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 programming (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 stability problems. Numerical experiments show that the new approach can improve the stability.

I. Introduction

A second order cone programming (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 programming (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_i, we will write the vector as $x_i = (x_i^0\,;\,\bar{x}_i)$ with x_i^0 being the first component and $\bar{x}_i$ being the vector consisting of the remaining components. We use a similar notation for z_i. For x_i, we define

$$\mathrm{Arr}(x_i) = \begin{bmatrix} x_i^0 & x_i^1 & \cdots & x_i^{n_i} \\ x_i^1 & x_i^0 & & \\ \vdots & & \ddots & \\ x_i^{n_i} & & & x_i^0 \end{bmatrix}, \tag{1}$$

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

We consider the standard primal and dual SOCP problems:

$$\begin{array}{lll} \textbf{(P)} & \min & c_1^T x_1 + c_2^T x_2 + \cdots + c_N^T x_N \\ \textbf{s.t.} & & A_1 x_1 + A_2 x_2 + \cdots + A_N x_N \;=\; b \\ & & x_i \geq_K 0. \\ \textbf{(D)} & \max & b^T y \\ \textbf{s.t.} & & A_i^T y + z_i \;=\; c_i, \quad i = 1, \ldots, N \\ & & z_i \geq_K 0, \end{array} \tag{2}$$

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

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:

$$\begin{array}{rcll} Ax & = & b & \textbf{(Primal Feasibility)} \\ A^T y + z & = & c & \textbf{(Dual Feasibility)} \\ x_i \circ z_i & = & 0, \quad i = 1, \ldots, N, & \textbf{(Complementary)} \end{array} \tag{3}$$

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

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:

$$x_i \circ z_i \;=\; \mu e_i, \tag{4}$$

where e_i is the first unit vector in $\boldsymbol{R}^{n_i}$ and μ is a positive parameter that is to be driven to 0 explicitly. As

Department of Mathematics, National University of Singapore, 2 Science Drive 2, Singapore 117543, Singapore (mattohkc@math.nus.edu.sg).

High Performance Computing for Engineered Systems, Singapore-MIT Alliance, 4 Engineering Drive 3, Singapore 117576. (smap0035@nus.edu.sg).

Sloan School of Management, Massachusetts Institute of Technology. (rfreund@mit.edu).

The first and third authors are Fellows of the Singapore-MIT Alliance.

μ 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 μ 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 μ, the relaxed KKT condition is:

$$\begin{aligned} Ax &= b \\ A^T y + z &= c \\ \mathrm{Arr}(x)\,\mathrm{Arr}(z)e^0 &= \mu e^0, \end{aligned} \tag{5}$$

where $e^0 = [e_1\,;\, e_2 \quad \cdots\,;\, e_N]$. The matrix $\mathrm{Arr}(x) = \mathrm{diag}(\mathrm{Arr}(x_1), \cdots, \mathrm{Arr}(x_N))$ is a block diagonal matrix with $\mathrm{Arr}(x_1), \cdots, \mathrm{Arr}(x_N)$ as the diagonal blocks. The matrix $\mathrm{Arr}(z)$ 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:

$$\mathrm{Arr}(Fx)\,\mathrm{Arr}(F^{-1}z)e^0 \;=\; \mu e^0, \tag{6}$$

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

Let

$$f_i \;=\; \begin{bmatrix} f_i^0 \\ \bar{f}_i \end{bmatrix} \;=\; \frac{1}{\sqrt{2[\gamma(z_i)\gamma(x_i) + x_i^T z_i]}} \begin{bmatrix} \frac{1}{\omega_i} z_i^0 + \omega_i x_i^0 \\ \frac{1}{\omega_i}\bar{z}_i - \omega_i \bar{x}_i \end{bmatrix},$$

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

$$F_i \;=\; \omega_i \begin{bmatrix} f_i^0 & \bar{f}_i^T \\ \bar{f}_i & I + \dfrac{\bar{f}_i \bar{f}_i^T}{1 + f_i^0} \end{bmatrix}. \tag{7}$$

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

$$\begin{bmatrix} A & 0 & 0 \\ 0 & A^T & I \\ \mathrm{Arr}(F^{-1}z)F & 0 & \mathrm{Arr}(Fx)F^{-1} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \\ \Delta z \end{bmatrix} = \begin{bmatrix} r_p \\ r_d \\ r_c \end{bmatrix}, \tag{8}$$

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

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 Δz, the Newton equation (8) reduces to the so-called augmented system:

$$\begin{bmatrix} -F^2 & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} r \\ r_p \end{bmatrix}, \tag{9}$$

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

The augmented system can further be reduced by eliminating Δx to produce the normal equation:

$$\underbrace{AF^{-2}A^T}_{M} \Delta y \;=\; r_y \;:=\; r_p + AF^{-2}r. \tag{10}$$

