# Inexact Primal-Dual Path-Following Algorithms for Certain Quadratic SDPs

Kim Toh, Reha Tütüncü, and Mike Todd

School of Operations Research and Industrial Engineering,

Cornell University

http://www.orie.cornell.edu/~miketodd/todd.html

SDP Workshop, Singapore

January 12, 2006

# SDPT3

The code SDPT3 is joint work of Kim Toh, National University of Singapore, Reha Tütüncü, Goldman-Sachs, and me.

SDPT3: Primal-Dual Predictor-Corrector Interior-Point Method for SQLP.

Available from `http://www.math.nus.edu.sg/~mattohkc/sdpt3.html`.

Matlab-based (like SeDuMi), using C for computationally intensive parts (like CSDP), exploits sparsity (based on ideas from SDPA).

Obtains relatively high-precision solutions but is computationally expensive.

SDP: dense: $n \leq 500$; sparse $n \leq 2000$.

SOCP: sparse: $n \leq 150,000$.

LP: ditto.

# QSDP

The focus of this talk: application to the quadratic SDP:

$$(QSDP) \quad \min_X \ \tfrac{1}{2} X \bullet \mathcal{Q}(X) + C \bullet X$$
$$\mathcal{A}(X) \ = \ b, \quad X \ \succeq \ 0, \tag{1}$$

where $\mathcal{Q} : \mathcal{S}^n \to \mathcal{S}^n$ is a given self-adjoint positive semidefinite operator on $\mathcal{S}^n$ and $\mathcal{A} : \mathcal{S}^n \to I\!R^m$ is a linear map, with $(\mathcal{A}(X))_i = A_i \bullet X$ for each $i$.

Its dual is

$$(QSDD) \quad \max_{X,y,S} \ -\tfrac{1}{2} X \bullet \mathcal{Q}(X) + b^T y$$
$$\mathcal{A}^T(y) - \mathcal{Q}(X) + S \ = \ C, \quad S \ \succeq \ 0. \tag{2}$$

Note: the quadratic term involves the matrix variable $X$ not the vector variable $y$.

# Outline

- Application
- Newton systems
- Preconditioned SQMR
- Computational results

# Applications

Consider the semidefinite least-squares problem:

$$
\begin{array}{llll}
\min_X & \|\mathcal{L}(X) - \hat{K}\|_F & & \\
(SDLS) & \mathcal{A}(X) & = & b, \\
& X & \succeq & 0,
\end{array}
$$

i.e., find a feasible psd $X$ such that $\mathcal{L}(X)$ is as close as possible to $\hat{K}$.

This includes the closest correlation matrix problem and the nearest Euclidean distance matrix problem for a weighted graph.

Often $\mathcal{L}(X) = U^{1/2} X U^{1/2}$, and then $\mathcal{Q} = U \circledast U$,

where $U \circledast U(Z) := UZU^T$ when $Z \in \mathcal{S}^n$, $U \in \mathbb{R}^{n\times n}$.

# Newton Systems

At a given iteration of a predictor-corrector algorithm, given the current solution $X, y, S$, we need to solve for the search directions from

$$\begin{array}{rcccccl} -\mathcal{Q}(\Delta X) & + & \mathcal{A}^T(\Delta y) & + & \Delta S & = & R_d \\ \mathcal{A}(\Delta X) & & & & & = & r_p \\ \mathcal{E}(\Delta X) & & & + & \mathcal{F}(\Delta S) & = & R_c, \end{array} \quad (3)$$

where $\mathcal{E}$ and $\mathcal{F}$ are linear operators on $\mathcal{S}^n$ that depend on $X$ and $S$ and the symmetrization scheme chosen. If we eliminate $\Delta S$ and set

$$\mathcal{H} = \mathcal{F}^{-1}\mathcal{E} + \mathcal{Q}, \quad (4)$$

this reduces to ...

