Local Linear Convergence of ADMM on Quadratic or Linear Programs

Daniel Boley
University of Minnesota
Minneapolis, MN 55455 USA

Abstract

In this paper, we analyze the convergence of the Alternating Direction Method of Multipliers (ADMM) as a matrix recurrence for the particular case of a quadratic program or a linear program. We identify a particular combination of the vector iterates in the standard ADMM iteration that exhibits almost monotonic convergence. We present an analysis which indicates the convergence depends on the eigenvalues of a particular matrix operator. The theory predicts that ADMM should exhibit linear convergence when close enough to the optimal solution, but when far away can exhibit slow "constant step" convergence. This is illustrated with a convergence trace from linear program.

Keywords: Alternating Direction Method of Multipliers, ADMM, linear programming, quadratic programming.

AMS Classification: 65K05, 90C05, 90C20.

1 Introduction

The alternating direction method of multipliers (ADMM) is a popular method for solving large scale convex optimization problems [2, 3, 8, 12, 14, 15, 17, 21, 30]. The simplicity of the method has made it an effective method in many large scale applications, particularly when the functions and constraints are separable [11, 22, 28]. As a result of its flexibility and simplicity, it has been used in many diverse application areas such as image processing and restoration, consensus computations, sparse regression and recovery applications, and many others (see e.g., [1, 4, 5, 9, 6, 10, 18, 24, 26, 27, 29] and references therein). An extensive recent survey of the current state of the art of ADMM from a computational point of view can be found in [4], including many other diverse applications of this method. In the present paper we present some results regarding the convergence of ADMM on a model quadratic or linear program:

$$\min {}^1\!/\!_2\mathbf{x}^T Q\mathbf{x} + \mathbf{c}^T\mathbf{x} \text{ s.t. } A\mathbf{x} = \mathbf{b},\ \mathbf{x} \geq 0, \tag{1}$$

where Q is symmetric positive semi-definite, and $Q = 0$ for a linear program. Existing convergence results [4, 8] for ADMM include those of the form of a bound on the sum of the norms of differences between consecutive iterates during the entire course of the algorithm, or else a global sublinear bound on the rate of convergence [1, 7, 16, 23]. Such bounds lead to the conclusion [4, 8, 7] that the norms of the residuals at each step converges to zero, starting with any initial iterate, as long as a finite optimal solution exists. Such results imply a powerful global convergence property for

ADMM, but says little about how fast it converges or how regular is the convergence behavior. A later paper [7] studied specifically linear programs, and included a global linear convergence result, but it is easily observed in practice that convergence can be slow. Several recent works report a global convergence rate of $O(1/k)$, or $O(1/k^2)$ if accelerated, on variations of general convex problems [23, 1, 16], where k is the iteration number. These bounds guarantee sub-linear global convergence under certain general conditions on the sequence of smoothing and acceleration parameters, but become less tight as k grows. Their methods of acceleration involve some sort of overrelaxation using linear combinations of old iterates. Like [1] we analyze the operator that maps the iterate at one iteration to the iterate at the next iteration, but unlike [1] we limit ourselves to problems in which we can write this operator explicitly as a matrix amenable to a detailed spectral analysis. In [16], the authors explicitly handle general linear equality constraints, and examine the linear mapping from one iteration to the next as a matrix operator. In our analysis, we carry the ADMM iteration using a novel vector recombination of the original iterates and examine the linear mapping on this particular combination.

In this paper we restrict our attention to linear and quadratic programs, as opposed to general convex problems, and examine a particular splitting in which the inequality and equality constraints are separated. We focus on the less ambitious problem of local convergence, as opposed to global convergence. We analyze the convergence of ADMM as a matrix recurrence for the particular case of a quadratic program or a linear program. We identify a particular combination of the vector iterates in the standard ADMM iteration that exhibits almost monotonic convergence to a final solution, and show that the rate at which this convergence occurs depends on the eigenvalues of a particular matrix operator within an open neighborhood of the optimal solution, if it exists. Under normal circumstances, the theory predicts that ADMM should pass through several stages or "regimes", many of which are taking constant steps, but finally reaching a regime of linear convergence when close enough to the optimal solution. The theory suggests that any acceleration scheme would be more effective if it took account of the particular regime currently in effect.

In this paper, unless otherwise specified, all vector and matrix norms are the "2-norms" (e.g., the largest singular value for a matrix). For symmetric matrices, the matrix 2-norm is the same as the spectral radius (largest absolute value of any eigenvalue), hence we use those interchangeably for symmetric matrices.

2 ADMM Iteration

The ADMM iteration for (1) can be constructed based on finding critical points for the augmented Lagrangian function

$$\mathcal{L}_\rho(\mathbf{x},\mathbf{z},\mathbf{y}) = {}^1\!/_2\mathbf{x}^TQ\mathbf{x} + \mathbf{c}^T\mathbf{x} + g(\mathbf{z}) + \mathbf{y}^T(\mathbf{x}-\mathbf{z}) + {}^1\!/_2\rho\|\mathbf{x}-\mathbf{z}\|_2^2, \quad \text{s.t. } A\mathbf{x}=\mathbf{b}, \tag{2}$$

where ρ is a relaxation parameter to be chosen by the user, and $g(\mathbf{z})$ is the indicator function for the non-negative orthant: $g(\mathbf{z}) = 0$ if $\mathbf{z} \geq 0$, $g(\mathbf{z}) = \infty$ if any component of $\mathbf{z}$ is negative. We adopt a splitting in which ADMM first minimizes (2) with respect to $\mathbf{x}$, then with respect to $\mathbf{z}$, and then performs one ascent step on the Lagrange multipliers $\mathbf{y} = -\rho\mathbf{u}$. Using hat $\widehat{()}$ to denote the new values (to reduce clutter), the resulting ADMM iteration (with no acceleration) consists of repeating the following steps until convergence, starting with a pair of vectors $\mathbf{z}, \mathbf{u}$ and a fixed smoothing parameter ρ. The output vectors from each pass are denoted $\widehat{\mathbf{z}}, \widehat{\mathbf{u}}$.

```
Algorithm 1: One Pass of ADMM
Start with $\mathbf{z}, \mathbf{u}$.
    1. Solve $\begin{pmatrix} Q+\rho I & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \boldsymbol{\nu} \end{pmatrix} = \begin{pmatrix} \rho(\mathbf{z}-\mathbf{u})-\mathbf{c} \\ \mathbf{b} \end{pmatrix}$ for $\mathbf{x}, \boldsymbol{\nu}$.
    2. Set $\widehat{\mathbf{z}} = \max\{0, \mathbf{x}+\mathbf{u}\}$.
    3. Set $\widehat{\mathbf{u}} = \mathbf{u}+\mathbf{x}-\widehat{\mathbf{z}}$.
Result is $\widehat{\mathbf{z}}, \widehat{\mathbf{u}}$ for next iteration.
```

Lemma 1. *After every iteration, the vectors* $\widehat{\mathbf{z}}, \widehat{\mathbf{u}}$ *satisfy*
a. $\widehat{\mathbf{z}} \geq 0$,
b. $\widehat{\mathbf{u}} \leq 0$,
c. $\widehat{z}_i \cdot \widehat{u}_i = 0$, $\forall i$ *(a complementarity condition).*
d. $\mathbf{x}$ *satisfies the equality constraints* $A\mathbf{x} = \mathbf{b}$.

Proof: In Algorithm 1 step 2: if $x_i + u_i \geq 0$ then $\widehat{z}_i = x_i + u_i \geq 0$ and $\widehat{u}_i = u_i + x_i - (x_i + u_i) = 0$. If $x_i + u_i \leq 0$ then $\widehat{z}_i = 0$ and $\widehat{u}_i = u_i + x_i \leq 0$. Point d follows directly from step 1. ∎

So we can assume $\mathbf{z}, \mathbf{u}$ satisfy these conditions at the beginning of each iteration, including the first iteration if we start with $\mathbf{z} = \mathbf{u} = 0$.

