.1])

$$\min_{\substack{p \in \mathbb{P}_k \\cap I \subseteq \mathbb{R} \\cap D}} \|p(z)\|_I = \frac{1}{|T_k(1/a)|} \leq \left( \frac{\sqrt{\gamma} - 1}{\sqrt{\gamma} + 1} \right)^k,$$

(26)

attained by $p^*(z) = T_k(z/a)/|T_k(1/a)|$ where $T_k(z)$ is the order-$k$ Chebyshev polynomial of the first kind. Since $\Lambda \subset I$, we may substitute $\|p^*(z)\|_I \geq \|p^*(z)\|_A$ into (24) to find that ADMM-GMRES converges to an $\epsilon$-accurate solution in $O(\kappa^{1/4})$ iterations with the parameter choice $\beta = \sqrt{\mu L}$, for an order of magnitude improvement over the worst-case estimate in Proposition 6.

5.1 Damping the outlier eigenvalues

In practice, $K(\beta)$ also has a number of eigenvalues with nonzero imaginary parts. These eigenvalues prevent (26) from being directly applicable, so we refer to them as outlier eigenvalues. The issue of outliers is standard in Krylov subspace methods. If the number of outliers is small, then a
standard technique is to annihilate them one at a time, and to apply a Chebyshev approximation to the remaining eigenvalues that lie along a line; see [28, p.53] or [16, Sec.5]. In our numerical experiments, however, the number of outlier eigenvalues was often observed to be quite large.

Instead, let us assume a different structure: that the outlier eigenvalues are *better conditioned* than the real eigenvalues. Rather than annihilating them one at a time, it may be sufficient to “dampen” their effect using a few fixed-point iterations, like in multigrid methods. Then, the Chebyshev approximation can be used to approximate the remaining purely-real but poorly-conditioned eigenvalues. Since GMRES is optimal, it must converge faster than this heuristic approach.

Consider the eigenvalue enclosure \( \Lambda \{ K \} \subset I \cup C \), where \( I = \mathbb{R} \cap D \) is the same real interval considered in the previous section, and

\[
C \triangleq \{ z \in D : \text{Re}\{z\} \leq \delta \cdot a \}, \quad \delta \triangleq 1 - \max_{\lambda \in \Lambda(K(\beta))} \frac{\text{Re}\lambda}{\|K(\beta)\|} \geq 0
\]

is used to encompasses the outlier eigenvalues; an illustration is shown in Figure 4. We view the quantity \( \delta^{-1} \) as a relative condition number of \( C \), due to the following result.

**Lemma 8.** The approximation problem for \( C \) in \((27)\) is bound

\[
\min_{p \in P_k} \|p(z)\|_C \leq \left(1 - \frac{\delta}{2}\right)^{k/2},
\]

using the polynomial \( p(z) = (z + a)^k/(1 + a)^k \).

**Proof.** We use the over-relaxation polynomial \( p(z) = (z + \omega)^k(1 + \omega)^{-k} \) to approximate \( C \). Since \( \omega > 0 \), the maximum modulus is attained at \( \arg\max_{z \in C} |p(z)| = a \left(1 - \frac{\delta}{2}\right) \pm j \sqrt{1 - (1 - \delta)^2} \). Substituting \( (1 + \omega)^{-1} \leq (a + \omega)^{-1} \) yields a convenient bound \( \max_{z \in C} |p(z)| \leq \left[1 - 2\delta \omega(a + \omega)^{-2}\right]^{k/2} \) that attains its minimum at \( \omega = a \).

Fixing \( \delta > 0 \), the error over \( C \) can be dampened to some prescribed accuracy in a fixed number of iterations, independent of all other considerations. This is the central insight that we use in our new iteration estimate; so long as \( \delta > 0 \), ADMM-GMRES will converge in \( O(\kappa^{1/4}) \) iterations.

**Theorem 9.** Under Assumption \( 3 \) the \( k \)-th iterate of ADMM-GMRES satisfies

\[
\frac{\|u^k - u^*\|_M}{\|u^0 - u^*\|_M} \leq 2(c_0 + \beta c_1)\kappa_X \left(\frac{\sqrt{\gamma} - 1}{\sqrt{\gamma} + 1}\right)^{\delta(k-2)/6},
\]

where \( \delta \) is defined in \((27)\), and \( c_0, c_1, \kappa_X \) are constants.

**Proof.** Consider the polynomial \( p_k(z) \) that is the product of \( \eta \) iterations of the fixed-point iterations in Lemma 8 and an order-(\( k-\eta \)) scaled Chebyshev polynomial for the interval \( I = \{ z \in \mathbb{R} : |z| \leq a \} \):

\[
p_k(z) \triangleq \frac{T_\xi(z/a)}{T_\xi(1/a)} \left(\frac{z + a}{1 + a}\right)^\xi, \quad \xi \triangleq k - \eta.
\]

We use \( p_k(z) \) to solve the approximation problem over \( I \cup C \supset \Lambda \). Noting that \( \|p_k(z)\|_{I \cup C} = \max \{\|p_k(z)\|_I, \|p_k(z)\|_C\} \), we bound each component

\[
\|p_k(z)\|_C \leq \left(\frac{2\sqrt{1 - \delta/2}}{1 + a}\right)\eta \left(1 + \frac{\sqrt{2}}{\xi}\right)^\xi, \quad \xi \triangleq k - \eta.
\]

\[
\|p_k(z)\|_I \leq 2 \frac{a}{1 + a} \frac{1}{T_\xi(1/a)} \leq 2 \left(\frac{\sqrt{\gamma} - 1}{\sqrt{\gamma} + 1}\right)^\xi,
\]

\[
\|p_k(z)\|_C \leq \left(1 - \frac{\delta}{2}\right)^{k/2}.
\]
using \( \max_{|z| \leq 1} |T_n(z)| \leq (1 + \sqrt{2})^n \). We will pick the ratio \( \eta/\xi \) to guarantee \( \|p_k(z)\|_I \geq \|p_k(z)\|_C \). This means choosing the ratio \( \eta/\xi \) to satisfy

\[
\left( 1 - \frac{\delta}{2} \right)^{\eta/\xi} \leq \frac{1}{(1 + \sqrt{2})^2}.
\]

(31)

