Infeasible Primal-Dual
(Path-Following)
Interior-Point Methods
for Semidefinite Programming*

Yin Zhang
Dept of CAAM, Rice University

Outline

(1) Introduction
(2) Formulation & a complexity theorem
(3) Computation and numerical results
(4) Comments

Primal SDP:

\[
\begin{align*}
\min & \quad C \cdot X \\
\text{s.t.} & \quad A_i \cdot X = b_i, \quad i = 1 : m \\
& \quad X \succeq 0,
\end{align*}
\]

where \( C \cdot X = \text{tr}(C^T X) \), \( A_i, C \) symmetric, and \( X \succeq 0 \) means symmetric positive semidefinite.

**Diagonal** \( C, A_i \Rightarrow LP \)

SDP constraints are nonlinear:

\[
X_{2 \times 2} \succeq 0 \iff \{ x_{11} \geq 0, x_{11}x_{22} \geq x_{12}^2 \}
\]

Dual SDP:

\[
\begin{align*}
\max & \quad b \cdot y \\
\text{s.t.} & \quad \sum_{i=1}^{m} y_i A_i + Z = C \\
& \quad Z \succeq 0
\end{align*}
\]
Strong Duality:

\[ C \cdot X^* - b \cdot y^* = X^* \cdot Z^* = 0 \]

if both primal and dual are strictly feasible (Slater constraint qualification).

Note: For \( X, Z \succeq 0 \),

\[ X \cdot Z = 0 \iff XZ = 0. \]

Proof: “\( \Leftarrow \)” is trivial.

\[
X \cdot Z = \text{tr}(XZ) = 0 \\
\Rightarrow \text{tr}(XZ^{1/2}Z^{1/2}) = 0 \\
\Rightarrow \text{tr}(Z^{1/2}XZ^{1/2}) = 0 \\
\Rightarrow \sum \lambda_i(Z^{1/2}XZ^{1/2}) = 0 \\
\Rightarrow \lambda_i(Z^{1/2}XZ^{1/2}) = 0, \forall i \\
\Rightarrow Z^{1/2}XZ^{1/2} = 0 \text{ (symmetry)} \\
\Rightarrow (Z^{1/2}X^{1/2})(Z^{1/2}X^{1/2})^T = 0 \\
\Rightarrow Z^{1/2}X^{1/2} = 0 \\
\Rightarrow ZX = 0.
\]
**SDP Applications:**

**Eigenvalue optimization**

**Control theory**


**Combinatorial optimization**

— Example: Max-Cut relaxation

**Non-convex optimization**

— Example: QCQP Relaxations
**Definitions:**

Interior-point: \((X, y, Z) : X, Z \succ 0\)

Infeasible Alg: allows infeasible iterates

Primal-dual Alg: perturbed and damped Newton (or composite Newton) method applied to a form of optimality condition system.

**Standard Optimality Conditions:**

\((X, y, Z), X, Z \succeq 0\) such that

\[
\begin{align*}
\forall i \ A_i \cdot X - b_i &= 0 & \text{m} \\
\sum y_i A_i + Z - C &= 0 & \text{n(n + 1)/2} \\
XZ &= 0 & \text{n^2}
\end{align*}
\]

This optimality system is **not square**. Complementarity needs symmetrization.

**Weighted symmetrization:** (YZ 95)

\[H_P(M) = (PMP^{-1} + P^{-T}M^T P^T)/2\]

\[XZ = \tau I \iff H_P(XZ) = \tau I\]

— complementarity \(\iff H_P(XZ) = 0\)

— the central path \(\iff H_P(XZ) = \mu I\)
$H_P$ unifies a family of primal-dual formulations.  
3 most promising members:

**AHO** (Alizadeh/Haeberly/Overton 94):

\[ P = I \Rightarrow XZ + ZX = 0 \]

**KSH** (Kojima et al 94, Monteiro 95):

\[ P^k = [Z^k]^{1/2} \Rightarrow Z^kXZ + ZXZ^k = 0 \]

**NT** (Nesterov/Todd 95):

\[ W = X^{.5}(X^{.5}ZX^{-.5})^{-5}X^{.5}, \quad P^k = [W^k]^{-5} \]

A sequence of equiv. systems for varying $P^k$:

<table>
<thead>
<tr>
<th>Formulation</th>
<th>AHO</th>
<th>KSH</th>
<th>NT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real Newton?</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
</tbody>
</table>

AHO — difficult to analyze its complexity

KSH — Complexity results were obtained

for **feasible** or **short-step** methods

In favor of: **infeasible long-step methods**

implementable and most efficient in practice
Central path:

\[
\{(X, Z) \succeq 0 : XZ = \mu I \succeq 0\}
\]

On the central path: \(\lambda_i(XZ) = \mu, \forall i\)

Short-step uses a narrow neighborhood.