Lemma 2. *If in Algorithm 1 step 1* $\mathbf{x} = \mathbf{z}$, *and* $\mathbf{z}, \mathbf{u}$ *satisfy the complementarity condition (c.) in the previous lemma, then* $\mathbf{z} = \widehat{\mathbf{z}}$, *and* $\mathbf{x}$, $\boldsymbol{\nu}$, $\mathbf{y} = -\rho\mathbf{u}$ *If* $\mathbf{x} = \mathbf{z} = \widehat{\mathbf{z}}$, *then* $\mathbf{x}$, $\boldsymbol{\nu}$, $\mathbf{y} = -\rho\mathbf{u}$ *satisfy the first order KKT conditions for (1), where* $\boldsymbol{\nu}$, $\mathbf{y}$ *are the Lagrange multipliers for the equality and inequality constraints, respectively.*

Proof: Let $x_i = z_i$, $\forall i$. By the complementarity condition, either $z_i = x_i = 0$ or $u_i = 0$. In the latter case, $x_i + u_i = x_i \geq 0$ so $\widehat{z}_i = x_i$. In the former case, $x_i + u_i = u_i \leq 0$ so $\widehat{z}_i = 0 = x_i$. In either case $\widehat{u}_i = u_i$. From step 1: $Q\mathbf{x} + \rho\mathbf{x} + A^T\boldsymbol{\nu} = \rho\mathbf{z} - \rho\mathbf{u} - \mathbf{c}$, which simplifies to $Q\mathbf{x} + A^T\boldsymbol{\nu} = \mathbf{y} - \mathbf{c}$. This, combined with the previous lemma, form the first order KKT conditions. ∎

Theorem 3. *If (1) has an optimal solution* $\mathbf{x}^*$ *with multipliers* $\boldsymbol{\nu}^*$, $\mathbf{y}^*$ *(corresponding to the equality and inequality constraints, respectively) such that* $\mathbf{x}^*$, $\mathbf{z}^* = \mathbf{x}^*$, $\mathbf{y}^*$ *are a saddle point for the unaugmented Lagrangian* $\mathcal{L}_0(\mathbf{x}, \mathbf{z}, \mathbf{y})$, *then Algorithm 1 converges to a fixed point satisfying Lemma 2.*

Outline of proof: For the purpose of this proof, we use the notation $\mathbf{x}^k, \mathbf{z}^k, \mathbf{u}^k$ to denote the iterates $\mathbf{x}, \widehat{\mathbf{z}}, \widehat{\mathbf{u}}$ at the k-th iteration of Algorithm 1, and also $\mathbf{y}^k = -\rho\mathbf{u}^k$, $\mathbf{r}^k = \mathbf{x}^k - \underline{z}^k = \mathbf{y}^k - \mathbf{y}^{k-1}$. From [4],

$$\rho \sum_{k=1}^{\infty} (\|\mathbf{r}^k\|_2^2 + \|\mathbf{z}^k - \mathbf{z}^{k-1}\|_2^2) \leq \|\mathbf{y}^0 - \mathbf{y}^*\|_2^2/\rho + \|\mathbf{z}^0 - \mathbf{z}^*\|_2^2 \cdot \rho.$$

This implies the residuals $\mathbf{r}^k$ and $\mathbf{z}^k - \mathbf{z}^{k-1}$ are converging to zero. For details we refer to [4, 8]. ∎
This theorem says little on the local behavior of the algorithm. The local behavior is the focus of the rest of this paper.

3 Auxiliary Variables with Local Monotonic Behavior

Instead of carrying the iteration using variables $\mathbf{z}, \mathbf{u}$, we use two auxiliary variables to carry the iteration. One variable turns out to exhibit almost monotonic convergence locally around a fixed point, and the other is simply a binary vector of flags marking which inequality constraints are active.

Let $\mathbf{v} = -\mathbf{u}$, let $\mathbf{w} = \mathbf{z} + \mathbf{v} = \mathbf{z} - \mathbf{u}$, and let $\mathbf{d}$ be a vector of flags such that $d_i = -1$ iff $u_i \neq 0$, otherwise $d_i = +1$. If $D = \textsc{Diag}(\mathbf{d})$ (the diagonal matrix with elements of vector $\mathbf{d}$ on the diagonal), then ${}^1\!/_2(I-D)\mathbf{w} = \mathbf{v} = -\mathbf{u}$ and ${}^1\!/_2(I+D)\mathbf{w} = \mathbf{z}$. The flags indicate which inequality constraints are actively enforced on $\mathbf{z}$ at each iteration. Then we can write ADMM steps 2 and 3 elementwise as follows (again using hat $\widehat{(\;)}$ to denote the new values):

$$\begin{aligned} \widehat{z}_i &= \begin{cases} 0 & \text{if} \quad x_i - v_i < 0 \\ x_i - v_i & \text{if} \quad x_i - v_i \geq 0 \end{cases} \\ \widehat{v}_i &= v_i + \widehat{z}_i - x_i = v + \max\{0, x_i - v_i\} - x_i \\ &= \begin{cases} v_i - x_i & \text{if} \quad x_i - v_i < 0 \\ 0 & \text{if} \quad x_i - v_i \geq 0 \end{cases} \end{aligned} \tag{3}$$

and so (using $v_i = {}^1\!/_2(1-d_i)w_i$)

$$\begin{aligned} &\widehat{d}_i = \begin{cases} -1 & \text{if} \quad x_i - {}^1\!/_2(1-d_i)w_i \leq 0 \\ +1 & \text{if} \quad x_i - {}^1\!/_2(1-d_i)w_i > 0 \end{cases} \\ &\widehat{w}_i = |x_i - {}^1\!/_2(1-d_i)w_i| = \widehat{d}_i(x_i - {}^1\!/_2(1-d_i)w_i) = \widehat{d}_i(x_i - {}^1\!/_2(1-d_i)w_i) \end{aligned} \tag{4}$$

where $\widehat{d}_i = \pm 1$ to match the effect of the absolute value sign. In matrix form, the modified ADMM iteration using the new variables can be written as:

Algorithm 2: One Pass of Modified ADMM

Start with $\mathbf{w}, D$.

1. Solve $\begin{pmatrix} Q/\rho + I & A^T/\rho \\ A & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \boldsymbol{\nu} \end{pmatrix} = \begin{pmatrix} \mathbf{w} - \mathbf{c}/\rho \\ \mathbf{b} \end{pmatrix}$ for $\mathbf{x}, \boldsymbol{\nu}$.
2. Set $\widehat{\mathbf{w}} = |\mathbf{x} - {}^1\!/_2(I-D)\mathbf{w}| = \widehat{D}(\mathbf{x} - {}^1\!/_2(I-D)\mathbf{w})$, where $D = \textsc{Diag}(\mathbf{d})$, and the new $\widehat{D} = \textsc{Diag}(\pm 1, \ldots, \pm 1)$ to match the effect of taking absolute values.

Result is $\widehat{\mathbf{w}}, \widehat{D}$ for next iteration.

Next, we focus on step 1 and find an explicit formula for $\mathbf{x}$ in terms of $\mathbf{w}$. The ultimate goal is to eliminate $\mathbf{x}, \boldsymbol{\nu}$ entirely from the formulas. We do this by explicitly inverting the matrix in Algorithm 2 step 1.

$$\begin{aligned} \begin{pmatrix} \mathbf{x} \\ \boldsymbol{\nu} \end{pmatrix} &= \begin{pmatrix} Q/\rho + I & A^T/\rho \\ A & 0 \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{w} - \mathbf{c}/\rho \\ \mathbf{b} \end{pmatrix} \\ &= \begin{pmatrix} N & RA^TS \\ \rho SAR & -\rho S \end{pmatrix} \begin{pmatrix} \mathbf{w} - \mathbf{c}/\rho \\ \mathbf{b} \end{pmatrix}, \end{aligned} \tag{5}$$

where $R = (Q/\rho + I)^{-1}$ is the resolvent of Q, $S = (ARA^T)^{-1}$ is the inverse of the Schur complement, and $N = R - RA^TSAR$. The operator N satisfies the following spectral properties.