The coefficient matrix $M := AF^{-2}A^T$ 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 μ 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 $F = \mathrm{diag}(F_1, \cdots, F_N)$. 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

$$F_i^2 \;=\; Q_i \Lambda_i Q_i^T, \tag{11}$$

where

$$\Lambda_i \;=\; \omega_i^2\, \mathrm{diag}((f_i^0 - \|\bar{f}_i\|)^2,\, (f_i^0 + \|\bar{f}_i\|)^2,\, 1, \cdots, 1), \tag{12}$$

$$Q_i \;=\; \begin{bmatrix} \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{2}} & 0 & \cdots & 0 \\ \frac{\bar{f}_i}{\sqrt{2}\|\bar{f}_i\|} & \frac{\bar{f}_i}{\sqrt{2}\|\bar{f}_i\|} & q_i^3 & \cdots & q_i^{n_i} \end{bmatrix}. \tag{13}$$

Notice that the first eigenvalue is the smallest and the second is the largest. The set $\{q_i^3, \cdots, q_i^{n_i}\}$ is an orthonormal basis of the subspace $\{v : v^T \bar{f}_i = 0\}$. To construct such an orthonormal basis, one may first construct the Householder matrix H_i associated with the vector $\bar{f}_i$, 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 effects 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 μ decreases due to an increasingly ill-conditioned Schur complement matrix. As a result this loss of accuracy in the computed solution, the primal infeasibility $\|r_p\|$ 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 ith 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_i^2 when $x_i \circ z_i = \mu e_i$ and μ is small. Note that $x_i^T z_i = \mu$.

Type 1: $\gamma(x_i^*) = 0$, $\gamma(z_i^*) = 0$. In this case, at the current iteration, $\gamma(x_i), \gamma(z_i) = \Theta(\sqrt{\mu})$, and $\omega_i = \Theta(1)$. This implies that $f_i^0, \|\bar{f}_i\| = \Theta(1/\sqrt{\mu})$. Thus The largest eigenvalue of F_i^2 is $\Theta(1/\mu)$ and by the fact that $\gamma(f_i) = 1$, the smallest eigenvalue of F_i^2 is $\Theta(\mu)$. The rest of the eigenvalues are $\Theta(1)$.

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

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

In general, eigenvalue structure of Type 1 occurs most frequently in practice. If the ith optimal dual solution is at the origin of the ith cone, then Type 2 occurs. Similarly, if the ith optimal primal solution is at the origin of the ith 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 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 is will not be severely affected by a small μ. 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 μ 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 μ 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 $Q = \mathrm{diag}(Q_1, \cdots, Q_N)$ and $D = \mathrm{diag}(\Lambda_1, \cdots, \Lambda_N)$. We can rewrite the augmented system (9) as follows.

$$\begin{bmatrix} -D & \widetilde{A}^T \\ \widetilde{A} & 0 \end{bmatrix} \begin{bmatrix} \Delta\widetilde{x} \\ \Delta y \end{bmatrix} = \begin{bmatrix} \widetilde{r} \\ r_p \end{bmatrix}, \tag{14}$$

where $\widetilde{A} = AQ$, $\widetilde{x} = Q^T x$, and $\widetilde{r} = Q^T r$.

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 $D = \mathrm{diag}(D_1, D_2)$ with $\mathrm{diag}(D_1)$ consists of the small eigenvalues of F^2 and $\mathrm{diag}(D_2)$ consists of the remaining eigenvalues. We also partition the matrices Q as $Q = [Q^{(1)} \;\; Q^{(2)}]$. Then $\widetilde{A}$ is partitioned as $\widetilde{A} = [\widetilde{A}_1 \;\; \widetilde{A}_2] = [AQ^{(1)} \;\; AQ^{(2)}]$. With such partitions, it is shown in [7] that the system (14) is equivalent to:

$$\begin{bmatrix} -D_1 E_1^{-1} & S_1^{-1/2}\widetilde{A}_1^T \\ \widetilde{A}_1 S_1^{-1/2} & \widetilde{A}\,\mathrm{diag}(S_1^{-1}, D_2^{-1})\widetilde{A}^T \end{bmatrix} \begin{bmatrix} S_1^{-1/2} E_1 \Delta\widetilde{x}_1 \\ \Delta y \end{bmatrix} = \begin{bmatrix} S_1^{-1/2}\widetilde{r} \\ r_p \end{bmatrix},$$

where E_1 is a given diagonal matrix that is usually chosen to be I and $S_1 = D_1 + E_1$. Here $\Delta\widetilde{x}_1 = (Q^{(1)})^T \Delta x$. We call the system in (15) the reduced augmented system. Note that once Δy is computed, $\Delta\widetilde{x}_2 = (Q^{(2)})^T \Delta x$ can be computed from the equation $\Delta\widetilde{x}_2 = D_2^{-1}(Q^{(2)})^T(A^T \Delta y - r)$. After that, Δx can be recovered from the equation $\Delta x = Q^{(1)}\Delta\widetilde{x}_1 + Q^{(2)}\Delta\widetilde{x}_2$.