# Newton Systems, II

$$\begin{bmatrix} -\mathcal{H} & \mathcal{A}^T \\ \mathcal{A} & 0 \end{bmatrix} \begin{bmatrix} \Delta X \\ \Delta y \end{bmatrix} = \begin{bmatrix} R_h \\ r_p \end{bmatrix}. \tag{5}$$

By further eliminating $\Delta X$, we get the Schur complement equation of dimension $m$ below:

$$M \Delta y := \mathcal{A}\mathcal{H}^{-1}\mathcal{A}^T \Delta y \;=\; h. \tag{6}$$

[Note: if $\mathcal{Q}$ is zero and instead (QSDD) has objective function $b^T y - y^T Q y/2$, then we must solve $M \Delta y = h$ where

$$M = \mathcal{A}\mathcal{E}^{-1}\mathcal{F}\mathcal{A}^T + Q, \tag{7}$$

and this is much easier.]

# Form of $\mathcal{H}$

If we use Nesterov-Todd scaling and $\mathcal{Q} = U \circledast U$, then

$$\mathcal{H} = W^{-1} \circledast W^{-1} + U \circledast U, \tag{8}$$

where $W$ is the Nesterov-Todd scaling matrix. Note: this form of $\mathcal{H}$ also arises from a linear SDP with a simple upper bound on $X$:

$$\min C \bullet X, \qquad \mathcal{A}(X) = b, \quad 0 \preceq X \preceq U. \tag{9}$$

After adding a slack matrix, formulating the Newton equations, and simplifying, we find we need to solve

$$\mathcal{A}(\mathcal{F}_1^{-1}\mathcal{E}_1 + \mathcal{F}_2^{-1}\mathcal{E}_2)^{-1}\mathcal{A}^T \Delta y = h \tag{10}$$

for suitable $\mathcal{E}_i$ and $\mathcal{F}_i$: if we use Nesterov-Todd scaling we get a similar form.

# Computational problem

For $\mathcal{H}$ as above, we find

$$\mathcal{H}^{-1} = (P \circledast P)\Big(\mathcal{I} + D \circledast D\Big)^{-1}(P \circledast P)^T, \tag{11}$$

for an easily computed $P \in \mathbb{R}^{n\times n}$ and diagonal $D$, so that

$$(\mathcal{A}\mathcal{H}^{-1}\mathcal{A}^T)_{ij} = A_i \bullet (P \circledast P)\Big(\mathcal{I} + D \circledast D\Big)^{-1}(P \circledast P)^T A_j. \tag{12}$$

This is too expensive to compute for each $i$, $j$ in the large sparse setting. Why? If the diagonal term were missing, we would only have to compute the entries of $(PP^T)A_j(PP^T)$ corresponding to nonzero entries of $A_i$. So the diagonal operator is the culprit!

So we use preconditioned SQMR. We need to be able to compute $Mv$ for an arbitrary $v$, and solve $\widehat{M}w = g$ for an arbitrary $g$, where $\widehat{M} \approx M$.

# Approximation of $M$, I

We have two ways to approximate $M$. The first uses

$$\left(\mathcal{I} + D \circledast D\right)^{-1} \approx \sum_{k=1}^{q} \alpha_k \Lambda_k \circledast \Lambda_k, \tag{13}$$

where each $\alpha_k$ is a scalar and each $\Lambda_k$ is a diagonal matrix. This is done via an eigenvalue computation. Then $M$ is approximated by the preconditioner

$$\widehat{M} := \sum_{k=1}^{q} \alpha_k \mathcal{A}(V_k \circledast V_k)\mathcal{A}^T. \tag{14}$$

with $V_k = P\Lambda_k P^T$. Unfortunately, this may not be positive definite.

# Approximation of $M$, II

The second method approximates

$$\mathcal{H} = W^{-1} \circledast W^{-1} + U \circledast U \tag{15}$$

by $V \circledast V$.

We do not know how to solve this problem exactly, but when " $\circledast$ " is replaced by "$\otimes$" the solution is easily obtained by solving a $2 \times 2$ eigenvalue problem. With this $V$, we set the preconditioner to be

$$\widehat{M} = \mathcal{A}(V^{-1} \circledast V^{-1})\mathcal{A}^T. \tag{16}$$

This approximation is always positive definite. [Actually, we find it preferable to approximate $I \circledast I + D \circledast D$ in this way, and hence obtain an approximation of $\mathcal{H}^{-1}$ — see (11).]