Lemma 4. *The operator $N = R - RA^TSAR$ is positive semi-definite and $\|N\|_2 \le \|R\|_2 \le 1$. If Q is strictly positive definite, then also $\|R\|_2 < 1$.*

Proof:

1. For symmetric matrices, the 2-norm is the same as the spectral radius, so we can use them interchangeably [19]. If the eigenvalues of Q are $0 \boxed{\le} \lambda_n \le \cdots \le \lambda_1$, then the eigenvalues of R are $0 < (\lambda_1/\rho+1)^{-1} \le \cdots \le (\lambda_n/\rho+1)^{-1} \boxed{\le} 1$. Hence $\|R\|_2 \boxed{\le} 1$. The inequalities in the boxes are strict iff Q is strictly positive definite.
2. Let $LL^T = R$ be its Cholesky factorization, and let $\tilde{A} = AL$. Then we can write $N = R - RA^TSAR = L[I - \tilde{A}^T(\tilde{A}\tilde{A}^T)^{-1}\tilde{A}]L^T = L[\cdots]L^T$ where the part within the square brackets is an orthogonal projector with eigenvalues 0 or 1. The matrix N is positive semi-definite because $\mathbf{x}^TL[\cdots]L^T\mathbf{x} \ge 0$ for any vector $\mathbf{x}$. Hence the eigenvalues of N are the same as the eigenvalues of $L^TL[\cdots]$ (where $\cdots$ stands for the orthogonal projector), and so we have $\|N\|_2 = \|L^TL[\cdots]\|_2 \le \|L^TL\|_2 = \|LL^T\|_2 = \|R\|_2$.

∎

Remark 5. *We remark that in the case of a linear program, $Q = 0$, the recurrence matrix $N = I - A^+A$ reduces to the orthogonal projector onto the nullspace of A (as noted in [7]), and the constant vector $\mathbf{h}$ can be written $\mathbf{h} = A^+\mathbf{b} - N\mathbf{c}/\rho$, where A^+ is the Moore-Penrose pseudo-inverse of A. In this case, N is guaranteed to have only eigenvalues 0 and 1 with various multiplicities. We also remark that in this case, the matrix N is completely independent of ρ.*

So we can use (5) to write the first ADMM step as

$$\mathbf{x} = N\mathbf{w} - N\mathbf{c}/\rho + RA^TS\mathbf{b} = N\mathbf{w} + \mathbf{h}, \tag{6}$$

for a constant vector $\mathbf{h} = RA^TS\mathbf{b} - N\mathbf{c}/\rho$, dropping the vector $\boldsymbol{\nu}$.

4 ADMM as a Matrix Recurrence

Next we focus on the entire ADMM iteration. The input at each pass consists of the vector $\mathbf{w}$ and the diagonal matrix of flags D. Substituting (6) into step 1 of Algorithm 2, we can reduce the entire ADMM pass to the following simple procedure.

```
Algorithm 3: One Pass of Reduced ADMM
Start with $\mathbf{w}, D$.
 1. $\widehat{D} = \textsc{Diag}(\textsc{sign}(N - \tfrac{1}{2}(I-D))\mathbf{w} + \mathbf{h})$
 2. $\widehat{\mathbf{w}} = \widehat{D}(N - \tfrac{1}{2}(I-D))\mathbf{w} + \widehat{D}\mathbf{h}$
Result is $\widehat{\mathbf{w}}, \widehat{D}$ for next iteration.
```

It is seen that $M = \widehat{D}(N - \tfrac{1}{2}(I-D))$ plays a critical role in the convergence of this procedure. Hence we now establish some spectral properties of $\widehat{D}(N - \tfrac{1}{2}(I-D))$. First we recall some theory relating the spectral radius to the matrix norm from [20, 25].

Theorem 6. *Let $\rho(A)$ denote the spectral radius of an arbitrary matrix A, and let $\|A\|_p = \max_{\|\mathbf{x}\|_p=1}\|A\mathbf{x}\|_p$ denote a matrix norm induced by some arbitrary vector norm $\|\cdot\|_p$. Then*

1. *For any induced matrix norm, $\rho(A) \le \|A\|_p$.*

2. *There exists an induced matrix norm such that $\|A\|_p = \rho(A)$ if and only if for any eigenvalue λ such that $|\lambda| = \rho(A)$, the algebraic and geometric multiplicities of λ are the same (all Jordan blocks for λ are 1×1). Such a matrix is said to be a member of* Class M.

3. *For any normal matrix A (i.e., satisfying $AA^T = A^TA$), $\rho(A) = \|A\|_2$.*

4. *If a λ such that $|\lambda| = \rho(A)$ has a Jordan block of dimension larger than 1 (the geometric multiplicity is strictly less than the algebraic multiplicity), then for any $\epsilon > 0$ there exists an induced matrix norm such that $\rho(A) < \|A\|_p \le \rho(A) + \epsilon$.*

Proof: The proof is based on the Jordan Canonical Form, or the Schur form for normal matrices. For details see [25, sec. 1.3] or [20, sec. 2.3]. ∎

Lemma 7. $\|M\|_2 = \|\widehat{D}(N - {}^1\!/_2(I-D))\|_2 \le 1$. *Any eigenvalues of $M = \widehat{D}(N - {}^1\!/_2(I-D))$ on the unit circle must have a complete set of eigenvectors (no Jordan blocks larger than 1×1).*
Proof: Proved as part of the proof of the next lemma. ∎

A special case occurs when $\widehat{D} = D$, i.e., the set of active inequality constraints enforced on the vector iterate $\mathbf{z}$ does not change from one iteration to the next.
Lemma 8. *Using the same notation as Lemma 7, if $\widehat{D} = D$ (the flags remain unchanged), then all eigenvalues of $D(N - {}^1\!/_2(I-D))$ must lie in the closed disk in the complex plane with center ${}^1\!/_2$ and radius ${}^1\!/_2$, denoted $\mathcal{D}({}^1\!/_2, {}^1\!/_2)$. The only possible eigenvalue on the unit circle is $+1$, and if present must have a complete set of eigenvectors. In the case of a linear program, $Q = 0$, N is an orthogonal projector, and all the eigenvalues of $M = D(N - {}^1\!/_2(I-D))$ lie on the boundary of $\mathcal{D}({}^1\!/_2, {}^1\!/_2)$.*
Proof: Returning to Lemma 7, we have $M = \widehat{D}(N - {}^1\!/_2(I-D)) = \widehat{D}DD(N - {}^1\!/_2(I-D)) = \widehat{D}D(D(N - I/2) + I/2)$. Here we have used the fact $D^2 = I$. From Lemma 4, N is symmetric positive semidefinite with norm at most 1 (strictly less than 1 if Q is strictly positive definite). Hence the eigenvalues of N are in the interval $[0, 1]$, where the right end will be open if Q is strictly positive definite. Hence we have the following

a. The eigenvalues of N are in $[0, 1]$.

b. The eigenvalues of $N - I/2$ are in $[-{}^1\!/_2, +{}^1\!/_2]$.

c. $\|N - I/2\|_2 \le {}^1\!/_2$.

d. $\|D(N - I/2)\|_2 \le {}^1\!/_2$, and $\|D(N - I/2) + I/2\|_2 = \|D(N - {}^1\!/_2(I-D))\|_2 \le 1$.

e. The eigenvalues of $D(N - I/2)$ lie in the closed circular disk on the complex plane with center 0 and radius ${}^1\!/_2$, denoted $\mathcal{D}(0, {}^1\!/_2)$.

f. The eigenvalues of $D(N - I/2) + I/2$ lie in the disk $\mathcal{D}({}^1\!/_2, {}^1\!/_2)$, which is entirely in the open right half plane plus the origin.

g. In particular, if $D(N - {}^1\!/_2(I-D))$ has any eigenvalue with absolute value $1 = \|D(N - {}^1\!/_2(I-D))\|_2$, then that eigenvalue must be exactly 1 and must have a complete set of eigenvectors (no non-trivial Jordan blocks).

h. The above proves Lemma 8 for the case $\widehat{D} = D$. In the general case of Lemma 7, $\|M\|_2 = \|\widehat{D}D[D(N - \nicefrac{1}{2}(I{-}D))]\|_2 \leq \|\widehat{D}D\|_2 \cdot \|D(N - \nicefrac{1}{2}(I{-}D))\|_2 \leq 1$, since $\widehat{D}D$ is a unitary matrix.

