Solving symmetric indefinite systems in an interior-point method for second order cone programming

Kim-Chuan Toh, Zhi Cai, and R. Freund

Abstract—Many optimization problems can be formulated as second order cone programing (SOCP) problems. Theoretical results show that applying interior-point method (IPM) to SOCP has global polynomial convergence. However, various stability issues arise in the implementation of IPM. The standard normal equation based implementation of IPM encounters stability problems in the computation of search direction. In this paper, an augmented system approach is proposed to overcome the instability problems. Numerical experiments show that the new approach can improve the stability.

I. INTRODUCTION

A second order cone programing (SOCP) problem is a linear optimization problem over a second order convex cone. In [4], an extended list of application problems are shown to be SOCP problems. In particular, linear programming (LP) problems, and convex quadratically constrained quadratic programming (QCQP) are both subclasses of SOCP. SOCP itself is a subclass of semidefinite programing (SDP). In theory, SOCP problems can be solved as SDP problems. However, it is far more efficient to solve an SOCP problem directly. In the past few years, global polynomial convergence results concerning the application of interior-point methods (IPM) to SOCP have been established [5]. But there is relative little published research work on the implementation of IPM for solving SOCP. It has been reported [9] that the major challenges in the implementation of IPM for SOCP are the stable and efficient computation of search directions in each iteration of the IPM. In this paper, we address the issue of stable computation of search directions. However, we are aware that the method we proposed must also be computationally efficient.

Given a vector $x$, we will write the vector as $x = (x^{(0)}; x)$ with $x^{(0)}$ being the first component and $x$ being the vector consisting of the remaining components.

We use a similar notation for $z_i$. For $x$, we define

$$
\bar{\text{Arr}}(x) = \begin{bmatrix}
x_i^{0} & x_i^{1} & \ldots & x_i^{n_i} \\
x_i^{1} & x_i^{0} & \ldots & \ldots \\
\vdots & \vdots & \ddots & \vdots \\
x_i^{n_i} & x_i^{0} & \ldots & x_i^{1}
\end{bmatrix},
$$

and $\gamma(x_i) = \sqrt{(x_i^{0})^2 - ||x_i||^2}$, where $||x_i||$ is the Euclidean norm.

We consider the standard primal and dual SOCP problems:

Primal:

$$
\begin{align*}
\text{min} & \quad c^T x_1 + c_2^T x_2 + \cdots + c_N^T x_N \\
\text{s.t.} & \quad A_1 x_1 + A_2 x_2 + \cdots + A_N x_N = b \\
& \quad x_i \geq \kappa 0,
\end{align*}
$$

Dual:

$$
\begin{align*}
\text{max} & \quad b^T y \\
\text{s.t.} & \quad A_1^T y + z_i = c_i, \quad i = 1, \ldots, N \\
& \quad z_i \geq \kappa 0,
\end{align*}
$$

where $A_i \in \mathbb{R}^{m_i \times n_i}, c_i, x_i, z_i \in \mathbb{R}^{n_i}, i = 1, \ldots, N$, and $y \in \mathbb{R}^m$. The second order cone constraint $x_i \geq \kappa 0$ means that $x_i^{0} \geq ||x_i||$.

For convenience, we define

$$
A = [A_1 \ A_2 \ \cdots \ A_N], \quad c = [c_1; c_2; \cdots; c_N], \\
x = [x_1; x_2; \cdots; x_N], \quad z = [z_1; z_2; \cdots; z_N].
$$

The KKT conditions of the above primal-dual systems are:

$$
\begin{align*}
Ax &= b & \text{(Primal Feasibility)} \\
A^T y + z &= c & \text{(Dual Feasibility)} \\
x_i \circ z_i &= 0, \quad i = 1, \ldots, N, & \text{(Complementary)}
\end{align*}
$$

where $x_i \circ z_i = (x_i^{T} z_i; x_i^{0} \bar{z}_i; z_i^{0} x_i)$.

In applying an IPM to solve an SOCP, one relaxes the complementary condition in (3). That is, the original complementary condition is replaced by the following condition:

$$
x_i \circ z_i = \mu e_i, \quad (4)
$$

where $e_i$ is the first unit vector in $\mathbb{R}^{n_i}$ and $\mu$ is a positive parameter that is to be driven to 0 explicitly. As
\( \mu \) varies, the solutions to the relaxed KKT conditions form a path (known as the central path) in the the interior of the primal-dual feasible region, and as \( \mu \) gradually reduces to 0, the path converges to an optimal solution of the primal and dual SOCP problems.