Viewing \( \eta/\xi \) as an “iteration estimate” to guarantee a relative error reduction of \( \epsilon = 1/(1 + \sqrt{2})^2 \) over \( C \), we take logarithms and obtain \( \eta/\xi \geq 2\delta^{-1} \log \epsilon^{-1} \). Setting \( \eta/\xi = c_3/\delta \) with \( c_3 = 4 \log(1 + \sqrt{2}) \) satisfies this bound, so that \( \|p(z)\|_I \geq \|p(z)\|_C \). Since \( \delta \leq 2 \), we have \( \xi = \delta k/(c_3 + \delta) \geq \delta k/6 \). Finally, substituting \( \|p_k(z)\|_{I \cup C} = \|p_k(z)\|_I \geq \|p_k(z)\|_A \) into (24) to yield the desired statement.

\[ \square \]

**Remark 10.** Intuitively, each increment of \( \xi \) increases the order of the Chebyshev polynomial, which decreases global error by a factor of \( (\sqrt{\gamma} - 1)/(\sqrt{\gamma} + 1) \), but also increases the relative error between \( C \) and \( I \) by a factor of \( 1 + \sqrt{2} \). To smooth this error between the two regions, we increment \( \eta \) by a fixed number that is determined only by Lemma [3] and \( \delta \). Alternating between decrementing the global error and smoothing the relative error allows us to converge at the accelerated rate of \( (\sqrt{\gamma} - 1)/(\sqrt{\gamma} + 1) \).

Theorem [9] says that ADMM-GMRES will converge to an \( \epsilon \)-accurate solution in

\[
2 + \left\lceil \frac{6(\kappa^{1/4} + 1)}{\delta} \log \left( \frac{2\kappa_X(c_0 + c_1\beta)}{\epsilon} \right) \right\rceil \text{ iterations}
\]

using the parameter choice \( \beta = \sqrt{\mu L} \). This is a factor of \( O((\kappa^{1/4} + 1) \log(\kappa_X/\epsilon)) \). So long as \( \delta \) is not too small relative to \( 1/\kappa^{1/4} \) and \( \kappa_X \) not too big relative to \( 1/\epsilon \), Theorem [9] guarantees convergence in \( O(\kappa^{1/4}) \) iterations.

### 5.2 Explaining the empirical results

Earlier in the introduction, we presented a comparison of ADMM and ADMM-GMRES for 1000 random trials. These problems were generated using the following algorithm.

**Algorithm 2.** Input: dimension parameters \( n, m \leq n, \ell \leq m \) and conditioning parameter \( s > 0 \). Output: random data matrices \( D \in \mathbb{S}^n, A \in \mathbb{R}^{m \times n}, B \in \mathbb{R}^{m \times \ell} \) satisfying Assumption [1].

1. Select the singular vectors \( U_A, U_B, U_D, V_A, V_B \) i.i.d. uniformly from their respective orthogonal groups.
2. Select the singular values \( \Sigma_A, \Sigma_B, \Sigma_D \) i.i.d. from the log-normal distribution \( \sim \exp(0, s^2) \).
3. Output \( A = U_A \Sigma_A V_A^T, B = U_B \Sigma_B V_B^T, \) and \( D = U_D \Sigma_D V_D^T \).

More specifically, the dimension parameters \( n, m, \ell \) were uniformly sampled from \( n = 1000, m \in \{1, \ldots, n\} \), and \( \ell \in \{1, \ldots, m\} \), and the log-standard-deviation is swept within the range \( s \in [0, 2] \). Both algorithms are tasked with solving the equation to a relative residual of \( \epsilon = 10^{-6} \). ADMM fails to converge within 100,000 iterations for 49 of the problems, while ADMM-GMRES converges on all of the problems.

To verify whether Theorem [9] is sufficient to explain the \( O(\kappa^{1/4}) \) behavior seen in these 1000 random problems, we plot the distribution of \( \delta \) and \( \kappa_X \) with respect to \( \kappa \) in Figure [5a] and Figure [5b]. The smallest value of \( \delta \) is 0.06, with mean and median both around 0.6. The largest value of \( \kappa_X \) is 775, with mean and median both around 50. These are both relatively modest, and as predicted by Theorem [9] ADMM-GMRES converges in \( O(\kappa^{1/4}) \) iterations.
Figure 5: Statistics for $\delta$ and $\kappa_X$ for the 1000 randomly-generated problems in Figure 1: (a) & (b) scatter plot and empirical CDF for $\delta$ and its lower bound $\delta_{lb}$ (Lemma 11); (c) & (d) scatter plot and empirical CDF for $\kappa_X$. 
The associated cumulative probability distributions are shown in Figure 5p and Figure 5l. An exponentially decaying probability tail for both quantities can be observed. The rapid roll-off in probability tail is a signature trait for concentration-of-measure type results. In the case of δ, consider the following bound.

**Lemma 11.** Define $K(β)$, $Q$, $P$ as in (18). Then

$$δ_{lb} ≜ 1 - \frac{∥Q^TK(β)Q∥ + ∥P^TK(β)P∥}{2∥K(β)∥} \leq δ.$$  

**Proof.** Note that $K(β)$ has the following block structure $K = \begin{bmatrix} X & Y \\ -Z^T & \mathbf{0} \end{bmatrix}$. For such matrices, Benzi & Simoncini [7] used a field-of-values type argument to show that if $λ \in \Lambda(K)$ and $\text{Im}(λ) \neq 0$, then $|\text{Re}λ| \leq \frac{1}{2} (∥X∥ + ∥Y∥)$. Substituting the definitions of $δ$, $X$, and $Y$ results in the desired bound. \hfill \Box

Hence, we see that the quantity $δ$ is bounded away from zero because the matrices $K$, $Q$, $P$ are incoherent. More specifically, let us write $K = USV^T$ as its singular value decomposition. If we treat $Q$, $P$, $U$, and $V$ all as random orthogonal matrices, then the matrices $Q^TU$, $P^TU$, $Q^TV$, and $P^TV$ are all dense with an overwhelming probability [15, Thm.VIII.1]. This observation bounds the expected value of $∥Q^TK(β)Q∥$ and $∥P^TK(β)P∥$ away from $∥K(β)∥$, thereby bounding $δ$ away from zero via Lemma 11.

### 6 The normality assumption

A weakness in our argument is Assumption 2, which takes $κ_X$, the condition number for the matrix of eigenvectors for $K(β)$, to be bounded. The assumption is closely related to the normality of $K(β)$. A matrix is normal if it has a complete set of orthogonal eigenvectors, so if $K(β)$ is normal, then $κ_X = 1$, and our bounds are sharp up to a multiplicative factor. On the other hand, if $K(β)$ is nonnormal, then our bounds may fail to be sharp to an arbitrary degree. The phenomenon has to do with the fact that eigenvalues are not necessarily meaningful descriptors for the behavior of nonnormal matrices; see the discussions in [28, 16, 18] for more details, and the book [56] for a thorough exposition.