In the case of a linear program, we have the following.

i. $Q = 0$ in (1), N is an orthogonal projector (see Remark 5), so that N is symmetric and $N^2 = N$. Hence $2(N - I/2)$ is an orthogonal matrix: $2(N - I/2)^T 2(N - I/2) = 4(N^2 - N + I/4) = I$.

j. $2D(N - I/2)$ is also an orthogonal matrix since it is the product of orthogonal matrices.

k. All the eigenvalues of $2D(N - I/2)$ lie on the unit circle. Hence all the eigenvalues of $M = D(N - I/2) + I/2$ lie on the boundary of $\mathcal{D}(\nicefrac{1}{2}, \nicefrac{1}{2})$.

$\blacksquare$

5 Convergence properties.

Now we write the heart of Algorithm 3 as a homogeneous matrix recurrence. We use this form to characterize its convergence properties. Step 2 of Algorithm 3 is written as follows:

$$\begin{pmatrix} \widehat{\mathbf{w}} \\ 1 \end{pmatrix} = \mathbf{M} \begin{pmatrix} \mathbf{w} \\ 1 \end{pmatrix} = \begin{pmatrix} M & \widehat{D}\mathbf{h} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{w} \\ 1 \end{pmatrix} \begin{pmatrix} \widehat{D}(N - \nicefrac{1}{2}(I{-}D)) & \widehat{D}\mathbf{h} \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \mathbf{w} \\ 1 \end{pmatrix}, \tag{7}$$

where $\mathbf{h} = RA^T S\mathbf{b} - N\mathbf{c}/\rho$ is as in (6). The eigenvalues of the augmented matrix $\mathbf{M}$ in (7) consist of those of M plus an extra eigenvalue equal to 1. If M already has an eigenvalue equal to 1, then the extra eigenvalue 1 might or might not add a corresponding eigenvector. We state two lemmas regarding the spectral properties of $\mathbf{M}$.

Lemma 9. *Let* $\mathbf{M} = \begin{pmatrix} M & \mathbf{p} \\ 0 & 1 \end{pmatrix}$ *be any block upper triangular matrix with a* 1×1 *lower right block, and suppose the upper left block* M *either has no eigenvalue equal to 1 or the eigenvalue 1 of* M *has a complete set of eigenvectors. Suppose* $\begin{pmatrix} \mathbf{w} \\ 1 \end{pmatrix} = \mathbf{M} \begin{pmatrix} \mathbf{w} \\ 1 \end{pmatrix}$. *Then* $\mathbf{M}$ *has an eigenvalue 1 and this eigenvalue has no non-trivial Jordan block.*

Proof: We can block diagonalize the upper left block $M = P \begin{pmatrix} \tilde{M}_{11} & 0 \\ 0 & I \end{pmatrix} P^{-1}$ with a suitable transformation matrix P, where $\tilde{M}_{11}$ has no eigenvalue equal to 1. Then

$$\begin{pmatrix} P^{-1} & 0 \\ 0 & 1 \end{pmatrix} \mathbf{M} \begin{pmatrix} P & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \tilde{M}_{11} & 0 & \tilde{\mathbf{p}}_1 \\ 0 & I & \tilde{\mathbf{p}}_2 \\ 0 & 0 & 1 \end{pmatrix},$$

Then we must have for $\begin{pmatrix} \tilde{\mathbf{w}}_1 \\ \tilde{\mathbf{w}}_2 \end{pmatrix} = P^{-1}\mathbf{w}$:

$$\begin{pmatrix} \tilde{\mathbf{w}}_1 \\ \tilde{\mathbf{w}}_2 \\ 1 \end{pmatrix} = \begin{pmatrix} P^{-1} & 0 \\ 0 & 1 \end{pmatrix} \mathbf{M} \begin{pmatrix} P & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{w}}_1 \\ \tilde{\mathbf{w}}_2 \\ 1 \end{pmatrix} = \begin{pmatrix} \tilde{M}_{11} & 0 & \tilde{\mathbf{p}}_1 \\ 0 & I & \tilde{\mathbf{p}}_2 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{w}}_1 \\ \tilde{\mathbf{w}}_2 \\ 1 \end{pmatrix} = \begin{pmatrix} \tilde{M}_{11}\tilde{\mathbf{w}}_1 + \tilde{\mathbf{p}}_1 \\ \tilde{\mathbf{w}}_2 + \tilde{\mathbf{p}}_2 \\ 1 \end{pmatrix}$$

This implies that $\tilde{\mathbf{p}}_2 = 0$, i.e., the eigenvalue 1 of the entire matrix $\mathbf{M}$ has a complete set of eigenvectors. $\blacksquare$

Lemma 10. *Let $\mathbf{M}$ be the matrix in (7) and assume $\widehat{D} = D$ is a flag matrix of the form* $\textsc{Diag}(\pm 1, \ldots, \pm 1)$*. Suppose* $\begin{pmatrix} \mathbf{w}_1 \\ 1 \end{pmatrix}$ *is an eigenvector corresponding to eigenvalue 1 of the matrix $\mathbf{M}$ and furthermore suppose $\mathbf{w} \geq 0$. Then the primal variables $\mathbf{x} = \mathbf{z} = {}^1\!/\!_2(I+D)\mathbf{w}$ and dual variables $\mathbf{y} = \rho\mathbf{v} = {}^\rho\!/\!_2(I-D)\mathbf{w}$ satisfy the first order KKT conditions for (1).*

Proof:

a. Let $\mathbf{z} = {}^1\!/\!_2(I+D)\mathbf{w}$, $\mathbf{v} = {}^1\!/\!_2(I-D)\mathbf{w}$. By construction, $\mathbf{z} \geq 0$, $\mathbf{v} \geq 0$, $\mathbf{z}^T\mathbf{v} = 0$.

b. By assumption we have

$$\mathbf{w} = D[N\mathbf{w} - {}^1\!/\!_2(I-D)]\mathbf{w} + DRA^T S\mathbf{b} - DN\mathbf{c}/\rho.$$

This equation can be rewritten

$$0 = DN(\mathbf{w} - \mathbf{c}/\rho) - {}^1\!/\!_2(I+D)\mathbf{w} + DRA^T S\mathbf{b},$$

or

$$\mathbf{z} = {}^1\!/\!_2(I+D)\mathbf{w} = DN(\mathbf{w} - \mathbf{c}/\rho) + DRA^T S\mathbf{b}.$$

Noting that $D\mathbf{z} = \mathbf{z}$ and $D^2 = I$, this means that the $\mathbf{z}$ here is the same as the $\mathbf{x}$ in equation (5).

c. Using $\rho\mathbf{v} = \rho(\mathbf{w} - \mathbf{z})$, the above means that $\mathbf{z}$ must satisfy the equation

$$\begin{pmatrix} Q & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \mathbf{z} \\ \boldsymbol{\nu} \end{pmatrix} = \begin{pmatrix} \rho\mathbf{v} - \mathbf{c} \\ \mathbf{b} \end{pmatrix}$$

d. We have thus satisfied all the KKT conditions:
(1) the gradients satisfy $Q\mathbf{z} + A^T\boldsymbol{\nu} - \mathbf{y} = 0$;
(2) the equality constraints are satisfied: $A\mathbf{z} = \mathbf{b}$;
(3) the inequality constraints are satisfied: $\mathbf{z} \geq 0$;
(4) the multipliers have the right sign: $\mathbf{y} \geq 0$;
(5) the complementarity conditions are satisfied: $\mathbf{y}^T\mathbf{z} = 0$;
where $\mathbf{y} = \rho\mathbf{v}$ are the multipliers for the inequality constraints and $\boldsymbol{\nu}$ are the multipliers for the equality constraints.

▮

Since the ADMM iteration has been converted into a variation of an eigenproblem, we can study the convergence in terms of the spectral properties of the operator $\mathbf{M}$ defined by (7). These properties are exposed by the decomposition in the following Lemma.