Long-step uses a wide neighborhood.
A narrow neighborhood: \( \alpha \in (0, 1) \)

\[
\mathcal{N}_\alpha = \{(X, Z) \succ 0 : \|XZ - \mu I\|_F \leq \alpha \mu\}
\]

where \( \mu \) is the normalized duality gap

\[
\mu = \frac{\text{tr}(XZ)}{n}.
\]

A wide neighborhood: \( \gamma \in (0, 1) \)

\[
\mathcal{N}_\gamma = \{(X, Z) \succ 0 : \lambda_{\text{min}}(XZ) \geq \gamma \mu\}
\]

where \( \mu = \frac{\text{tr}(XZ)}{n} \).

A neighborhood with feasibility priority:

\[
\widehat{\mathcal{N}}_\gamma = \left\{(X, Z) \in \mathcal{N}_\gamma : \frac{r}{r^0} \leq \frac{\mu}{\mu^0}\right\}
\]

where \( r \) measures infeasibility.
Column-Stacking Operator $\text{vec} : \mathbb{R}^{n \times n} \rightarrow \mathbb{R}^{n^2}$

$\text{vec} \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} = (1 \ 2 \ 3 \ 4)^T$

**SDP:** Find $v = (X, y, Z), X, Z \succeq 0$, such that

$$F_k(v) = \begin{pmatrix} \mathcal{A}^T y + \text{vec} Z - \text{vec} C \\ \mathcal{A}(\text{vec} X) - b \\ \text{vec}(Z^k X Z + ZX Z^k) \end{pmatrix} = 0,$$

where $\mathcal{A}^T = [\text{vec} A_1 \cdots \text{vec} A_m]$.

**Infeasible long-step Algorithm:** (YZ 95)

| Choose $v^0 = (X^0, y^0, Z^0), X^0, Z^0 \succ 0,$ |
| For $k = 0, 1, 2, \ldots$, until $\mu^k \leq 2^{-L}$ do |
| (1) solve $F'_k(v^k) \Delta v = -F_k(v^k) + \sigma \mu^k p^k$ |
| (2) choose $\alpha$, $v^{k+1} \leftarrow v^k + \alpha \Delta v \in \partial \tilde{N}$ |
| End |

where $\sigma \in (0, 1)$, and the centering term $p^k$ is derived from

$$Z^k X Z + ZX Z^k - 2\sigma \mu^k Z^k = 0.$$
The key to convergence analysis is to estimate the 2nd-order term.

**Key Lemma:** (YZ 95)
Under the conditions that
(1) a solution \((X^*, y^*, Z^*)\) exists,
(2) \(X^0 = Z^0 = \rho I\) for
\[
\rho \geq \max \left\{ \lambda_1(X^*), \lambda_1(Z^*), \frac{\text{tr}(X^* + Z^*)}{n} \right\} \quad (*)
\]
(3) \((X, Z) \in \mathcal{N}_\gamma^c\),
\[
\|Z^{1/2}(\Delta X \Delta Z)Z^{-1/2}\|_F \leq 19n \sqrt{\frac{\lambda_1(XZ)}{\lambda_n(XZ)}} \frac{\text{tr}(XZ)}{\gamma}
\]

**Polynomial Complexity:** (YZ 95)
**Theorem:** Under the conditions (1)-(3), the algorithm terminates \((\mu^k \leq 2^{-L})\) in at most \(O(n^{2.5}L)\)-iterations.

Global convergence holds without \((*)\) in (2).
Computation:
How difficult is it to solve SDP?