II. THE AUGMENTED SYSTEM AND NORMAL EQUATION

In this section, we take a closer look at the system of nonlinear equations solved in a typical IPM iteration. For a given \( \mu \), the relaxed KKT condition is:

\[
\begin{align*}
Ax & = b \\
A^T y + z & = c \\
\text{Arr}(x) \text{Arr}(z)e^0 & = \mu e^0,
\end{align*}
\]

where \( e^0 = [e_1; e_2; \cdots; e_N] \). The matrix \( \text{Arr}(x) = \text{diag}(\text{Arr}(x_1), \cdots, \text{Arr}(x_N)) \) is an block diagonal matrix with \( \text{Arr}(x_1), \cdots, \text{Arr}(x_N) \) as the diagonal blocks. The matrix \( \text{Arr}(z) \) is defined similarly.

In order to construct a symmetric Schur complement matrix, a block diagonal scaling matrix (NT scaling matrix) is applied to the relaxed complementarity equation in (5) to produce the following equation:

\[
\text{Arr}(Fx) \text{Arr}(F^{-1}z)e^0 = \mu e^0, \quad (6)
\]

where \( F = \text{diag}(F_1, \cdots, F_N) \) is chosen such that \( Fx = F^{-1}z \). For details about the conditions that \( F \) must satisfy, we refer the reader to [5].

Let

\[
f_i = \begin{bmatrix} f_i^0 \\ f_i \end{bmatrix} = \frac{1}{\sqrt{2[\gamma(z_i)\gamma(x_i) + x_i^T z_i]}} \begin{bmatrix} \frac{1}{\omega_i} x_i^0 + \omega x_i^0 \\ \frac{1}{\omega_i} x_i - \omega x_i \end{bmatrix},
\]

where \( \omega_i = \sqrt{\gamma(z_i)/\gamma(x_i)} \). (Note that \( \gamma(f_i) = 1 \).) The precise form of \( F_i \) is given by

\[
F_i = \omega_i \begin{bmatrix} f_i^0 & f_i^T \\ f_i & I + f_i f_i^T \end{bmatrix}. \quad (7)
\]

The Newton equation associated with the relaxed KKT conditions (5) is given by

\[
\begin{bmatrix}
A & 0 & 0 & A^T & I & 0 & \Delta x \\
0 & A^T & I & 0 & 0 & \Delta y \\
\text{Arr}(F^{-1}z)F & 0 & \text{Arr}(Fx)F^{-1} & 0 & 0 & \Delta z
\end{bmatrix}
\begin{bmatrix}
\Delta x \\
\Delta y \\
\Delta z
\end{bmatrix}
= \begin{bmatrix}
r_p \\
r_d \\
r_c
\end{bmatrix},
\]

where \( r_p = b - Ax \), \( r_d = c - z - A^Ty \), and \( r_c = \mu e^0 - \text{Arr}(Fx)\text{Arr}(F^{-1}z)e^0 \).

The solution \( (\Delta x, \Delta y, \Delta z) \) of the Newton equation (8) is referred to as the search direction. Solving (8) for the search direction is the most computationally expensive step in each iteration of an IPM. Observe that by eliminating \( \Delta z \), the Newton equation (8) reduces to the so-called augmented system:

\[
\begin{bmatrix}
-F^2 & A^T \\
A & 0
\end{bmatrix}
\begin{bmatrix}
\Delta x \\
\Delta y
\end{bmatrix}
= \begin{bmatrix}
r \\
r_p
\end{bmatrix},
\]

where \( r = r_d - (\text{Arr}(Fx)^{-1}F^{-1})^{-1}r_c \).

The augmented system can further be reduced by eliminating \( \Delta x \) to produce the normal equation:

\[
\chi F^2 \Delta y = r_y := r_p + AF^{-2}r. \quad (10)
\]

The coefficient matrix \( \chi := AF^{-2}A^T \) is known as the Schur complement matrix, and it is a symmetric positive definite matrix.

Currently, most implementations of IPM are based on solving the normal equation (10). The advantage of using the normal equation is that it is a smaller system compared to the augmented system (9) or the Newton equation (8). Furthermore, the Schur complement matrix has the desirable property of being symmetric positive definite. On the other hand, the coefficient matrix in (9) is symmetric but indefinite while that of (8) is nonsymmetric.