In the case of ADMM, there are reasons to believe that $K(β)$ is relatively close to normal, and that Assumption 2 is not too strong in practice. To explain, consider the following dimensionless nonnormality measure

$$ν(A) ≜ ∥A^TA - AA^T∥_F^{1/2}/∥A∥_F,$$

which takes on values from 0 (attained by any normal matrix) to $\sqrt{2}$ (attained by highly nonnormal matrices like $[1 \; 0 \; 0 \; 0]$). The measure is closely associated to Henrici’s departure from normality [29], and can be used to bound many other measures of nonnormality; see the survey in [17].

**Proposition 12.** Let $K(β)$ be the $m \times m$ matrix in (18). Then

$$ν(K(β)) \leq \frac{\left(8\min\{ℓ, m-ℓ\}\right)^{1/4}}{∥K(β)∥_F/∥K(β)∥} \leq \sqrt{2}m^{1/4}\frac{∥K(β)∥}{∥K(β)∥_F}.$$  

**Proof.** Note that $K ≜ K(β)$ has the structure $K = JU^TW$, where $J = \text{diag}(I_ℓ, -I_{m-ℓ})$, $U$ is orthonormal, and $W = W^T$ shares its singular values with $K$. Then $∥K^TK - KK^T∥_F = ∥UW^2U - JU^TW^2UJ∥_F^2 = 2∥Q^TW^2P∥_F^2$, where $Q$ is the first $ℓ$ columns of $U$, and $P$ is the
remaining $m - \ell$ columns. But $\|Q^TW^2P\|_F^2 = \text{tr}W^2QQ^TW^2PP^T \leq \|W\|^4 \min\{\text{tr}QQ^T, \text{tr}PP^T\}$, and taking fourth roots produces the first inequality. The second inequality follows by maximizing the bound with $\ell = m/2$.

In the literature, the ratio $\|K\|_F^2/\|K\|^2$ is sometimes known as the numerical rank of $K$; see [17]. Taking on a value from 1 to $m$, it is always bounded by, and is a stable relaxation of, the rank of $K$.

Assuming that the numerical rank of $K$ grows linearly with its dimension $m$ (e.g. if the data were generated using Algorithm 2), then substituting $\|K\|_F^2/\|K\|^2 \in \Omega(m)$ into Proposition 12 produces $\nu(K) \in O(m^{-1/4})$. The matrix $K$ becomes more and more normal as its dimension $m$ grows large, since $\nu(K)$ decays to zero. While the observation does not provide a rigorous bound for $\kappa_X$, it does concur with our numerical results presented in later sections.

Finally, even if Assumption 2 fails to hold, ADMM-GMRES will still obey its worst-case bound (Proposition 6). Convergence is guaranteed in $O(\sqrt{\kappa})$ iterations, although the significant acceleration from GMRES may be lost.

7 Comparison with classic preconditioners

Throughout this paper, we have treated ADMM-GMRES as a preconditioned Krylov subspace method for the KKT equations associated with (ECQP)

$$
\begin{bmatrix}
D & 0 & A^T \\
0 & 0 & B^T \\
A & B & 0
\end{bmatrix}
\begin{bmatrix}
x \\
z \\
y
\end{bmatrix}
= 
\begin{bmatrix}
-c \\
-p \\
d
\end{bmatrix},
$$

while assuming that certain block operations can be efficiently performed. But (32) is a standard saddle-point system—albeit with a singular (1, 1) block—and preconditioned Krylov subspace methods for such problems are mature and well-developed; we refer the reader to the authoritative surveys [6, 1].

In this section, we benchmark ADMM-GMRES against classic preconditioned Krylov subspace methods for saddle-point problems (selected from [6]), over random instances of (32) generated using Algorithm 2. As mentioned earlier in Section 2, ADMM requires efficient matrix-vector products with $(B^TB)^{-1}$ and $(\beta^{-1}D + A^TA)^{-1}$. For the purposes of these numerical examples, we make the following assumption, which we satisfy by precomputing and storing the ingredients.

**Assumption 3.** The Cholesky factorizations $B^TB = L_BL_B^T$, $D = L_DL_D^T$, and the eigendecomposition $(AD^{-1}A^T)^{-1} = V\Lambda V^T$ are explicitly available.

Then the identities

$$
(\beta^{-1}D + A^TA)^{-1}x = \beta D^{-1} [I - A^T(\beta^{-1}I + AD^{-1}A^T)^{-1}AD^{-1} ] x
$$

$$
A(\beta^{-1}D + A^TA)^{-1}A^Ty = [\beta^{-1}(AD^{-1}A^T)^{-1} + I]^{-1}y = V(\beta^{-1}\Lambda + I)^{-1}V^Ty
$$

render the cost of each application of the ADMM preconditioner—each iteration of ADMM—to be quadratic with respect to the input dimensions $n$, $\ell$, and $m$. (To concur with the notations used throughout this paper, we dimension the variables $x \in \mathbb{R}^n$, $z \in \mathbb{R}^\ell$, and $y \in \mathbb{R}^m$.)}
7.1 Alternative Preconditioners

We restrict our attention to classic preconditioners based on the assumed structure in Assumption 3. These methods are selected from the survey [6], and solve either the reduced augmented system
\[
\begin{bmatrix}
0 & B^T \\
B & -AD^{-1}A
\end{bmatrix}
\begin{bmatrix}
z \\
y
\end{bmatrix} =
\begin{bmatrix}
-p \\
d'
\end{bmatrix}, \tag{33}
\]
or the Schur complement problem
\[
\begin{bmatrix}
B^T(AD^{-1})^{-1}B
\end{bmatrix}
z = -p', \tag{34}
\]
where the new right-hand sides are obtained via forward substitution
\[
d' = d + AD^{-1}c, \quad p' = p - B^T(AD^{-1}A^T)^{-1}d', \tag{35}
\]
and the unknown variables are recovered via back substitution
\[
y = (AD^{-1}A^T)^{-1}(Bz - d', \quad x = -D^{-1}(A^Ty + c). \tag{36}
\]
Under Assumption 3, each forward substitution (35), backward substitution (36), and matrix-vector product with (33) and (34) can be performed at quadratic cost with respect to the input size.