Observe that the $(2,2)$ block of the coefficient matrix of the reduced augmented system has the same form as the Schur complement matrix $M = \widetilde{A}\,\mathrm{diag}(D_1^{-1}, D_2^{-1})\widetilde{A}^T$. But for the $(2,2)$ block, $\mathrm{diag}(S_1^{-1}, D_2^{-1}) = O(1)$, whereas for M, $\mathrm{diag}(D_1^{-1}, D_2^{-1}) = O(1/\mu)$. 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 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 $x^T z$ is denoted as** `gap`**.**

The numerical results show that the implementation based on 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 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.
*************************************************************
           Infeasible path-following algorithms
*************************************************************
it  pstep dstep p-infeas d-infeas  gap        obj
-------------------------------------------------------------

 0  0.000 0.000 9.7e-001 1.7e+000  2.2e+001  3.935775e+000
 1  1.000 1.000 5.0e-016 0.0e+000  2.7e+000  1.932067e+000
 2  0.945 0.902 2.8e-016 7.2e-017  2.5e-001  1.600947e+000
 :              :                  :         :
 :              :                  :         :
 7  0.985 1.000 2.9e-013 7.3e-017  1.6e-007  1.576568e+000
 8  0.989 0.994 1.9e-013 7.2e-017  2.3e-009  1.576568e+000

 Stop: max(relative gap, infeasibilities) < 1.00e-008
-------------------------------------------------------

 number of iterations   =  8
 gap                    = 2.28e-009
 relative gap           = 1.44e-009
 infeasibilities        = 1.93e-013
-------------------------------------------------------------
```

```
b) Implementation Based on Augmented Equation
*************************************************************
it  pstep dstep p-infeas d-infeas  gap        obj
-------------------------------------------------------------

 0  0.000 0.000 9.7e-001 1.7e+000  2.2e+001  3.935775e+000
 1  1.000 1.000 5.0e-016 0.0e+000  2.7e+000  1.932067e+000
 2  0.945 0.902 2.1e-016 1.2e-016  2.5e-001  1.600947e+000
 :              :                  :         :
 :              :                  :         :
 7  0.985 1.000 1.2e-016 1.5e-016  1.6e-007  1.576568e+000
 8  0.989 0.994 1.4e-016 1.1e-016  2.3e-009  1.576568e+000

 Stop: max(relative gap, infeasibilities) < 1.00e-008
-------------------------------------------------------

 number of iterations   =  8
 gap                    = 2.28e-009
 relative gap           = 1.44e-009
 infeasibilities        = 1.35e-016
------------------------------------------------------------
```

Dimacs Challenge problem: nb. Problem size: $m = 123$, $N = 793$, $n_i = 3$, $i = 1 : N$**; Linear block-4.**

```
a) Implementation Based on Normal Equation
*************************************************************
it  pstep dstep p-infeas d-infeas  gap        obj
-------------------------------------------------------------

 0  0.000 0.000 1.4e+003 5.8e+002  4.0e+004  0.000000e+000
 1  0.981 0.976 2.6e+001 1.4e+001  7.8e+002 -1.423573e+001
 2  1.000 0.989 3.7e-014 1.5e-001  2.7e+001 -1.351345e+001
 :               :                   :          :
 :               :                   :          :
13  0.672 0.775 2.4e-005 1.5e-008  2.4e-004 -5.059703e-002
14  0.227 0.754 2.8e-005 3.6-009  1.9e-004 -5.061571e-002

  Schur complement matrix not positive definite
-----------------------------------------------------

 number of iterations   = 15
 gap                    = 1.86e-004
 relative gap           = 1.86e-004
 infeasibilities        = 2.82e-005
------------------------------------------------------------
```

```
b) Implementation Based on Augmented Equations
*************************************************************
it  pstep dstep p-infeas d-infeas  gap        obj
-------------------------------------------------------------

 0  0.000 0.000 1.4e+003 5.8e+002  4.0e+004  0.000000e+000
 1  0.981 0.976 2.6e+001 1.4e+001  7.8e+002 -1.423573e+000
 2  1.000 0.989 4.3e-014 1.5e-001  2.7e+001 -1.351345e+001
 :                 :                 :          :
 :                 :                 :          :
13  0.674 0.775 1.6e-012 1.4e-008  2.4e-004 -5.059720e-002
14  0.222 0.748 1.0e-012 3.6e-009  1.9e-004 -5.061568e-002
 :                 :                 :          :
 :                 :                 :          :
23  0.549 0.532 9.9e-011 1.9e-014  1.3e-008 -5.070309e-002
24  0.533 0.943 1.5e-010 1.4e-015  7.1e-009 -5.070309e-002