Distinctions from LP:
(1) Special code for special prob., e.g. LP
(2) Cost for linear system varies with method
(3) Parallelism more important than sparsity

Linear system $F_k^l(v^k)\Delta v = -F_k(v^k) + \sigma \mu^k p^k$:

$$
\begin{bmatrix}
0 & A^T & I \\
A & 0 & 0 \\
E & 0 & F
\end{bmatrix}
\begin{bmatrix}
\text{vec}\Delta X \\
\Delta y \\
\text{vec}\Delta Z
\end{bmatrix}
= 
\begin{bmatrix}
\text{vec}R_d \\
r_p \\
\text{vec}R_c
\end{bmatrix},
$$

$E$ and $F$ are no longer diagonal.

Solution Procedure:
(1) $M = A(E^{-1}F)A^T$.
(2) $M \Delta y = r_p + A[E^{-1}(F \text{vec}R_d - \text{vec}R_c)]$.
(3) $\Delta Z = R_d - \sum_{i=1}^{m} \Delta y_i A_i$.
(4) $\Delta X = \sigma \mu Z^{-1} - X - H(X(\Delta Z)Z^{-1})$.

Most computation is in steps (1)-(2).
$M$ is $m \times m$, $m \in [0, n(n + 1)/2]$. 
Consider forming:  \[ M = \mathcal{A}(\mathcal{E}^{-1}\mathcal{F})\mathcal{A}^T \]
Notation: Kronecker product \( U \otimes V = [u_{ij}V] \)

**KSH:**
\[
\mathcal{E} = Z \otimes Z, \quad \mathcal{F} = (ZX \otimes I + I \otimes ZX)/2
\]
\( M \succ 0, \quad M_{ij} = \text{tr}(Z^{-1}A_iXA_j) \)

**AHO:**
\[
\mathcal{E} = Z \otimes I + I \otimes Z, \quad \mathcal{F} = X \otimes I + I \otimes X
\]
\( M \) asymmetric (LU)
\[
ZP_i + P_iZ = A_i \quad \text{Lyapunov eqn for } P_i
\]
\( M_{ij} = \text{tr}(P_i(XA_j + A_jX)) \)

**NT:**
\[
W = X^{.5}(X^{.5}ZX^{.5})^{-.5}X^{.5}
\]
\[
\mathcal{E} = I \otimes I, \quad \mathcal{F} = W \otimes W
\]
\( M \succ 0, \quad M_{ij} = \text{tr}(A_iWA_jW) \)

2 Cholesky + SVD for \( W \) (TTT 96)

work(AHO) > work(NT) > work(KSH)
Leading Cost* per Iter for Dense Problems
(Form: $O(m(m + n)n^2)$, Solve $O(m^3)$)

<table>
<thead>
<tr>
<th></th>
<th>$O(1)$</th>
<th>$O(n)$</th>
<th>$O(n^2)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Form</td>
<td>$O(n^3)^*$</td>
<td>$O(n^4)^*$</td>
<td>$O(n^6)^*$</td>
</tr>
<tr>
<td>Solve</td>
<td>$O(1)$</td>
<td>$O(n^3)$</td>
<td>$O(n^6)^*$</td>
</tr>
</tbody>
</table>

Forming $M$ is most expensive.

**Bad news:** $O(n^4)$ per iteration most likely

**Good news:** **Rich parallelism**

**Coarse grain:** compute $M_{ij}$ independently

**Fine grain:** rich in matrix multiplications

Expect (almost) linear speed-up
Preliminary Implementation:

Predictor-Corrector Algorithm:

Choose \( v^0 = (X^0, y^0, Z^0) \), \( X^0, Z^0 \succ 0 \),

For \( k = 0, 1, 2, \ldots \), until \( \mu^k \leq 2^{-L} \) do

(1) \( F_k'(v^k) \Delta v^p = -F_k(v^k) \). Choose \( \sigma^k \).