# Computational Results

The next slides give some results for four classes of problems (the first three are nearest correlation matrix problems with a certain $\hat{K}$ and $U$, and the last linear SDPs with upper bounds) and four algorithms:

A1: Direct solution of the Schur complement equation;

A2: Solution of the equation using unpreconditioned SQMR;

A3: Solution of the equation using preconditioned SQMR:

the preconditioner is the first if it is positive definite, otherwise the second;

and

A4: Solution of the equation using preconditioned SQMR,

using the second preconditioner.

The dimension $n$ varies from 200 to 2000. The column "psqmr" gives the average number of PSQMR iterations to solve each Schur complement equation.

| | $n$ | A1 it | A1 $\phi$ | A1 Time (s) | A2 it | A2 $\phi$ | A2 Time (s) | A2 psqmr | A3 it | A3 $\phi$ | A3 Time (s) | A3 psqmr |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| E3 | 200 | 12 | 1.8e-08 | 26.5 | 12 | 2.5e-08 | 6.5 | 1.7 | 12 | 1.8e-08 | 7.7 | 1.0 |
| | 400 | 12 | 1.6e-08 | 269.6 | 12 | 2.1e-08 | 37.6 | 1.6 | 12 | 1.6e-08 | 44.3 | 1.0 |
| | 800 | 13 | 1.4e-08 | 3245.9 | 13 | 1.5e-08 | 269.0 | 1.7 | 13 | 5.2e-08 | 338.5 | 1.0 |
| | 1600 | | | | 14 | 6.8e-09 | 1932.0 | 1.7 | 14 | 6.7e-09 | 2302.3 | 1.1 |
| | 2000 | | | | 14 | 1.1e-08 | 3582.9 | 1.6 | 14 | 1.0e-08 | 4347.9 | 1.1 |
| E6 | 200 | 20 | 1.2e-08 | 54.2 | 21 | 9.6e-09 | 32.1 | 32.9 | 20 | 1.2e-08 | 13.4 | 1.1 |
| | 400 | 21 | 1.4e-08 | 487.7 | 23 | 2.5e-08 | 218.3 | 39.5 | 21 | 1.4e-08 | 76.7 | 1.0 |
| | 800 | 22 | 5.5e-08 | 5633.7 | 24 | 4.9e-08 | 1544.3 | 43.7 | 22 | 5.5e-08 | 522.6 | 1.0 |
| | 1600 | | | | 27 | 8.6e-08 | 16579.6 | 63.9 | 25 | 2.4e-08 | 4223.8 | 1.0 |
| | 2000 | | | | 28 | 8.0e-08 | 30566.7 | 62.1 | 26 | 1.1e-08 | 8632.0 | 1.0 |
| E9 | 200 | 12 | 2.0e-08 | 34.5 | 13 | 4.8e-08 | 17.1 | 41.4 | 12 | 2.0e-08 | 8.3 | 1.0 |
| | 400 | 13 | 1.8e-08 | 330.4 | 14 | 2.6e-08 | 119.8 | 45.9 | 13 | 1.8e-08 | 53.1 | 1.0 |
| | 800 | 13 | 9.5e-08 | 3623.9 | 14 | 9.4e-08 | 815.6 | 45.4 | 13 | 9.5e-08 | 359.4 | 1.0 |
| | 1600 | 15 | 2.5e-08 | 50155.2 | 16 | 2.1e-08 | 7243.8 | 46.9 | 15 | 2.5e-08 | 2964.0 | 1.0 |
| | 2000 | | | | 16 | 2.7e-08 | 13428.5 | 47.3 | 15 | 4.6e-08 | 5549.2 | 1.0 |
| E10 | 200 | 14 | 5.5e-08 | 55.2 | 15 | 6.1e-08 | 71.1 | 76.1 | 14 | 5.2e-08 | 18.5 | 1.0 |
| | 400 | 14 | 7.1e-08 | 325.2 | 14 | 6.9e-08 | 335.8 | 96.0 | 14 | 7.5e-08 | 106.7 | 1.4 |
| | 800 | 15 | 6.6e-08 | 3955.8 | 15 | 1.1e-07 | 4312.7 | 236.4 | 15 | 1.4e-07 | 508.8 | 1.3 |
| | 1600 | 17 | 2.7e-07 | 56457.0 | 17 | 3.3e-07 | 62418.6 | 440.9 | 17 | 1.3e-07 | 5818.5 | 1.3 |
| | 2000 | | | | 18 | 1.4e-06 | 136231.9 | 520.2 | 17 | 1.2e-07 | 9128.1 | 1.1 |