 Stop: max(relative gap, infeasibilities) < 1.00e-008
-------------------------------------------------------

 number of iterations   = 24
 gap                    = 7.10e-009
 relative gap           = 7.10e-009
 infeasibilities        = 1.46e-010
------------------------------------------------------------
```

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
*************************************************************
it  pstep dstep p-infeas d-infeas  gap        obj
-------------------------------------------------------------

 0  0.000 0.000 1.0e+000 1.6e+002  2.2e+005  3.877983e+002
 1  1.000 0.805 4.7e-014 3.1e+001  5.0e+004  3.545945e+003
 2  1.000 0.958 7.1e-013 1.3e+000  3.3e+003  7.272815e+002
 :                 :                 :          :
 :                 :                 :          :
15  0.946 0.928 1.7e-006 2.4e-010  1.7e-004 -1.301225e+001
16  1.000 0.854 2.5e-005 3.5e-011  2.3e-005 -1.301227e+001

 Stop: relative gap < 5*infeasibility
-------------------------------------------------------

 number of iterations   = 16
 gap                    = 2.30e-005
 relative gap           = 1.76e-006
 infeasibilities        = 2.47e-005
-------------------------------------------------------------
```

```
b) Implementation Based on Augmented Equation
*************************************************************
it  pstep dstep p-infeas d-infeas  gap        obj
-------------------------------------------------------------

 0  0.000 0.000 1.0e+000 1.6e+002  2.2e+005  3.877983e+002
 1  1.000 0.805 4.7e-014 3.1e+001  5.0e+004  3.545945e+003
 2  1.000 0.958 4.5e-013 1.3e+000  3.3e+003  7.271454e+002
```

```
 :             :                :         :
 :             :                :         :
15  0.944 0.924 4.9e-013 2.6e-010  1.8e-004 -1.301225e+001
16  1.000 0.760 1.1e-011 6.2e-011  4.1e-005 -1.301227e+001
 :             :                :         :
 :             :                :         :
24  1.000 0.506 5.8e-012 2.2e-013  2.0e-007 -1.301227e+001
25  1.000 0.506 7.1e-012 1.1e-013  1.1e-007 -1.301227e+001

 Stop: max(relative gap, infeasibilities) < 1.00e-008
---------------------------------------------------
 number of iterations   = 25
 gap                    = 1.05e-007
 relative gap           = 8.10e-009
 infeasibilities        = 7.14e-012
---------------------------------------------------------
```

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

[1] F. Alizadeh and S. H. Schmieta, *Optimization with Semi-definite, Quadratic and Linear Constraints*, Rutcor Research Report RRR 23-97, Rutgers Center for Operations Research, Rutgers University, 1997. `http://rutcor.rutgers.edu/ rrr`

[2] K.D. Anderson, *A modified Schur complement method for handling dense columns in interior point methods for linear programming*, ACM Transactions on Mathematical Software.

[3] P. E. Gill, W. Murray, D. B. Poncelen and M. A. Saunders, Solving Reduced KKT Systems in Barrier Methods for Linear and Quadratic Programming, Tech. Report SOL 91-7, Systems Optimization Laboratory, Stanford University, 1991.

[4] M. S. Lobo, L. Vandenberghe, S. Boyd and H. Lebret, *Applications of Second-order Cone Programming*, Linear Algebra Appl., 284 (1998), pp.193–228.

[5] R. D. C. Monteiro and T. Tsuchiya, *Polynomial Convergence of Primal-Dual Algorithms for the Second-Order Cone Program Based on the MZ-Family of Directions*, Math. Programming, 88 (2000), pp. 61–83.

[6] G. Pataki and S. Schmieta, *The DIMACS Library of Mixed Semidefinite-Quadratic-Linear Programs*, `http://dimacs.rutgers.edu/Challenges/Seventh/Instances/`

[7] K. C. Toh, *Solving Ill-conditioned Symmetric Indefinite Systems by Iterative Methods*, Technical Report, National University of Singapore, 1999.

[8] K.C. Toh, M.J. Todd, and R.H. Ttnc, *SDPT3- A Matlab Software package for Semidefinite Programming*, Optimization Methods and Software, 11 (1999), pp. 545–581.

[9] R. H. Ttnc, K. C. Toh and M. J. Todd, *Solving Semidefinite-Quadratic-Linear Programming Using SDPT3*, Technical Report, National University of Singapore, 2001.

[10] R. J. Vanderbei, *Symmetric Quasidefinite Matrices*, SIAM J. Optimization, 5 (1995), pp. 100–113.