(2) \( F_k'(v^k) \Delta v^c = -F_k(v^k + \Delta v^p) + \sigma^k \mu^k p^k \)

(3) Let \( \Delta v = \Delta v^p + \Delta v^c \)

(4) choose \( \alpha \), \( v^{k+1} \leftarrow v^k + \alpha \Delta v \) (interior)

End

(Extension of Mehrotra’s framework to SDP)

— implemented in MATLAB
— for general problems, dense or sparse
— only KSH formulation at this moment
— backtracking for \( \alpha \) using Cholesky
— no stabilizing measure yet at the end
— initial point \( X^0 = Z^0 = \sqrt{n}I \)
— tested on randomly generated problems
— run on SGI workstation (MIPS/R8000)
Max-Cut: $n = 50$ (general code)

<table>
<thead>
<tr>
<th>Iter. No</th>
<th>P_inf</th>
<th>D_inf</th>
<th>Gap</th>
<th>Cpu</th>
</tr>
</thead>
<tbody>
<tr>
<td>iter 0</td>
<td>4.29e+01</td>
<td>1.84e+02</td>
<td>6.38e+03</td>
<td>0.21</td>
</tr>
<tr>
<td>iter 1</td>
<td>2.13e+01</td>
<td>1.16e-14</td>
<td>1.85e+03</td>
<td>0.30</td>
</tr>
<tr>
<td>iter 2</td>
<td>1.06e-14</td>
<td>1.05e-14</td>
<td>2.73e+02</td>
<td>0.32</td>
</tr>
<tr>
<td>iter 3</td>
<td>5.45e-15</td>
<td>1.05e-14</td>
<td>1.53e+02</td>
<td>0.30</td>
</tr>
<tr>
<td>iter 4</td>
<td>2.49e-15</td>
<td>1.34e-14</td>
<td>2.05e+01</td>
<td>0.32</td>
</tr>
<tr>
<td>iter 5</td>
<td>4.01e-15</td>
<td>1.33e-14</td>
<td>3.95e+00</td>
<td>0.31</td>
</tr>
<tr>
<td>iter 6</td>
<td>6.14e-14</td>
<td>1.23e-14</td>
<td>3.53e-01</td>
<td>0.32</td>
</tr>
<tr>
<td>iter 7</td>
<td>7.97e-13</td>
<td>1.23e-14</td>
<td>3.39e-02</td>
<td>0.31</td>
</tr>
<tr>
<td>iter 8</td>
<td>2.83e-11</td>
<td>1.18e-14</td>
<td>7.43e-03</td>
<td>0.30</td>
</tr>
<tr>
<td>iter 9</td>
<td>6.16e-10</td>
<td>1.41e-14</td>
<td>2.59e-04</td>
<td>0.30</td>
</tr>
<tr>
<td>iter 10</td>
<td>6.25e-09</td>
<td>1.27e-14</td>
<td>7.96e-06</td>
<td>Conv</td>
</tr>
</tbody>
</table>

[m n] = [50 50] mflops = 2.39e+01
CPU: form 1.86, factor 0.04, total 3.11.

Cpu/iter can be cut to 0.05 if using the special structure $\text{diag}(X) = e \Rightarrow A_i = e_i e_i^T$. 

15
**Max-Cut:** \( n = 200 \) (general code)

<table>
<thead>
<tr>
<th>Iter. No</th>
<th>( P_{\text{inf}} )</th>
<th>( D_{\text{inf}} )</th>
<th>Gap</th>
<th>Cpu</th>
</tr>
</thead>
<tbody>
<tr>
<td>iter  0:</td>
<td>1.86e+02</td>
<td>1.20e+03</td>
<td>1.97e+05</td>
<td>3.72</td>
</tr>
<tr>
<td>iter  1:</td>
<td>9.22e+01</td>
<td>9.40e-14</td>
<td>7.48e+04</td>
<td>12.56</td>
</tr>
<tr>
<td>iter  2:</td>
<td>7.58e-14</td>
<td>1.04e-13</td>
<td>3.51e+03</td>
<td>12.66</td>
</tr>
<tr>
<td>iter  3:</td>
<td>1.25e-14</td>
<td>9.10e-14</td>
<td>9.08e+02</td>
<td>13.00</td>
</tr>
<tr>
<td>iter  4:</td>
<td>8.53e-15</td>
<td>1.03e-13</td>
<td>5.94e+02</td>
<td>13.04</td>
</tr>
<tr>
<td>iter  5:</td>
<td>6.21e-15</td>
<td>1.14e-13</td>
<td>1.09e+02</td>
<td>13.18</td>
</tr>
<tr>
<td>iter  6:</td>
<td>8.23e-15</td>
<td>9.29e-14</td>
<td>8.52e+00</td>
<td>13.00</td>
</tr>
<tr>
<td>iter  7:</td>
<td>1.70e-12</td>
<td>1.07e-13</td>
<td>9.40e-01</td>
<td>12.66</td>
</tr>
<tr>
<td>iter  8:</td>
<td>3.84e-11</td>
<td>8.79e-14</td>
<td>9.82e-02</td>
<td>12.89</td>
</tr>
<tr>
<td>iter  9:</td>
<td>3.23e-08</td>
<td>9.79e-14</td>
<td>1.66e-02</td>
<td>13.24</td>
</tr>
<tr>
<td>iter 10:</td>
<td>1.12e-07</td>
<td>1.01e-13</td>
<td>7.80e-03</td>
<td>12.63</td>
</tr>
<tr>
<td>iter 11:</td>
<td>2.09e-08</td>
<td>8.87e-14</td>
<td>8.22e-04</td>
<td>conv</td>
</tr>
</tbody>
</table>