**Block-diagonal preconditioner (Blk-Diag).** Solve the reduced augmented system (33) using a symmetric indefinite Krylov method like MINRES, with the positive definite matrix
\[
M_1 =
\begin{bmatrix}
\beta B^TB & 0 \\
0 & AD^{-1}A
\end{bmatrix}
\]
serving as preconditioner. The \((2,2)\) block of \(M_1\) matches that of \(\hat{H}\), while its \((1,1)\) block is used to precondition the Schur complement. The preconditioned matrix has the eigenvalue \(-1\) with multiplicity \(m - \ell\), and \(2\ell\) eigenvalues \(\lambda_i = \frac{1}{2}(-1 \pm \sqrt{1 + 4\eta_i})\), where \(\eta_1, \ldots, \eta_\ell\) are the \(\ell\) eigenvalues of \(\beta^{-1}Q^T\hat{D}Q\) [20, Lem.2.1] (see also [6, Thm.3.8] and [48, Lem.2.1]). Applying the classic two-interval approximation result [13] (see also [28, Ch.3]) shows that MINRES converges to an \(\epsilon\)-accurate solution within \(k \leq 1 + \left\lceil \sqrt{\kappa} \log\left(1 + \sqrt{1 + 4L/\beta}\right)/\epsilon \right\rceil\) iterations. We set \(\beta = L\) to obtain convergence in \(O(\sqrt{\kappa})\) iterations.

**Constraint preconditioner I (Constr I).** Solve the reduced augmented system (33) using a general Krylov method like GMRES, with
\[
M_2 =
\begin{bmatrix}
0 & B^T \\
B & -\beta I
\end{bmatrix} =
\begin{bmatrix}
I & \beta^{-1}B^T \\
0 & I
\end{bmatrix}
\begin{bmatrix}
\beta^{-1}B^TB & 0 \\
0 & -\beta I
\end{bmatrix}
\begin{bmatrix}
I & 0 \\
0 & I
\end{bmatrix}
\]
serving as preconditioner. The preconditioner is designed to replicate the governing matrix, while modifying the \((2,2)\) block in a way as to make the overall matrix considerably easy to invert. The preconditioned matrix has the eigenvalue 1 with multiplicity \(2\ell\), and \(m - \ell\) eigenvalues that coincide with the eigenvalues of \(\beta^{-1}P^T\hat{D}^{-1}P\) [32, Thm.2.1]; see also [3, Thm.10.1]. The latter \(m - \ell\) eigenvalues lie within the real interval \([1/(L\beta), 1/(\mu\beta)]\), so assuming diagonalizability (i.e. adopting a version of Assumption 2), GMRES converges to an \(\epsilon\)-accurate solution within \(O(\sqrt{\kappa})\) iterations for all choices of \(\beta \geq 1/L\). We set \(\beta = \sqrt{\mu L}\) to concur with ADMM.

**Constraint preconditioner II (Constr II).** Solve the Schur complement system (34) using a symmetric positive definite Krylov method like conjugate residuals, with
\[
M_3 = B^TB
\]
serving as preconditioner. This is derived by using the Schur complement from the previous preconditioner to precondition the Schur complement of (33). The preconditioned problem has coefficient matrix $Q^T \tilde{D} Q$, whose eigenvalues lie in the real interval $[\mu, L]$. Accordingly, conjugate residuals converge to an $\epsilon$-accurate solution within $O(\sqrt{\kappa})$ iterations.

**Hermitian Skew-Hermitian Splitting (HSS).** Solve the reduced augmented system (33) using a general Krylov method like GMRES, with

$$M_4 = \begin{bmatrix} \alpha I & 0 \\ 0 & -(AD^{-1}A + \alpha I) \end{bmatrix} \begin{bmatrix} \alpha I & B^T \\ -B & \alpha I \end{bmatrix}$$

as preconditioner. Small choices of the parameter $\alpha$ work best, though the method is not sensitive to its exact value [4, 5, 52]. Note that $M_4$ requires matrix-vector products with $(\alpha^2 I + B^T B)^{-1}$ to be efficient. When $\alpha$ is sufficiently small, this matrix may be approximated, e.g. using a few iteration of conjugate gradients preconditioned by $B^T B$. To keep our implementation simple, we set $\alpha = 1/L$ (as recommended by Simoncini and Benzi [52]) and explicitly precompute a Cholesky factorization for $\alpha^2 I + B^T B$.

Our list excludes the Uzawa method (and its inexact variants), incomplete factorizations, and multilevel / hierarchical preconditioners, because they cannot be efficiently realized using only the assumed structure in Assumption 3. We have excluded the Arrow–Hurwicz as it is simply a lower-cost, less accurate version of “Constr I”. We have also excluded the block-triangular version of “Blk-Diag”, because it can be shown to be almost identical to “Constr II” after a single iteration, but requires the more computationally expensive GMRES algorithm.

### 7.2 Results

We solved 1000 random problems with $n = 1000$, $1 \leq m \leq n$, $1 \leq \ell \leq m$, and 1000 random problems with $n = 3000$, $1 \leq m \leq n$, $1 \leq \ell \leq m$, using ADMM, ADMM-GMRES, and the four preconditioned Krylov methods described above, on an Intel Core i7-3960X CPU with six 3.30 GHz cores. All six methods were set to terminate at 1000 iterations. Accuracy was measured as the relative residual norm with respect to the saddle-point equation (32).

Tables 1 & 2 show the number of iterations and CPU time to $10^{-6}$ accuracy. ADMM and all four of the preconditioner Krylov methods converge in $O(\sqrt{\kappa})$ iterations, but ADMM-GMRES consistently converges in $O(\kappa^{1/4})$ iterations. This square-root factor acceleration is large enough to offset the high per-iteration cost of the method, making it the fastest overall. The restarted GMRES variant enjoys some of this acceleration, but is also susceptible to stalling once the problem becomes sufficiently ill-conditioned.

The constraint preconditioner “Constr I” performs surprisingly well for the $n = 1000$ examples in Table 1, consistently outperforming its $O(\sqrt{\kappa})$ iteration bound. Examining closer, however, we find this to be an artifact of the finite convergence property of GMRES. Once the problem size is increased to $n = 3000$, the method is no longer able to solve ill-condition problems with $\kappa \geq 10^6$.