However, as we shall see later, the Schur complement matrix can be severely ill-conditioned when \( \mu \) is close to 0, and this imposes a limit as to how accurately one can solve an SOCP. To analyze the conditioning of the Schur complement matrix, we need to know the eigenvalue decomposition of \( F^2 \) and we shall discuss that in the next section.

III. EIGENVALUE DECOMPOSITION OF \( F^2 \) AND CONDITIONING OF \( M \)

Recall that \( F = \text{diag}(F_1, \cdots, F_N) \). Thus to find the eigenvalue decomposition of \( F^2 \), we need to find the eigenvalue decomposition of \( F_i^2 \), where \( F_i \) is the matrix in (7).

By noting that \( F_i^2 \) can be written in the form

\[
F_i^2 = \omega_i^2 I + 2(f_i f_i^T - e_i e_i^T),
\]

the eigenvalue decomposition of \( F_i^2 \) can be found readily. Without going through the algebraic details, the eigenvalue decomposition of \( F_i^2 \) is

\[
F_i^2 = Q_i \Lambda_i Q_i^T, \quad (11)
\]

where

\[
\Lambda_i = \begin{bmatrix}
\frac{1}{\sqrt{\gamma_1}} & -\frac{1}{\sqrt{\gamma_1}} & 0 & \cdots & 0 \\
\frac{1}{\sqrt{2\gamma_1}} & \frac{1}{\sqrt{2\gamma_1}} & q_1 & \cdots & q_n
\end{bmatrix}, \quad (12)
\]

Notice that the first eigenvalue is the smallest and the second is the largest. The set \( \{q_1, \cdots, q_n\} \) is an orthonormal basis of the subspace \( \{v : v^T f_i = 0\} \). To construct such an orthonormal basis, one may first construct the Householder matrix \( H_i \) associated with the vector \( f_i \), then the last \( n_i - 2 \) columns of \( H_i \) is such an orthonormal basis.

The normal equation can be solved by Cholesky factorization efficiently. Although Cholesky factorization is stable for any symmetric positive definite matrix, the condition number of the matrix still affects the accuracy of the computed solution of the normal equation. It is a common phenomenon that for SOCP, the accuracy of the computed search direction deteriorates as \( \mu \) decreases due to an increasingly
ill-conditioned Schur complement matrix. As a result this loss of accuracy in the computed solution, the primal indefeasibility \( \| r_p \| \) typically increases when the IPM iterates approach an optimal solution.

To analyze the conditioning of the Schur complement matrix, we will make the following assumption on the SOCP problem:

Assumption 1. Strict complementarity holds at the optimal solution.

Referring to (3), strict complementarity [1] means for each \( i \)th pair of optimal primal and dual solutions \( x_i^* \) and \( z_i^* \), we have (a) either \( \gamma(x_i^*) = 0 \) or \( z_i^* = 0 \); and (b) either \( \gamma(z_i^*) = 0 \) or \( x_i^* = 0 \).

Under the assumption that strict complementarity holds at the optimal solution, we have the following three types of eigenvalue structure for \( F^2 \) when \( x_i, z_i = \mu e_i \) and \( \mu \) is small. Note that \( x_i^2 = \mu^2 \).

Type 1: \( \gamma(x_i^* ) = 0, \gamma(z_i^* ) = 0 \). This means for the current iteration, \( \gamma(x_i), \gamma(z_i) = \Theta(\sqrt{\mu}) \), and \( \omega_i = \Theta(1) \). This implies that \( F_i^{1,1} \parallel f_i \parallel = \Theta(1/\sqrt{\mu}) \). Thus The largest eigenvalue of \( F_i^2 \) is \( \Theta(1/\mu^2) \) and the smallest eigenvalue of \( F_i^2 \) is \( \Theta(\mu) \). The rest of the eigenvalues are \( \Theta(1) \).

Type 2: \( \gamma(x_i^* ) = 0, \gamma(z_i^* ) = 0 \). In this case, \( \gamma(x_i) = \Theta(1), \gamma(z_i) = \Theta(\mu) \), and \( \omega_i = \Theta(1) \). Also, \( \tilde{f}_i \parallel f_i \parallel = \Theta(1) \), implying that all the eigenvalues of \( F_i^2 \) are \( \Theta(1) \).

Type 3: \( \gamma(x_i^* ) = 0, \gamma(z_i^* ) = 0 \). In this case, \( \gamma(x_i) = \Theta(\mu), \gamma(z_i) = \Theta(1) \), and \( \omega_i = \Theta(1/\sqrt{\mu}) \). Also, \( \tilde{f}_i \parallel f_i \parallel = \Theta(1) \), implying that all the eigenvalues of \( F_i^2 \) are \( \Theta(1/\mu^2) \).