Lemma 11. *The matrix $\mathbf{M}$ defined by (7) for any flag matrices $\widehat{D}$ and D has a spectral decomposition $\mathbf{M} = \mathbf{P}\mathbf{J}\mathbf{P}^{-1}$, where $\mathbf{J}$ is a block diagonal matrix:*

$$\mathbf{J} = \begin{pmatrix} \mathbf{J}_1 & 0 & 0 & 0 \\ 0 & \mathbf{J}_2 & 0 & 0 \\ 0 & 0 & \mathbf{J}_3 & 0 \\ 0 & 0 & 0 & \mathbf{J}_4 \end{pmatrix} = \begin{pmatrix} \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} & 0 & 0 & 0 \\ 0 & I & 0 & 0 \\ 0 & 0 & \mathbf{J}_3 & 0 \\ 0 & 0 & 0 & \mathbf{J}_4 \end{pmatrix}, \tag{8}$$

where $\mathbf{J}_1$ is a single 2×2 Jordan block for eigenvalue 1 (possibly absent), I is an identity matrix (possibly empty), $\mathbf{J}_3$ is a diagonal matrix with diagonal entries all having absolute value 1, but not equal to 1, and $\mathbf{J}_4$ is a matrix with spectral radius strictly less than 1 (possibly empty). If $\widehat{D} = D$ in (7), then the $\mathbf{J}_3$ block is absent.

Proof: The upper left block of (7) satisfies Lemma 7 and hence contributes blocks of the form I, $\mathbf{J}_3$, $\mathbf{J}_4$. No eigenvalue with absolute value 1 can have a non-diagonal Jordan block, so the blocks corresponding to those eigenvalues must be diagonal. Embedding that upper left block into the entire matrix yields a matrix with the exact same set of eigenvalues. Except for eigenvalue 1, the algebraic and geometric multiplicities of the eigenvalues of $\mathbf{M}$ match those of its upper left block M.

If the upper left block of $\mathbf{M}$ (7) has no eigenvalue equal to 1, then $\mathbf{M}$ has a simple eigenvalue 1. In general for eigenvalue 1, the algebraic multiplicity goes up by one and the geometric multiplicity can either stay the same or increase by 1. In other words, $\mathbf{M}$ either satisfies the conditions of Lemma 9, or the algebraic and geometric multiplicities of eigenvalue 1 for $\mathbf{M}$ differ by 1, meaning we have a single 2×2 Jordan block.

If $\widehat{D} = D$, then the upper left block of (7) satisfies Lemma 8, hence the $\mathbf{J}_3$ block must be absent. ▮

Lemma 11 immediately yields the possible local behaviors or "regimes" that can arise from the ADMM iteration, in terms of the recurrence (7). There are four possible regimes that can arise, depending on the eigenvalues of the augmented matrix $\mathbf{M}$, summarized as follows:

(a) The spectral radius of M is strictly less than 1. If close enough to the optimal solution (if it exists), the result is linear convergence to that solution.

(b) M has an eigenvalue equal to 1 which results in a 2×2 Jordan block for $\mathbf{M}$. The process tends to a constant step, either diverging, or driving some component negative, resulting in a change in the operator M.

(c) M has an eigenvalue equal to 1, but $\mathbf{M}$ still has no non-diagonal Jordan block for eigenvalue 1; If close enough to the optimal solution (if it exists), the result is linear convergence to that solution.

(d) M has an eigenvalue of absolute value 1, but not equal to 1. This is possible only if $\widehat{D} \neq D$, i.e., if a sign of an element of $\widehat{\mathbf{w}}$ differs from $\mathbf{x} - \mathbf{v}$. The next iteration will be using a different operator with different flags.

The four regimes correspond to four possible configurations in the diagonalization (8). The resulting behavior depends on the corresponding spectral properties. The first three regimes can occur only when the flag matrix remains unchanged: $\widehat{D} = D$. In detail the regimes are as follows.

(a) If the spectral radius of M is strictly less than 1, then the blocks $\mathbf{J}_1, \mathbf{J}_3$ are absent from (8), and the block $\mathbf{J}_2 = I$ is 1×1. The recurrence (7) will converge linearly to some fixed point. If we are close enough to the solution to (1), the set of active inequality constraints at the current iteration should match those at the optimal solution, and hence this fixed point is the optimal solution to the original problem (1).

(b) If M has an eigenvalue equal to 1, but no other eigenvalue on the unit circle, then the block $\mathbf{J}_3$ is absent, but the non-diagonal Jordan block $\mathbf{J}_1$ might or might not be present. If present, we'd have a Jordan chain [13]: two non-zero vectors $\mathbf{q}, \mathbf{r}$ such that $(\mathbf{M}-I)\mathbf{q} = \mathbf{r}$, $(\mathbf{M}-I)\mathbf{r} = 0$. Any vector which includes a component of the form $\alpha\mathbf{q} + \beta\mathbf{r}$ would be transformed by $\mathbf{M}$ into $\mathbf{M}(\alpha\mathbf{q} + \beta\mathbf{r}) = \alpha\mathbf{q} + (\alpha + \beta)\mathbf{r}$, i.e., each iteration would add a constant vector $\alpha\mathbf{r}$, plus fading lower order terms from the other lesser eigenvalues [20, sec. 7.3]. This would result in constant steps: the difference between consecutive iterates, $\begin{pmatrix} \widehat{\mathbf{w}} \\ 1 \end{pmatrix} - \begin{pmatrix} \mathbf{w} \\ 1 \end{pmatrix}$, would converge to a constant vector, possibly in oscillatory fashion. The ADMM iteration will not converge unless and until a sign change in $\mathbf{w}$ forces a change in the flags D. If we satisfy the conditions for global convergence of ADMM, then such a sign change is guaranteed to occur.

(c) If M has an eigenvalue equal to 1, but the block $\mathbf{J}_1$ is absent, and $\mathbf{J}_3$ is also absent, then the recurrence (7) would still linearly converge to a fixed point at a rate determined by the next biggest eigenvalue in absolute value (largest eigenvalue of the block $\mathbf{J}_4$).

If there is a unique optimal solution to (1) with corresponding $\mathbf{w}^*$ and flags D^*, then $\begin{pmatrix} \mathbf{w}^* \\ 1 \end{pmatrix}$ must be a fixed point for (7), i.e., an eigenvector of the augmented matrix corresponding to eigenvalue 1. This is possible only if eigenvalue 1 has no 2×2 Jordan blocks (Lemma 9). Hence we conclude that, close enough to the solution to have the correct active inequality constraints, we should observe linear convergence at a rate determined by the largest eigenvalue of $M^* = D^*(N - {}^1\!/_2(I - D^*))$ that is strictly less than 1 in absolute value.

The above three cases apply when $\widehat{D} = D$. As the Theorem 12 below indicates, this will happen if we are close enough to the optimal solution, if it exists. In the general case, we can also have the following regime.

(d) If $M = \widehat{D}(N - {}^1\!/_2(I - D))$ has an eigenvalue with absolute value 1, then $\mathbf{J}_3$ is present, and we are likely to see some oscillation. This can happen only if the flags D are changing from one iteration to the next, so cannot happen if we are close enough to the optimal solution of (1), if one exists. Because ADMM satisfies a global convergence property (assuming a solution exists and other weak assumptions), at some stage a change in sign in $\mathbf{w}$ should change the flags and result in a different convergence regime, eventually ending up in regime (a) or (c).

We now state the main theorem regarding local linear convergence.