In all of these examples, the per-iteration costs remain approximately constant—even for methods that relied on GMRES—due to the relatively high cost of the preconditioners. Profiling the code, we find that each GMRES iteration takes no more than 4 milliseconds to execute. By contrast, even our fastest preconditioner (Constr II) requires $\sim 36$ milliseconds per application.
Table 1: Max. Iterations (Max. CPU time in seconds) to $\epsilon = 10^{-6}$ for random problems with dimensions $n = 1000$, $1 \leq m \leq n$, $1 \leq \ell \leq m$.

| $\log_{10} \kappa$ | (0, 2] | (2, 4] | (4, 6] | (6, 8] | (8, 10] |
|---------------------|--------|--------|--------|--------|--------|
| Num. trials         |        |        |        |        |        |
| 10$\sqrt{\kappa}$  | 32     | 316    | 3162   | $10^3$ | $10^3$ |
| ADMM                | 126 (0.97) | 982 (5.09) | $10^3$ | $10^3$ | $10^3$ |
| Blk-Diag            | 102 (0.52) | 506 (3.27) | $10^3$ | $10^3$ | $10^3$ |
| Constr I            | 42 (0.45) | 155 (1.19) | 269 (2.11) | 553 (3.10) | 678 (4.43) |
| Constr II           | 49 (0.19) | 227 (1.28) | 722 (3.00) | $10^3$ | $10^3$ |
| HSS                 | 97 (2.72) | 278 (12.6) | 532 (19.4) | $10^3$ | $10^3$ |
| 6$\kappa^{1/4}$    | 11     | 34     | 107    | 337    | 1067   |
| ADGM                | 13 (0.23) | 29 (0.55) | 76 (1.24) | 198 (3.94) | 469 (6.44) |
| ADGM(5)             | 14 (0.25) | 50 (0.68) | $10^3$ | $10^3$ | $10^3$ |
| ADGM(10)            | 13 (0.21) | 37 (0.51) | $10^3$ | $10^3$ | $10^3$ |
| ADGM(25)            | 13 (0.20) | 30 (0.50) | $10^3$ | $10^3$ | $10^3$ |

Table 2: Max. Iterations (Max. CPU time in seconds) to $\epsilon = 10^{-6}$ for random problems with dimensions $n = 3000$, $1 \leq m \leq n$, $1 \leq \ell \leq m$.

| $\log_{10} \kappa$ | (0, 2] | (2, 4] | (4, 6] | (6, 8] | (8, 10] |
|---------------------|--------|--------|--------|--------|--------|
| Num. trials         |        |        |        |        |        |
| 10$\sqrt{\kappa}$  | 32     | 316    | 3162   | $10^3$ | $10^3$ |
| ADMM                | 85 (9.87) | 910 (98.8) | $10^3$ | $10^3$ | $10^3$ |
| Blk-Diag            | 96 (4.14) | 568 (20.2) | $10^3$ | $10^3$ | $10^3$ |
| Constr I            | 52 (3.75) | 244 (10.8) | 636 (20.5) | $10^3$ | $10^3$ |
| Constr II           | 46 (1.65) | 254 (8.18) | $10^3$ | $10^3$ | $10^3$ |
| HSS                 | 87 (19.8) | 378 (72.8) | 843 (212) | $10^3$ | $10^3$ |
| 6$\kappa^{1/4}$    | 11     | 34     | 107    | 337    | 1067   |
| ADGM                | 12 (1.36) | 28 (4.01) | 116 (7.77) | 199 (23.15) | 431 (69.54) |
| ADGM(5)             | 13 (1.65) | 45 (5.98) | $10^3$ | $10^3$ | $10^3$ |
| ADGM(10)            | 12 (1.28) | 37 (4.07) | $10^3$ | $10^3$ | $10^3$ |
| ADGM(25)            | 12 (1.37) | 30 (4.61) | $10^3$ | $10^3$ | $10^3$ |
8 Solving the SDP Newton Subproblem

The vectorized version of the SDP Newton subproblem (NEWT) is

\[
\min \left\{ \frac{1}{2} s^T (W \otimes W) s - \tilde{x}^T s - \tilde{b}^T y : s + Ay = c \right\} 
\]

where \( A = [\text{vec } A_1, \ldots, \text{vec } A_m] \), \( s, c, z \) are vectorizations of \( S, C, Z \), and \( W \) is the interior-point scaling matrix. To match standard notations for interior-point methods, we use \( n \) to refer to the order of the semidefinite cone, and \( m \) to refer to the the number of equality constraints in the primal problem (SDP). Accordingly, our decision variables are dimensioned \( s \in \mathbb{R}^{n(n+1)/2} \) and \( y \in \mathbb{R}^m \). The dimension of the concatenated variable \( u = [s; y; x] \) is \( N = n(n+1) + m \).

The standard approach is to solve (37) using Cholesky factorization in \( O(n^3) \) flops and \( \Theta(m^2) \) storage; our goal is to solve it at lower cost using ADMM and ADMM-GMRES. Viewing (37) as an instance of (ECQP), we note that condition number scales \( \kappa = \text{cond}(W^2) = \Theta(1/\epsilon^2) \), where \( \epsilon \) is the duality gap of the outer interior-point iterate; see \[59\] and also \[55, 54\].

8.1 Implementation

When (NEWT) is treated as as an instance of (ECQP), ADMM can be applied to result in the following iterations

\[
\begin{align*}
\beta^{-1} x^{k+1} &= (\beta^{-1}W \otimes W + I)^{-1}(z + c - Ay^k - \tilde{x}^k) \quad (38a) \\
\beta^{-1} y^{k+1} &= (A^T A)^{-1}[\beta^{-1} b - A^T (s^{k+1} + \tilde{x}^k - c)] \quad (38b) \\
\beta^{-1} s^{k+1} &= \tilde{s}^k + (s^{k+1} + Ay^k - c) \quad (38c)
\end{align*}
\]

where \( \beta > 0 \) is the algorithm parameter, and \( \tilde{x} = \beta^{-1} x \) is a scaled version of the primal variable \( x \). Precomputing the eigendecomposition of the scaling matrix \( W = V \Lambda V^T \) in \( O(n^3) \) flops and \( O(n^2) \) storage, we may select the optimal parameter value \( \beta = \sqrt{\mu L} = \lambda_{\min}(W) \lambda_{\max}(W) \), and perform each \( s \)-update (38a) in \( O(n^3) \) flops via

\[
s = (\beta^{-1}W \otimes W + I)^{-1}c \iff S = V \left[ \frac{\beta}{\lambda_i \lambda_j + \beta} \right]^{n}_{i,j=1} \circ (V^T CV) V^T.
\]

Here, \( s = \text{vec } S \), \( c = \text{vec } C \), and \( \circ \) denotes the matrix element-wise product (i.e. the Hadamard product).