In general, eigenvalue structure of Type 1 occurs most frequently in practice. If the \( i \)th optimal dual solution is at the origin of the \( i \)th cone, then Type 2 occurs. Similarly, if the \( i \)th optimal primal solution is at the origin of the \( i \)th cone, then Type 3 occurs.

Assuming that the matrix \( A \) is well-conditioned, then the worsening of the conditioning of the Schur complement matrix is caused by an increasingly ill-conditioned \( F^2 \). Based on the above eigenvalue analysis, we see that if the optimal solution of the SOCP problem is purely of Type 2 or purely of Type 3, then the conditioning of \( M \) will be not severely affected by a small \( \mu \). However, if there are cones that lead to optimal solutions of Type 1 or a mixture of Type 2 and 3, then the condition number of the \( M \) is likely to grow like \( O(1/\mu^2) \).

In practice, it is reasonable to expect that an SOCP has at least two types of cones mixed. Hence, according to the above analysis, as \( \mu \) decreases, then the condition number of \( M \) will grow like \( O(1/\mu^2) \).

IV. Reduced augmented system approach

In this section, we present a new approach to compute the search direction via a better-conditioned linear system of equations. Hence, the accuracy in computed search direction can be expected to be better than that computed from the normal equation when \( \mu \) is small.

In this approach, we start with the augmented system in (9). By using the eigenvalue decomposition of \( F^2 \) with \( F^2 = QDQ^T \), where \( Q = \text{diag}(Q_1, \cdots, Q_N) \) and \( D = \text{diag}(\Lambda_1, \cdots, \Lambda_N) \), we can rewrite the augmented system (9) as follows.

\[
\begin{pmatrix}
-D & \tilde{A}^T \\
\tilde{A} & 0
\end{pmatrix}
\begin{pmatrix}
\Delta \bar{x} \\
\Delta y
\end{pmatrix}
= \begin{pmatrix}
\bar{r} \\
r_p
\end{pmatrix},
\]

where \( \bar{A} = AQ, \bar{x} = Q^T x, \) and \( \bar{r} = Q^T r \).

Based on (14), we will apply splitting technique in [7] to obtain a smaller indefinite system which is typically better conditioned than the original indefinite system.

Let the diagonal matrix \( D \) be partitioned into two parts as \( D = \text{diag}(D_1, D_2) \) with \( \text{diag}(D_1) \) consists of the small eigenvalues of \( F^2 \) and \( \text{diag}(D_2) \) consists of the remaining eigenvalues. We also partition the matrices \( Q \) as \( Q = [Q^{(1)}, Q^{(2)}] \). Then \( A \) is partitioned as \( A = [A_1, A_2] = [AQ^{(1)}, AQ^{(2)}] \). With such partitions, it is shown in [7] that the system (14) is equivalent to:

\[
\begin{pmatrix}
-D_1 & S_1^{-1/2} \tilde{A}_1^T \\
\tilde{A}_1 S_1^{-1/2} & \tilde{A} \text{diag}(S_1^{-1}, D_2^{-1}) \tilde{A}^T
\end{pmatrix}
\begin{pmatrix}
S_1^{-1/2} \Delta \bar{x}_1 \\
\Delta y
\end{pmatrix}
= \begin{pmatrix}
S_1^{-1/2} \bar{r} \\
r_p
\end{pmatrix},
\]

where \( E_1 \) is a given diagonal matrix that is usually chosen to be \( J \) and \( S_1 = D_1 + I_1 \). Here \( \Delta \bar{x}_1 = (Q^{(1)})^T \Delta x \).

We call the system in (15) the reduced augmented system. Note that once \( \Delta y \) is computed, \( \Delta \bar{x}_2 = (Q^{(2)})^T \Delta x \) can be computed from the equation \( \Delta \bar{x}_2 = D_2^{-1} (Q^{(2)})^T (A \Delta y) - r \). After that, \( \Delta \bar{x} \) can be recovered from the equation \( \Delta \bar{x} = Q^{(1)} \Delta \bar{x}_1 + Q^{(2)} \Delta \bar{x}_2 \).