Theorem 12. *Suppose the problem (1) has a unique solution $\mathbf{x}^*, \mathbf{y}^*$ such that for every index i, either $\mathbf{x}^*_i = \mathbf{z}^*_i > 0$ or $\mathbf{y}^*_i > 0$ (but not both by the complementarity condition). Then there exists an open neighborhood around the solution in which ADMM converges linearly to the solution.*

Proof: The optimal solution to (1) is a fixed point for the ADMM iteration, and hence an eigenvector of $\mathbf{M}$ corresponding to eigenvalue 1, with an appropriate choice of flags $D = \widehat{D} = D^*$. By Lemma 9 the eigenvalue 1 cannot have a non-trivial Jordan block, and by Lemma 8 all other eigenvalues are less than 1 in absolute value. So there is a small neighborhood around the eigenvector where the power method will converge linearly to this eigenvector [20, sec. 7.3] according to the ratio $|\lambda_2|/1$, where λ_2 is the second highest eigenvalue in absolute value. If no element of the eigenvector is zero or negative, then there is a small neighborhood around this eigenvector where all the power method iterates will also be strictly positive. Hence, within this small neighborhood, we must be in regime (a) or (c) above. ∎

6 Acceleration

A proposed way to accelerate ADMM (Algorithm 1) is the following [4]

```
Algorithm 4: One Pass of Accelerated ADMM
Start with $\mathbf{z}, \mathbf{u}$.
    1. Solve $\begin{pmatrix} Q+\rho I & A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \boldsymbol{\nu} \end{pmatrix} = \begin{pmatrix} \rho(\mathbf{z}-\mathbf{u})-\mathbf{c} \\ \mathbf{b} \end{pmatrix}$ for $\mathbf{x}, \boldsymbol{\nu}$.
    2. Set $\widehat{\mathbf{x}} = \alpha\mathbf{x} + (1-\alpha)\mathbf{z}$.        $\longleftarrow$ (acceleration step)
    3. Set $\widehat{\mathbf{z}} = \max\{0, \widehat{\mathbf{x}} + \mathbf{u}\}$.
    4. Set $\widehat{\mathbf{u}} = \mathbf{u} + \widehat{\mathbf{x}} - \widehat{\mathbf{z}}$.
Result is $\widehat{\mathbf{z}}, \widehat{\mathbf{u}}$ for next iteration.
```

An analysis similar to sec. 3 yields the same complementarity conditions for $\widehat{\mathbf{z}}, \widehat{\mathbf{u}}$, and the following expressions for $\widehat{\mathbf{x}}$, based on (6):

$$\widehat{\mathbf{x}} = \alpha\mathbf{x} + (1-\alpha)\mathbf{z}. = \alpha N\mathbf{w} + \alpha\mathbf{h} + (1-\alpha)\mathbf{z}. \tag{9}$$

Using $z_i = \nicefrac{1}{2}(1+d_i)w_i$, we can follow the analysis similar to (3) to obtain

$$\begin{array}{rcl} \widehat{d}_i & = & \begin{cases} -1 & \text{if} \quad \widehat{x}_i - \nicefrac{1}{2}(1-d_i)w_i \le 0 \\ +1 & \text{if} \quad \widehat{x}_i - \nicefrac{1}{2}(1-d_i)w_i > 0 \end{cases} \\ \widehat{w}_i & = & |\widehat{x}_i - \nicefrac{1}{2}(1-d_i)w_i| = \widehat{d}_i[\widehat{x}_i - \nicefrac{1}{2}(1-d_i)w_i] \;\; , \\ & = & \widehat{d}_i[\alpha x_i + \nicefrac{1}{2}(1-\alpha)(1+d_i)w_i - \nicefrac{1}{2}(1-d_i)w_i] \\ & = & \widehat{d}_i[\alpha(x_i - (1+d_i)w_i) + d_i w_i] \end{array} \tag{10}$$

which yields the accelerated formula

$$\widehat{\mathbf{w}} = M(\alpha)\mathbf{w} + \alpha\widehat{D}\mathbf{h} = \widehat{D}[\alpha(N - \nicefrac{1}{2}(I+D)) + D]\mathbf{w} + \alpha\widehat{D}\mathbf{h}. \tag{11}$$

This reduces to step 2 of Algorithm 3 when $\alpha = 1$. We have the following lemma
Lemma 13. *For any $0 < \alpha < 2$, the spectrum of $M(\alpha) = \widehat{D}[\alpha(N - \nicefrac{1}{2}(I+D)) + D]$ lies in the unit disk on the complex plane. When $\widehat{D} = D$, the spectrum of $D[\alpha(N - \nicefrac{1}{2}(I+D)) + D]$ lies in the disk $\mathcal{D}(1-\nicefrac{\alpha}{2}, \nicefrac{\alpha}{2})$. For a linear program $Q = 0$ and $\widehat{D} = D$, the eigenvalues lie on the boundary of $\mathcal{D}(1-\nicefrac{\alpha}{2}, \nicefrac{\alpha}{2})$.*
Proof:

a. The eigenvalues of N are in $[0, 1]$.

b. $\widehat{D}[\alpha(N - \nicefrac{1}{2}(I+D)) + D] = \widehat{D}D[\alpha D(N - \nicefrac{1}{2}I) + I(1-\nicefrac{\alpha}{2})]$.

c. The eigenvalues of $\alpha(N - \nicefrac{1}{2}I))$ are in $[-\nicefrac{\alpha}{2}, \nicefrac{\alpha}{2}]$. So $\|\alpha(N - \nicefrac{1}{2}I))\| \le \nicefrac{\alpha}{2}$.

d. $\|\alpha D(N - \nicefrac{1}{2}I))\| \le \nicefrac{\alpha}{2}$. Hence the eigenvalues of $\alpha D(N - \nicefrac{1}{2}I)$ are in $\mathcal{D}(0, \nicefrac{\alpha}{2})$.

e. The eigenvalues of $\alpha D(N-{}^1\!/_2 I)+I(1-{}^\alpha\!/_2)$ are in $\mathcal{D}(1-{}^\alpha\!/_2, {}^\alpha\!/_2)$. If $\widehat{D}=D$, then $\widehat{D}D=I$, and $M(\alpha)=\alpha D(N-{}^1\!/_2 I)+I(1-{}^\alpha\!/_2)$.

f. $\|\alpha D(N-{}^1\!/_2 I)+I(1-{}^\alpha\!/_2)\| \le \|\alpha D(N-{}^1\!/_2 I)\| + \|I(1-{}^\alpha\!/_2)\| \le {}^\alpha\!/_2 + (1-{}^\alpha\!/_2) = 1$.

g. For a linear program, $N–{}^1\!/_2 I$ is half a unitary matrix, hence its eigenvalues lie on the boundary of $\mathcal{D}(0, {}^1\!/_2)$, hence the eigenvalues of $\alpha D(N-{}^1\!/_2 I)+I(1-{}^\alpha\!/_2)$ are on the boundary of the disk $\mathcal{D}(1-{}^\alpha\!/_2, {}^\alpha\!/_2)$.

h. If $D=\widehat{D}$ we are done, Otherwise, $\|M(\alpha)\| = \|\widehat{D}D[\alpha D(N-{}^1\!/_2 I)+I(1-{}^\alpha\!/_2)]\| = \|\alpha D(N-{}^1\!/_2 I)+I(1-{}^\alpha\!/_2)\| \le 1$. since $\widehat{D}D$ is unitary. Hence the eigenvalues of $M(\alpha)=\widehat{D}D[\alpha D(N-{}^1\!/_2 I)+I(1-{}^\alpha\!/_2)]$ are in $\mathcal{D}(0,1)$.

∎

This suggests that one should choose α to push the eigenvalues away from the boundary of the unit disk, but this turns out to be difficult if the eigenvalues are located on the boundary of the disk $\mathcal{D}(1-{}^\alpha\!/_2, {}^\alpha\!/_2)$, as we now elaborate for a linear program.

Adjusting the acceleration parameter $\alpha \neq 1$ will not accelerate the iteration during regime (a) or (c). In such a regime, $\widehat{D}=D$ and $M(\alpha)=\alpha[M-I(1-{}^1\!/_\alpha)]=\alpha[M-\sigma I]$ where $\sigma = 1-{}^1\!/_\alpha$ is a shift such that the eigenvalue 1 of M is mapped to the eigenvalue 1 of $M(\alpha)$. We can examine the ratio of an eigenvalue of $M-\sigma I$ to the eigenvalue resulting from shifting the eigenvalue 1. Let $\lambda=(1+c+is)/2$ be prospective eigenvalue of M on the boundary of $\mathcal{D}({}^1\!/_2, {}^1\!/_2)$, with $c^2+s^2=1$. We can calculate the ratio $|\lambda-\sigma|/|1-\sigma|$ for some real shift σ. A tedious algebraic manipulation yields the result that this ratio is minimized when $\sigma=0$, i.e., no shift. So during the last stage of the ADMM process, in regime (a) or (c), a shift will not yield a useful acceleration, and can actually slow down the convergence. We also remark that during the last linear stage, the spectrum of the matrix operator is also independent of ρ.