The \( y \)-update (38b) requires matrix-vector products with \((A^T A)^{-1}\). For many SDPs that arise from combinatorial problems, \( A^T A \) is sparse and can be factorized with very little fill-in. For others, however, the factorization can be fully-dense. An iterative or inexact scheme can be used here, but this greatly increases the per-iteration cost, and diminishes the appeal of ADMM. For a more detailed discussion of these issues, see \[58\] Rem.2], \[40\] Sec.4] and \[39\] Ch.4] and the references therein.

8.2 The SDPLIB problems

We generate instances of (37) using SeDuMi \[53\] over the 80 problems in the SDPLIB suite \[8\] with less than 700 constraints, i.e. with \( m \leq 700 \). This collection encompasses a diversity of practical semidefinite programs, and \( m \) is small enough so that the matrix \( A^T A \) may always be inverted at a reasonable cost. At the same time, the iterates \( u^k \) have dimensions up to \( N \leq 395,508 \) (for the
Figure 6: Iterations to $\epsilon = 10^{-6}$ residual convergence for the 1038 Newton direction problems described in-text: (a) & (b) ADMM vs ADMM-GMRES; (c) & (d) ADMM vs ADMM-GMRES(25)
Table 3: Solving DIMACS Problems using SeDuMi modified to compute search directions using ADMM-GMRES(25). “CPU” is total CPU time in seconds, “Iter” is the ADMM-GMRES iterations taken, and “Feas.” and “Gap” stand for “digits of feasibility” and “digits of duality gap”; see text.

| Key | Name               | m    | n    | N            | CPU  | Iter | Feas. | Gap    |
|-----|--------------------|------|------|--------------|------|------|-------|--------|
| a   | hamming_7_5_6      | 1793 | 128  | 1.83 × 10^4 | 2.86 | 300  | 6.59  | 7.92   |
| b   | hamming_8_3_4      | 16129| 256  | 8.19 × 10^4 | 11.5 | 347  | 6.37  | 8.16   |
| c   | hamming_9_5_6      | 53761| 512  | 3.16 × 10^5 | 85.2 | 422  | 6.45  | 7.89   |
| d   | hamming_9_8        | 2305 | 512  | 2.64 × 10^5 | 70.4 | 411  | 5.39  | 7.47   |
| e   | hamming_10_2       | 23041| 1024 | 1.07 × 10^6 | 256  | 328  | 5.40  | 7.94   |
| f   | hamming_11_2       | 56321| 2048 | 4.25 × 10^6 | 1262 | 217  | 3.98  | 6.36   |
| g   | toruspm3-8-50      | 512  | 512  | 2.63 × 10^5 | 201  | 1651 | 2.53  | 4.91   |
| h   | torus3-8           | 512  | 512  | 2.63 × 10^5 | 950  | 8006 | 3.64  | 6.07   |
| i   | torus3-15          | 3375 | 3375 | 1.14 × 10^7 | 26721| 3280 | 1.68  | 4.88   |
| j   | toruspm3-15-50     | 3375 | 3375 | 1.14 × 10^7 | 36782| 4387 | 2.20  | 5.42   |

8.3 The DIMACs problems

Finally, we incorporate ADMM-GMRES within SeDuMi, in order to solve large-scale SDPs from the Seventh DIMACS Implementation Challenge [44]. In other words, we modify SeDuMi to use ADMM-GMRES to compute the Newton search directions, in lieu of its internal Cholesky-based solver, while leaving the remainder of the solver unchanged. The associated (ECQP) problems are large enough to prevent the full GMRES from being used, so we restart GMRES every 25 iterations, allowing up to 1000 GMRES iterations to be performed. We quantify the accuracy of problem truss8), which is large enough for the comparisons to be realistic. For each problem, the predictor and corrector Newton subproblems with \( \kappa \leq 10^6 \) are extracted and solved using ADMM and ADMM-GMRES to \( 10^{-6} \) relative residual, on an Intel Xeon E5-2687W CPU with eight 3.10 GHz cores. The maximum number of iterations for both methods is capped at 1000.

Figure 6a shows the number of iterations to convergence. Results validate the \( O(\sqrt{\kappa}) \) figure expected of ADMM, and the \( O(\kappa^{1/4}) \) figure expected of GMRES. In fact, the multiplicative constants associated with each appear to be very similar to the results shown earlier in Figure 1. Figure 6b compares the associated CPU times with the number of iterations. The per-iteration cost of ADMM-GMRES is constant for small \( k \), but grows linearly with \( k \) beyond about 30 iterations. For many of the problems considered, the square-root factor reduction in iterations to convergence is offset by the quadratic growth in computation time, and both methods end up using a similar amount of time, despite the considerable difference in iteration count.

A practical implementation of ADMM-GMRES will require the use of a limited-memory version of GMRES. We consider the simplest approach of restarting every 25 iterations; the results are shown in Figures 6c & 6d. The method requires a factor of two more iterations to converge when compared to the usual GMRES algorithm, but the amortized per-iteration cost of the restarted variant is also a constant factor of two times higher that of basic ADMM.
Figure 7: The convergence behavior of modified SeDuMi on the DIMACS problems. Data for the keys are shown in Table 3. The dash lines indicate sublinear $O(1/k)$ and $O(1/k^2)$ convergence rates.

Each interior-point iterate $\{X, y, S\}$ using the DIMACS metrics:

\[
p_{\text{inf}} = \frac{\sqrt{\sum_{i=1}^{m} (A_i \cdot X - b_i)^2}}{1 + \|b\|_2}, \quad d_{\text{inf}} = \frac{\|\sum y_i A_i + S - C\|_F}{1 + \|C\|_F},
\]

\[
gap = \frac{|C \cdot X - b^T y|}{1 + |C \cdot X| + |b^T y|}.
\]

SeDuMi terminates either by achieving the default accuracy tolerance, or prematurely if it considers the computed search direction to be too inaccurate to make further progress.

The experiments are performed on an Intel Xeon E5-2609 v4 CPU with eight 1.70 GHz cores; Table 3 shows these results. Here, $N = n(n + 1) + m$ refers to the total number of primal-dual variables in the corresponding (ECQP) problems, and accuracy in digits of feasibility and duality gap is quantified by computing $-\log_{10}(p_{\text{inf}} + d_{\text{inf}})$ and $-\log_{10}(\text{gap})$ respectively for the final iterate. The evolution of the interior-point iterates over different interior-point iterations are plotted in Figure 7.

In these results, ADMM-GMRES with restarts is able to converge within $O(\kappa^{1/4})$ iterations for 8 out of the 10 problems. In these cases, the overall interior-point method converges with a time complexity of $O(1/\sqrt{\epsilon})$, comparable to the optimal Nesterov methods. For the remaining 2 problems, however, the method only converges in $O(\sqrt{\kappa})$ iterations, and this gives the overall method a time complexity of $O(1/\epsilon)$.