\([m \ n] = [200 \ 200] \quad \text{mflops} = 1.61e+03\)

CPU: form 110.03, factor 0.73, total 133.08

**Note:** deteriorating Primal feasibility
**General Problem:**  $m = 50, n = 200$

<table>
<thead>
<tr>
<th>Iter. No</th>
<th>P_inf</th>
<th>D_inf</th>
<th>Gap</th>
<th>Cpu</th>
</tr>
</thead>
<tbody>
<tr>
<td>iter 0:</td>
<td>3.34e+03</td>
<td>2.40e+02</td>
<td>5.65e+02</td>
<td>4.78</td>
</tr>
<tr>
<td>iter 1:</td>
<td>6.49e-12</td>
<td>8.05e+01</td>
<td>1.48e+03</td>
<td>255.71</td>
</tr>
<tr>
<td>iter 2:</td>
<td>2.65e-11</td>
<td>8.43e+00</td>
<td>5.31e+02</td>
<td>255.67</td>
</tr>
<tr>
<td>iter 3:</td>
<td>2.01e-11</td>
<td>1.05e+00</td>
<td>1.30e+02</td>
<td>256.10</td>
</tr>
<tr>
<td>iter 4:</td>
<td>2.38e-11</td>
<td>1.55e-01</td>
<td>2.57e+01</td>
<td>255.99</td>
</tr>
<tr>
<td>iter 5:</td>
<td>2.55e-11</td>
<td>1.97e-02</td>
<td>3.97e+00</td>
<td>256.34</td>
</tr>
<tr>
<td>iter 6:</td>
<td>2.77e-11</td>
<td>2.36e-03</td>
<td>6.18e-01</td>
<td>256.23</td>
</tr>
<tr>
<td>iter 7:</td>
<td>1.89e-10</td>
<td>2.36e-04</td>
<td>5.46e-01</td>
<td>256.16</td>
</tr>
<tr>
<td>iter 8:</td>
<td>4.40e-10</td>
<td>6.62e-05</td>
<td>2.21e-01</td>
<td>255.99</td>
</tr>
<tr>
<td>iter 9:</td>
<td>4.26e-09</td>
<td>1.85e-05</td>
<td>2.61e-02</td>
<td>256.16</td>
</tr>
<tr>
<td>iter 10:</td>
<td>7.18e-09</td>
<td>1.85e-06</td>
<td>2.45e-03</td>
<td>255.85</td>
</tr>
<tr>
<td>iter 11:</td>
<td>1.24e-07</td>
<td>1.85e-07</td>
<td>4.55e-04</td>
<td>Conv</td>
</tr>
</tbody>
</table>

[m n] = [ 50 200]  mflops = 3.11e+03
CPU: form 2528.25, factor 0.04, total 2566

**Note:** Almost all time spent on forming $M$
Final remarks:

Superlinear convergence:

Proved for “tangential convergence” along the central path, expensive to enforce.
— Kojima et al (95)
— Potra and Sheng (95)
— Luo, Sturm and S. Zhang (96)

Trouble:
off-diag($XZ$)$\to 0$ slower than diag($XZ$)$\to 0$
unless $XZ \approx \mu I \to 0$.

Is it worth the effort?

Parallel implementation:

A necessity for solving large-scale problems