Observe that the \( (2,2) \) block \( \text{LDL}^T \) factorization of the reduced augmented system has the same form as the Schur complement matrix \( M = A \text{diag}(D_1^{-1}, D_2^{-1}) A^T \). But for the \( (2,2) \) block, \( \text{diag}(S_1^{-1}, D_2^{-1}) = O(1) \), whereas for \( M_1, \text{diag}(D_1^{-1}, D_2^{-1}) = O(1/\mu^2) \). Because of this difference, the reduced augmented system is expected to be better-conditioned than the normal equation. Rigorous analysis on the conditioning of the reduced augmented system can be found in [7].

V. Numerical experiments

The reduced augmented system (15) is computationally more expensive to solve than the normal equations because it is larger in size. But if we use the approach suggested by K.D Anderson in [2] to solve the reduced augmented system via the sparse Cholesky factorization of the \( (2,2) \) block, in theory the cost should not be much more expensive than that of solving the normal equation. But we have yet to implement this approach for solving (15).

The coefficient matrix of the reduced augmented system (15) is a quasi-definite matrix [10]. Thus we can also solve it directly via the \( \text{LDL}^T \) factorization proposed in [10] for a quasi-definite matrix. However, efficient computation of such a factorization depends heavily on careful handling of sparsity of the matrix.

As our goal in this paper is merely to demonstrate that the reduced augmented system can produce more accurate computed search direction, in our numerical experiments, we use the standard sparse \( LU \) factorization to solve (15). The experiments are based on the IPM Matlab software SDPT3 [TTT99], [TTT01].

We compared the normal equation and reduced augmented system approaches on 3 SOCP problems. The first is a random SOCP problem, the second and third are SOCP problems arising from antenna array design, and they come from the library of mixed semidefinite-quadratic-linear programs collected by G. Pataki and S. Schmieta in the Dimacs Implementation Challenge 7 [PSD7]. In the tables presented below, the primal
and dual infeasibilities, \(|r_p|\) and \(|r_d|\), are denoted as p-infeas and d-infeas, respectively. The duality gap \(x^Tz\) is denoted as gap.

The numerical results show that the implementation of the normal equation encounters stability problems that are manifested through deteriorating primal infeasibilities towards the end of the IPM iterations. On the other hand, the reduced augmented system approach demonstrated better stability in the primal infeasibilities do not deteriorate significantly towards the end of the IPM iterations.

**Random SOCP problem.** Problem size: \(m = 5, N = 5, n_i = 3, i = 1 : N\).

**a) Implementation Based on the Normal Equation.**

<table>
<thead>
<tr>
<th>it</th>
<th>pstep</th>
<th>dstep</th>
<th>p-infeas</th>
<th>d-infeas</th>
<th>gap</th>
<th>obj</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
<td>9.7e-001</td>
<td>1.7e+000</td>
<td>2.2e+001</td>
<td>3.935775e+000</td>
</tr>
<tr>
<td>1</td>
<td>1.000</td>
<td>1.000</td>
<td>5.0e-016</td>
<td>0.0e+000</td>
<td>2.7e+000</td>
<td>1.932067e+000</td>
</tr>
<tr>
<td>2</td>
<td>0.945</td>
<td>0.902</td>
<td>2.8e-016</td>
<td>7.2e-017</td>
<td>2.5e-001</td>
<td>1.600947e+000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.985</td>
<td>1.000</td>
<td>2.9e-013</td>
<td>7.3e-017</td>
<td>1.6e-007</td>
<td>1.575656e+000</td>
</tr>
<tr>
<td>8</td>
<td>0.989</td>
<td>0.994</td>
<td>1.9e-013</td>
<td>7.2e-017</td>
<td>2.3e-009</td>
<td>1.575656e+000</td>
</tr>
</tbody>
</table>

Stop: max(relative gap, infeasibilities) < 1.e-008

number of iterations = 8

gap = 2.28e-009
relative gap = 1.44e-009
infeasibilities = 1.93e-013

**b) Implementation Based on Augmented Equation**

<table>
<thead>
<tr>
<th>it</th>
<th>pstep</th>
<th>dstep</th>
<th>p-infeas</th>
<th>d-infeas</th>
<th>gap</th>
<th>obj</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
<td>9.7e-001</td>
<td>1.7e+000</td>
<td>2.2e+001</td>
<td>3.935775e+000</td>
</tr>
<tr>
<td>1</td>
<td>1.000</td>
<td>1.000</td>
<td>5.0e-016</td>
<td>0.0e+000</td>
<td>2.7e+000</td>
<td>1.932067e+000</td>
</tr>
<tr>
<td>2</td>
<td>0.945</td>
<td>0.902</td>
<td>2.8e-016</td>
<td>7.2e-017</td>
<td>2.5e-001</td>
<td>1.600947e+000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.985</td>
<td>1.000</td>
<td>2.9e-013</td>
<td>7.3e-017</td>
<td>1.6e-007</td>
<td>1.575656e+000</td>
</tr>
<tr>
<td>8</td>
<td>0.989</td>
<td>0.994</td>
<td>1.9e-013</td>
<td>7.2e-017</td>
<td>2.3e-009</td>
<td>1.575656e+000</td>
</tr>
</tbody>
</table>