Note that the objective of this subsection is primarily to benchmark the performance of the method in a large-scale environment. Applying (nonlinear) ADMM directly to each of the problems considered in this section produces a sequence that converges at a linear rate (as opposed to its sublinear worst-case rate), so it is unlikely for our method here to substantially outperform it.

9 Conclusions and Future Work

In this paper, we have provided theoretical and numerical evidence that ADMM-GMRES can consistently converge in $O(\kappa^{1/4})$ iterations for a smooth strongly convex quadratic objective, despite a
worst-case bound of $O(\sqrt{\kappa})$ iterations. The order-of-magnitude reduction in iterations over the basic ADMM method was widely observed for both randomized examples and in the Newton subproblems for the interior-point solution of semidefinite programs. These results confirm the possibility for an over-relaxation scheme, momentum scheme, or otherwise, to significantly accelerate the convergence of ADMM, beyond the constant factor typically observed for existing schemes, and suggest the direct use of ADMM-GMRES as a practical solution method.

It remains an open question whether the same sort of acceleration can be extended to ADMM for general nonquadratic objectives. One possible approach is to use GMRES to a linearized approximation of the nonlinear fixed-point equation, in a Krylov-Newton method [11]. Alternatively, a Broyden-like secant approximation may be constructed from previous iterates, and used to extrapolate the current step, in an Anderson acceleration method [57]. Both approaches reduce to ADMM-GMRES in the case of quadratic objectives, but further work is needed to under their effectiveness.

Acknowledgments

We wish to thank José E. Serrallés for proofreading an early draft, and for assisting with the numerical results; László Miklós Lovász for discussions on random matrix theory that led to Section 5.2. A large part of the paper was written during R.Y. Zhang’s visit to UC Berkeley as postdoctoral scholar, and he would like to thank his faculty mentor Javad Lavaei for his warm accommodation.

References

[1] O. Axelsson, *Unified analysis of preconditioning methods for saddle point matrices*, Numerical Linear Algebra with Applications, 22 (2015), pp. 233–253.

[2] A. Beck and M. Teboulle, *A fast iterative shrinkage-thresholding algorithm for linear inverse problems*, SIAM journal on imaging sciences, 2 (2009), pp. 183–202.

[3] S. Becker, J. Bobin, and E. J. Candès, *Nesta: A fast and accurate first-order method for sparse recovery*, SIAM Journal on Imaging Sciences, 4 (2011), pp. 1–39.

[4] M. Benzi, M. J. Gander, and G. H. Golub, *Optimization of the Hermitian and skew-Hermitian splitting iteration for saddle-point problems*, BIT Numerical Mathematics, 43 (2003), pp. 881–900.

[5] M. Benzi and G. H. Golub, *A preconditioner for generalized saddle point problems*, SIAM Journal on Matrix Analysis and Applications, 26 (2004), pp. 20–41.

[6] M. Benzi, G. H. Golub, and J. Liesen, *Numerical solution of saddle point problems*, Acta numerica, 14 (2005), pp. 1–137.

[7] M. Benzi and V. Simoncini, *On the eigenvalues of a class of saddle point matrices*, Numerische Mathematik, 103 (2006), pp. 173–196.

[8] B. Borchers, *SDPLIB 1.2, a library of semidefinite programming test problems*, Optimization Methods and Software, 11 (1999), pp. 683–690.

[9] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, *Distributed optimization and statistical learning via the alternating direction method of multipliers*, Foundations and Trends® in Machine Learning, 3 (2011), pp. 1–122.
[10] J. H. Bramble and J. E. Pasciak, A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems, Mathematics of Computation, 50 (1988), pp. 1–17.

[11] P. N. Brown and Y. Saad, Convergence theory of nonlinear Newton–Krylov algorithms, SIAM Journal on Optimization, 4 (1994), pp. 297–330.

[12] D. Davis and W. Yin, Faster convergence rates of relaxed Peaceman-Rachford and ADMM under regularity assumptions, arXiv preprint arXiv:1407.5210, (2014).

[13] C. de Boor and J. R. Rice, Extremal polynomials with application to Richardson iteration for indefinite linear systems, SIAM Journal on Scientific and Statistical Computing, 3 (1982), pp. 47–57.

[14] W. Deng and W. Yin, On the global and linear convergence of the generalized alternating direction method of multipliers, Journal of Scientific Computing, (2012), pp. 1–28.

[15] D. L. Donoho and X. Huo, Uncertainty principles and ideal atomic decomposition, Information Theory, IEEE Transactions on, 47 (2001), pp. 2845–2862.

[16] T. A. Driscoll, K.-C. Toh, and L. N. Trefethen, From potential theory to matrix iterations in six steps, SIAM review, 40 (1998), pp. 547–578.

[17] L. Elsner and M. Pardekooper, On measures of nonnormality of matrices, Linear Algebra and its Applications, 92 (1987), pp. 107–123.

[18] M. Embree, How descriptive are GMRES convergence bounds?, (1999).

[19] V. Faber and T. Manteuffel, Necessary and sufficient conditions for the existence of a conjugate gradient method, SIAM Journal on Numerical Analysis, 21 (1984), pp. 352–362.

[20] B. Fischer, A. Ramage, D. J. Silvester, and A. J. Wathen, Minimum residual methods for augmented systems, BIT Numerical Mathematics, 38 (1998), pp. 527–543.

[21] M. Fortin and R. Glowinski, Augmented Lagrangian methods, (1983).

[22] D. Gabay and B. Mercier, A dual algorithm for the solution of nonlinear variational problems via finite element approximation, Computers & Mathematics with Applications, 2 (1976), pp. 17–40.

[23] E. Ghadimi, A. Teixeira, I. Shames, and M. Johansson, Optimal parameter selection for the alternating direction method of multipliers (ADMM): quadratic problems, Automatic Control, IEEE Transactions on, 60 (2015), pp. 644–658.

[24] P. Giselsson and S. Boyd, Diagonal scaling in Douglas-Rachford splitting and ADMM, in Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on, IEEE, 2014, pp. 5033–5039.

[25] R. Glowinski and A. Marroco, Sur l’approximation, par éléments finis d’ordre un, et la résolution, par pénalisation-dualité d’une classe de problèmes de Dirichlet non linéaires, Revue française d’automatique, informatique, recherche opérationnelle. Analyse numérique, 9 (1975), pp. 41–76.
[26] T. Goldstein, B. O’Donoghue, S. Setzer, and R. Baraniuk, *Fast alternating direction optimization methods*, SIAM Journal on Imaging Sciences, 7 (2014), pp. 1588–1623.