During regime (b) the process converges to a "constant step," that is, the difference between consecutive iterates $\widehat{\mathbf{w}}-\mathbf{w}$ converges to a constant vector. In such a regime, a shift will still yield a speedup since it amounts to over-relaxation on the constant step.

7 Example

We show in Fig. 1 the ADMM convergence behavior on an 80 variable linear program with 46 constraints derived from the analysis of metabolic networks. Here $\rho=1$, and no acceleration is used. The software used was the Matlab code for linear programs obtained from the web site associated with the paper [4] (which includes the acceleration represented by Alg. 4), modified to record a complete history of the iterates. The software was also modified to attempt to detect the regime at each iteration, with the option of applying a shortcut to the process if it decided it was in regime (a) or (c) and could make a good guess as to which inequality constraints were active. If the active inequality constraints could be identified, the variables that are supposed to be zero were completely removed, and the reduced system solved directly.

Using the notation from the proof of Theorem 3, the curves in Fig. 1 are, respectively from top to bottom, the error $\|\mathbf{w}^k-\mathbf{w}^*$ (A), the difference between two consecutive iterates $\|\mathbf{w}^k-\mathbf{w}^{k-1}\|$ (B), the primal residual $\|\mathbf{x}^k-\mathbf{z}^k\|$ (C), and the dual residual $\|\mathbf{z}^{k+1}-\mathbf{z}^k\|$ (D), where curve D is scaled by 1/10 just to separate it from the other curves in the figure. The quantities C, D are already computed within the algorithm for its stopping test. We find that these last two quantities jump

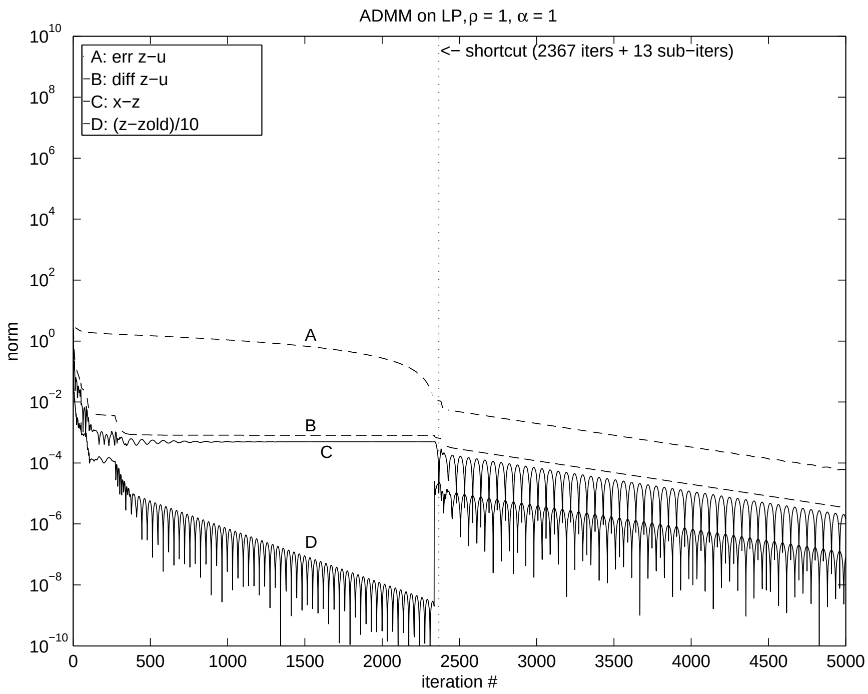

Figure 1: Unaccelerated ADMM applied to a 46×80 linear program, using $\rho = 1$. The curve D has been scaled by $1/10$ just to separate it from the rest.

around a lot, while the first two exhibit almost monotonic behavior, as predicted by the theory presented in this paper.

From passes 652 to 2336, we observe "constant step convergence" consistent with regime (b) in section 4. At each pass we add a constant vector to the iterate $\mathbf{w} = \mathbf{z} - \mathbf{u}$, until one or more of its entries is about to become negative. It then goes through a few transitional steps until by iteration 2341 the process has, in effect, correctly identified the active inequality constraints and enters regime (c), i.e., linear convergence to the fixed point. During this stage of linear convergence, the augmented matrix $\mathbf{M}$ has eigenvalue 1 with a complete set of 11 eigenvectors. This implies that the LP has multiple optimal solutions occupying a simplex of dimension at most 10. The spectrum of the augmented matrix $\mathbf{M}$ during the last stage of the ADMM iteration is shown in Fig. 2. The next highest eigenvalue (in absolute value) is $\lambda_{12} = \mathtt{0.9965}\pm\mathtt{0.0594i}$ with absolute value 0.9982. Since the second highest eigenvalue is complex, the iterates should follow a shrinking spiral with period $2\pi/(\angle\lambda_{12}) \approx 106$ (or some divisor thereof) toward the solution with a linear convergence rate of 0.9982. This spiraling behavior is consistent with that observed in [7]. Because this particular LP has multiple solutions, we need to apply a very conservative check to determine which regime is currently operative, and which constraints can be considered active. Hence we used a rather tight tolerance of 10^{-8} as a zero tolerance, but even in this case it was able to correctly identify the correct constraints at step 2366, and take a shortcut directly to the exact solution.

We tried an accelerated ADMM using $\alpha = 1.8$ on the same problem, using the same software as in the previous case. We find that regime (b) "constant step convergence" starts at step 652 as before, but ends sooner at iteration number 1887, but the final regime (c) "linear convergence" is slower, so that more than 5000 iterations are needed to achieve the same accuracy as in the

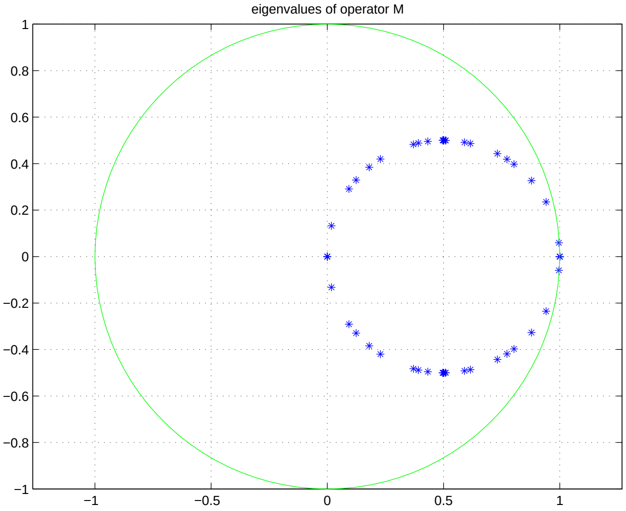

Figure 2: Spectrum (stars) of the ADMM iteration operator $\mathbf{M}^*$ on the complex plane during the last regime of the ADMM process on the linear programming example. The unit circle (outer circle) is shown for reference.

unaccelerated method. If acceleration were limited to only iterations in regime (b), the final regime would converge at the same rate as before, and the same short cut could also be taken. The result is that the exact solution should be obtainable with only 1918 iterations. The accelerated convergence is illustrated in Fig. 3.

8 Conclusions

In this paper, we have analyzed the ADMM method for a quadratic or linear program in standard form by modelling it as a matrix recurrence. The spectrum of the matrix recurrence has been used to analyze the convergence of the method. It is shown that the method normally passes through several regimes as it searches for the correct set of active constraints. When the method finally settles on the correct set of active constraints, convergence is linear, depending on the absolute value of second largest eigenvalue of the matrix recurrence.

We have also illustrated that ADMM goes through several stages or regimes on its way toward the final solution. We conclude that any attempt to accelerate ADMM would be much more successful if it took these regimes into account. With a reliable and inexpensive way to detect which regime was currently operative, it would be possible to apply the appropriate acceleration technique. During regime (b) “constant step”, over-relaxation would be very appropriate. During (a) or (c) “linear convergence”, the algorithm has essentially settled on a set of active constraints, represented by the diagonal matrix of flags D. In this case, it is possible to apply a shortcut by removing the variables which are supposed to be zero corresponding to the active constraints and solving the reduced problem directly. In our example, the shortcut was successful, resulting in a process in which the “linear convergence” regime was cut short, and most of the iterations were in a sub-linear “constant step convergence” regime.