| A2 | | | | A3 | | | | A4 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| it | $\phi$ | Time (s) | psqmr | it | $\phi$ | Time (s) | psqmr | it | $\phi$ | Time (s) | psqmr |
| 12 | 2.5e-08 | 6.5 | 1.7 | 12 | 1.8e-08 | 7.7 | 1.0 | 12 | 4.6e-08 | 6.9 | 1.6 |
| 12 | 2.1e-08 | 37.6 | 1.6 | 12 | 1.6e-08 | 44.3 | 1.0 | 12 | 1.7e-08 | 39.1 | 1.6 |
| 13 | 1.5e-08 | 269.0 | 1.7 | 13 | 5.2e-08 | 338.5 | 1.0 | 13 | 1.4e-08 | 276.7 | 1.7 |
| 14 | 6.8e-09 | 1932.0 | 1.7 | 14 | 6.7e-09 | 2302.3 | 1.1 | 14 | 6.7e-09 | 1974.7 | 1.7 |
| 14 | 1.1e-08 | 3582.9 | 1.6 | 14 | 1.0e-08 | 4347.9 | 1.1 | 14 | 1.0e-08 | 3761.2 | 1.7 |
| 21 | 9.6e-09 | 32.1 | 32.9 | 20 | 1.2e-08 | 13.4 | 1.1 | 20 | 1.3e-08 | 19.0 | 14.7 |
| 23 | 2.5e-08 | 218.3 | 39.5 | 21 | 1.4e-08 | 76.7 | 1.0 | 21 | 4.2e-08 | 111.0 | 15.4 |
| 24 | 4.9e-08 | 1544.3 | 43.7 | 22 | 5.5e-08 | 522.6 | 1.0 | 22 | 7.3e-08 | 696.0 | 13.6 |
| 27 | 8.6e-08 | 16579.6 | 63.9 | 25 | 2.4e-08 | 4223.8 | 1.0 | 25 | 9.6e-08 | 6759.0 | 20.6 |
| 28 | 8.0e-08 | 30566.7 | 62.1 | 26 | 1.1e-08 | 8632.0 | 1.0 | 27 | 7.1e-08 | 14097.6 | 21.7 |
| 13 | 4.8e-08 | 17.1 | 41.4 | 12 | 2.0e-08 | 8.3 | 1.0 | 13 | 1.3e-08 | 8.2 | 6.2 |
| 14 | 2.6e-08 | 119.8 | 45.9 | 13 | 1.8e-08 | 53.1 | 1.0 | 14 | 9.8e-09 | 53.9 | 7.2 |
| 14 | 9.4e-08 | 815.6 | 45.4 | 13 | 9.5e-08 | 359.4 | 1.0 | 15 | 1.3e-08 | 401.7 | 8.2 |
| 16 | 2.1e-08 | 7243.8 | 46.9 | 15 | 2.5e-08 | 2964.0 | 1.0 | 16 | 2.0e-08 | 3185.9 | 9.7 |
| 16 | 2.7e-08 | 13428.5 | 47.3 | 15 | 4.6e-08 | 5549.2 | 1.0 | 17 | 1.0e-08 | 6379.2 | 10.2 |
| 15 | 6.1e-08 | 71.1 | 76.1 | 14 | 5.2e-08 | 18.5 | 1.0 | 14 | 8.2e-08 | 61.1 | 69.2 |
| 14 | 6.9e-08 | 335.8 | 96.0 | 14 | 7.5e-08 | 106.7 | 1.4 | 14 | 1.0e-07 | 313.7 | 66.6 |
| 15 | 1.1e-07 | 4312.7 | 236.4 | 15 | 1.4e-07 | 508.8 | 1.3 | 15 | 9.1e-08 | 650.8 | 12.9 |
| 17 | 3.3e-07 | 62418.6 | 440.9 | 17 | 1.3e-07 | 5818.5 | 1.3 | 17 | 7.7e-08 | 3917.3 | 6.4 |
| 18 | 1.4e-06 | 136231.9 | 520.2 | 17 | 1.2e-07 | 9128.1 | 1.1 | 17 | 8.6e-08 | 7046.7 | 5.4 |