[27] G. H. Golub and C. Greif, *On solving block-structured indefinite linear systems*, SIAM Journal on Scientific Computing, 24 (2003), pp. 2076–2092.

[28] A. Greenbaum, *Iterative methods for solving linear systems*, vol. 17, Siam, 1997.

[29] P. Henrici, *Bounds for iterates, inverses, spectral variation and fields of values of non-normal matrices*, Numerische Mathematik, 4 (1962), pp. 24–40.

[30] M. Kadkhodaie, K. Christakopoulou, M. Sanjabi, and A. Banerjee, *Accelerated alternating direction method of multipliers*, in Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 497–506.

[31] A. Kalbat and J. Lavaei, *A fast distributed algorithm for decomposable semidefinite programs*, in Decision and Control (CDC), 2015 IEEE 54th Annual Conference on, IEEE, 2015, pp. 1742–1749.

[32] C. Keller, N. I. Gould, and A. J. Wathen, *Constraint preconditioning for indefinite linear systems*, SIAM Journal on Matrix Analysis and Applications, 21 (2000), pp. 1300–1317.

[33] R. Madani, A. Kalbat, and J. Lavaei, *ADMM for sparse semidefinite programming with applications to optimal power flow problem*, in Decision and Control (CDC), 2015 IEEE 54th Annual Conference on, IEEE, 2015, pp. 5932–5939.

[34] N. M. Nachtigal, L. Reichel, and L. N. Trefethen, *A hybrid GMRES algorithm for nonsymmetric linear systems*, SIAM Journal on Matrix Analysis and Applications, 13 (1992), pp. 796–825.

[35] A. Nemirovskii, D. B. Yudin, and E. R. Dawson, *Problem complexity and method efficiency in optimization*, Wiley, 1983.

[36] Y. Nesterov, *A method of solving a convex programming problem with convergence rate $O(1/k^2)$*, in Soviet Mathematics Doklady, vol. 27, 1983, pp. 372–376.

[37] ———, *Introductory lectures on convex optimization*, vol. 87, Springer Science & Business Media, 2004.

[38] Y. Nesterov, *Smooth minimization of non-smooth functions*, Mathematical programming, 103 (2005), pp. 127–152.

[39] R. Nishihara, L. Lessard, B. Recht, A. Packard, and M. I. Jordan, *A general analysis of the convergence of ADMM*, arXiv preprint arXiv:1502.02009, (2015).

[40] B. O’Donoghue, E. Chu, N. Parikh, and S. Boyd, *Conic optimization via operator splitting and homogeneous self-dual embedding*, Journal of Optimization Theory and Applications, 169 (2016), pp. 1042–1068.

[41] Y. Ouyang, Y. Chen, G. Lan, and E. Pasiliao Jr, *An accelerated linearized alternating direction method of multipliers*, SIAM Journal on Imaging Sciences, 8 (2015), pp. 644–681.

[42] S. K. Pakazad, A. Hansson, and M. S. Andersen, *Distributed interior-point method for loosely coupled problems*, IFAC Proceedings Volumes, 47 (2014), pp. 9587–9592.
[43] S. K. Pakazad, A. Hansson, M. S. Andersen, and A. Rantzer, Distributed semidefinite programming with application to large-scale system analysis, arXiv preprint arXiv:1504.07755, (2015).

[44] G. Pataki and S. Schmieta, The DIMACS library of semidefinite-quadratic-linear programs, tech. rep., Tech. Rep. Preliminary draft, Computational Optimization Research Center, Columbia University, New York, 2002.

[45] P. Patrinos, L. Stella, and A. Bemporad, Douglas-Rachford splitting: Complexity estimates and accelerated variants, in Decision and Control (CDC), 2014 IEEE 53rd Annual Conference on, IEEE, 2014, pp. 4234–4239.

[46] T. J. Rivlin, The Chebyshev Polynomials: From Approximation Theory to Algebra and Number Theory, John Wiley & Sons, 1974.

[47] M. Rudelson and R. Vershynin, Sampling from large matrices: An approach through geometric functional analysis, Journal of the ACM (JACM), 54 (2007), p. 21.

[48] T. Rusten and R. Winther, A preconditioned iterative method for saddlepoint problems, SIAM Journal on Matrix Analysis and Applications, 13 (1992), pp. 887–904.

[49] Y. Saad, Iterative methods for sparse linear systems, Siam, 2003.

[50] Y. Saad and M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM Journal on scientific and statistical computing, 7 (1986), pp. 856–869.

[51] V. Simoncini, Block triangular preconditioners for symmetric saddle-point problems, Applied Numerical Mathematics, 49 (2004), pp. 63–80.

[52] V. Simoncini and M. Benzi, Spectral properties of the Hermitian and skew-Hermitian splitting preconditioner for saddle point problems, SIAM Journal on Matrix Analysis and Applications, 26 (2004), pp. 377–389.

[53] J. F. Sturm, Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones, Optimization methods and software, 11 (1999), pp. 625–653.

[54] K.-C. Toh, Solving large scale semidefinite programs via an iterative solver on the augmented systems, SIAM Journal on Optimization, 14 (2004), pp. 670–698.

[55] K.-C. Toh and M. Kojima, Solving some large scale semidefinite programs via the conjugate residual method, SIAM Journal on Optimization, 12 (2002), pp. 669–691.

[56] L. N. Trefethen and M. Embree, Spectra and pseudospectra: the behavior of nonnormal matrices and operators, Princeton University Press, 2005.

[57] H. F. Walker and P. Ni, Anderson acceleration for fixed-point iterations, SIAM Journal on Numerical Analysis, 49 (2011), pp. 1715–1735.

[58] Z. Wen, D. Goldfarb, and W. Yin, Alternating direction augmented Lagrangian methods for semidefinite programming, Mathematical Programming Computation, 2 (2010), pp. 203–230.
[59] M. H. Wright, *Interior methods for constrained optimization*, Acta numerica, 1 (1992), pp. 341–407.

[60] Y. Zheng, G. Fantuzzi, A. Papachristodoulou, P. Goulart, and A. Wynn, *Fast ADMM for semidefinite programs with chordal sparsity*, arXiv preprint arXiv:1609.06068, (2016).

[61] W. Zulehner, *Analysis of iterative methods for saddle point problems: a unified approach*, Mathematics of computation, 71 (2002), pp. 479–505.