Stop: max(relative gap, infeasibilities) < 1.e-008

number of iterations = 8

gap = 2.28e-009
relative gap = 1.44e-009
infeasibilities = 1.93e-013

**Dimacs Challenge problem: nb-L1.** Problem size: \(m = 915, N = 793, n_i = 3, i = 1 : N\); Linear block-797.

**a) Implementation Based on Normal Equation**

<table>
<thead>
<tr>
<th>it</th>
<th>pstep</th>
<th>dstep</th>
<th>p-infeas</th>
<th>d-infeas</th>
<th>gap</th>
<th>obj</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
<td>9.7e-001</td>
<td>1.7e+000</td>
<td>2.2e+001</td>
<td>3.935775e+000</td>
</tr>
<tr>
<td>1</td>
<td>1.000</td>
<td>1.000</td>
<td>5.0e-016</td>
<td>0.0e+000</td>
<td>2.7e+000</td>
<td>1.932067e+000</td>
</tr>
<tr>
<td>2</td>
<td>0.945</td>
<td>0.902</td>
<td>2.8e-016</td>
<td>7.2e-017</td>
<td>2.5e-001</td>
<td>1.600947e+000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.985</td>
<td>1.000</td>
<td>2.9e-013</td>
<td>7.3e-017</td>
<td>1.6e-007</td>
<td>1.575656e+000</td>
</tr>
<tr>
<td>8</td>
<td>0.989</td>
<td>0.994</td>
<td>1.9e-013</td>
<td>7.2e-017</td>
<td>2.3e-009</td>
<td>1.575656e+000</td>
</tr>
</tbody>
</table>

Stop: max(relative gap, infeasibilities) < 1.e-008

number of iterations = 8

gap = 2.28e-009
relative gap = 1.44e-009
infeasibilities = 1.35e-016

**b) Implementation Based on Augmented Equation**

<table>
<thead>
<tr>
<th>it</th>
<th>pstep</th>
<th>dstep</th>
<th>p-infeas</th>
<th>d-infeas</th>
<th>gap</th>
<th>obj</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.000</td>
<td>0.000</td>
<td>9.7e-001</td>
<td>1.7e+000</td>
<td>2.2e+001</td>
<td>3.935775e+000</td>
</tr>
<tr>
<td>1</td>
<td>1.000</td>
<td>1.000</td>
<td>5.0e-016</td>
<td>0.0e+000</td>
<td>2.7e+000</td>
<td>1.932067e+000</td>
</tr>
<tr>
<td>2</td>
<td>0.945</td>
<td>0.902</td>
<td>2.8e-016</td>
<td>7.2e-017</td>
<td>2.5e-001</td>
<td>1.600947e+000</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.985</td>
<td>1.000</td>
<td>2.9e-013</td>
<td>7.3e-017</td>
<td>1.6e-007</td>
<td>1.575656e+000</td>
</tr>
<tr>
<td>8</td>
<td>0.989</td>
<td>0.994</td>
<td>1.9e-013</td>
<td>7.2e-017</td>
<td>2.3e-009</td>
<td>1.575656e+000</td>
</tr>
</tbody>
</table>

Stop: max(relative gap, infeasibilities) < 1.e-008

number of iterations = 8

gap = 2.28e-009
relative gap = 1.44e-009
infeasibilities = 1.35e-016

**Random SOCP problem. Problem size: \(m = 5, N = 5, n_i = 3, i = 1 : N\).**
VI. Conclusion and future work

The proposed reduced augmented system approach can improve the stability of IPM for SOCP. However, in our current implementation, we did not pay much attention to the efficient computation of the search direction. More work needs to be done to improve the efficiency of the matrix construction, factorization and solution of the reduced augmented system. In particular, great cares need to be taken to preserve the sparsity structure of the problem. Another important issue is the efficient handling of dense columns under the augmented system framework.

REFERENCES