In future work, these results might be useful in designing more effective and robust methods to accelerate the method, especially if the acceleration technique is tailored to match whatever is the

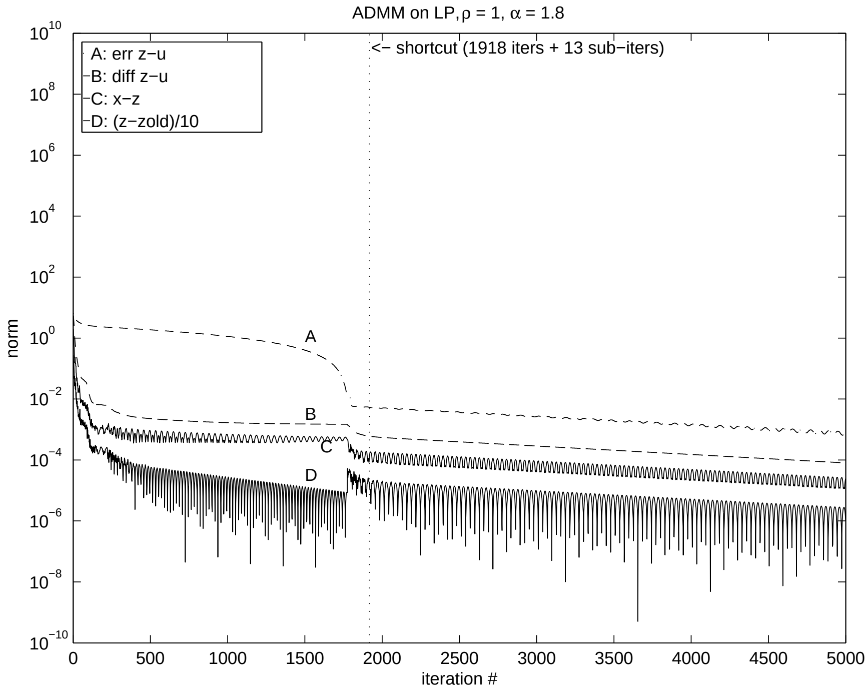

Figure 3: Accelerated ADMM applied to a 46 × 80 linear program, using $\rho = 1$. The curve D has been scaled by 1/10 just to separate it from the rest.

current regime of the process. In particular, we have investigated the effect of varying α but not ρ, but not how these would affect each other if both were varied. In addition, we have analyzed only one way to split the original model problem (1) into parts suitable for an alternating direction method. Even though the present analysis is only a first step, it does give a hint on the limits on the performance one can expect in the general case.

Acknowledgments

The author would like to thank Arindam Banerjee, Huahua Wang for introducing this problem, Shuzhong Zhang, Shiqian Ma for helpful discussions, and NSF for their financial support via grant 0916750.

References

[1] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM J. Imaging Sciences*, 2(1):183–202, 2009.

[2] D. Bertsekas and J. Tsitsiklis. *Parallel and Distributed Computation.* Prentice Hall, 1989.

[3] D. Bertsekas and J. Tsitsiklis. *Parallel and Distributed Computation: Numerical Methods.* Athena Scientific, 1997.

[4] 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(1):1–122, 2011. `http: //www.stanford.edu /~boyd /papers /admm/`.

[5] C. H. Chen, B. S. He, and X. M. Yuan. Matrix completion via an alternating direction method. *IMA J. Numer. Anal.,*, 32:227–245, 2012.

[6] E. Dall'Anese, J. A. Bazerque, and G. B. Giannakis. Group sparse LASSO for cognitive network sensing robust to model uncertainties and outliers. *Physical Communication*, 5(5):161–172, 2012.

[7] J. Eckstein and D. P. Bertsekas. An alternating direction method for linear programming. MIT Lab. for Info. and Dec. Sys. report LIDS-P-1967, April 1990.

[8] J. Eckstein and D. P. Bertsekas. On the Douglas-Rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical Programming*, 55:293–318, 1992.

[9] T. Erseghe, D. Zennaro, E. Dall'Anese, and L. Vangelista. Fast consensus by the alternating direction multipliers method. *IEEE Trans. Signal Proc.*, 59(11):5523–5537, 2011.

[10] E. Esser. Applications of lagrangian-based alternating direction methods and connections to split Bregman. UCLA CAM Report 09-31, University of California, Los Angeles, 2009.

[11] M. Fukushima. Application of the alternating direction method of multipliers to separable convex programming problems. *Comput. Optim. Appl.,*, 2:93–111, 1992.

[12] D. Gabay and B. Mercier. A dual algorithm for the solution of non- linear variational problems via finite-element approximations. *Comp. Math. Appl.*, 2:17–40, 1976.

[13] F. R. Gantmacher. *The Theory of Matrices*. Chelsea Publishing Company, New York, 1959.

[14] R. Glowinski and A. Marrocco. Sur l'approximation, par éléments finis d'ordre un, et la résolution par pénalisation-dualité, d'une classe de problémes de Dirichlet non linéaires. *Revue Française d'Automatique, Informatique, et Recherche Opérationelle*, 9(2):41–76, 1975.

[15] R. Glowinski and P. L. Tallec. *Augmented Lagrangian and Operator-Splitting Methods in Nonlinear Mechanics*, volume 9. SIAM Studies in Applied and Numerical Mathematics, 1989.

[16] B. He and X. Yuan. On the $O(1/t)$ convergence rate of alternating direction method. http://www.optimization-online.org/DB_HTML/2011/09/3157.html, 2011.

[17] B. S. He, L. Z. Liao, D. R. Han, and H. Yang. A new inexact alternating directions method for monontone variational inequalities. *Math. Program. Ser. A,*, 92:103–118, 2002.

[18] B. S. He, M. H. Xu, and X. M. Yuan. Solving large-scale least squares semidefinite programming by alternating direction methods. *SIAM J. Matrix Anal. Appl.,*, 32:136–152, 2011.

[19] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, Cambridge, 1985.

[20] A. S. Householder. *The Theory of Matrices in Numerical Analysis.* Dover Publishing, New York, 1964. Originally published by Ginn Blaisdell.

[21] S. Kontogiorgis and R. R. Meyer. A variable-penalty alternating directions method for convex optimization. *Math. Program.,*, 83:29–53, 1998.

[22] J. F. C. Mota, J. M. F. Xavier, P. M. Q. Aguiar, and M. Püschel. D-ADMM: A communication-efficient distributed algorithm for separable optimization. arXiv:1202.2805v1 [math.OC], 2012.

[23] Y. E. Nesterov. Gradient methods for minimizing composite objective function. CORE report, 2007. http://www.ecore.be/DPs/dp 1191313936.pdf.

[24] M. K. Ng, P. Weiss, and X. M. Yuan. Solving constrained total-variation image restoration problems via alternating direction methods. *SIAM J. Sci. Comput.*, 32:2710–2736, 2010.

[25] J. M. Ortega. *Numerical Analysis: A Second Course.* Academic Press, New York, 1972. (republished by SIAM, 1990).

[26] M. Tao and X. M. Yuan. Recovering low-rank and sparse components of matrices from incomplete and noisy observations. *SIAM J. Optim.,*, 21:57–81, 2011.

[27] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight. Sparsity and smoothness via the fused LASSO. *J. Royal Statist. Soc.,*, 67:91–108, 2005.

[28] H. Wang and A. Banerjee. Online alternating direction method. In *Proc. 29th Intl. Conf. Machine Learning*, 2012. to appear.

[29] J. Yang and Y. Zhang. Alternating direction algorithms for 1 -problems in compressive sensing. *SIAM J. Sci. Comput.,*, 33:250–278, 2011.

[30] C.-H. Ye and X.-M. Yuan. A descent method for structured monotone variational inequalities. *Optim. Methods Softw.,*, 22:329–338, 2007.