Partial facial reduction: simplified, equivalent SDPs via approximations of the PSD cone

Frank Permenter (Joint with Pablo Parrilo)

Massachusetts Institute of Technology

Oct 6th, 2014

Our pre-processing philosophy: do simple things quickly.

"The strategy of detecting **simple** forms of redundancy, but doing it **fast**, seems to be the **best** strategy."
– Andersen and Andersen, Presolving in linear programming.

This talk:

- Pre-processing technique based on *facial reduction* (Borwein, Wolkowicz '81) consistent with this philosophy.

I'll also discuss:

- Dual solution recovery.
- A software implementation (`frlib`).

Facial reduction applies to semidefinite programs not strictly feasible.

- SDP feasible set is intersection of subspace with PSD cone

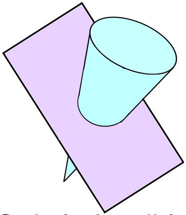
$$\begin{array}{ll} \text{minimize} & C \cdot X \\ \text{subject to} & A_i \cdot X = b_i \quad \forall i \in \{1, \ldots, m\} \\ & X \in \mathbb{S}^n_+ \end{array}$$

Facial reduction applies to semidefinite programs not strictly feasible.

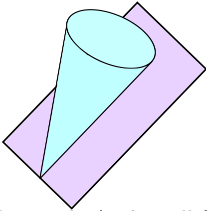
- SDP feasible set is intersection of subspace with PSD cone

$$\begin{array}{ll} \text{minimize} & C \cdot X \\ \text{subject to} & A_i \cdot X = b_i \quad \forall i \in \{1, \ldots, m\} \\ & X \in \mathbb{S}^n_+ \end{array}$$

- Strictly feasible when subspace intersects interior of cone—i.e. if subspace contains positive *definite* matrix



Strictly feasible



Not strictly feasible

Example: strict feasibility can fail in SDP-based bounds of completely-positive rank.

The following SDP (Fawzi, et al '14) bounds the *completely positive rank* of a matrix A:

$$\begin{array}{ll} \text{minimize } t & \\ \text{subject to} & \begin{pmatrix} t & \text{vect } A^T \\ \text{vect } A & X \end{pmatrix} \in \mathbb{S}^n_+ \\ & X_{ij,ij} \leq A_{ij}^2 \\ & (\text{ additional constraints}) \end{array}$$

i.e. it bounds smallest R for which

$$A = \sum_{i=1}^{R} v_i v_i^T \quad v_i \geq 0.$$

.

Example: strict feasibility can fail in SDP-based bounds of completely-positive rank.

The following SDP (Fawzi, et al '14) bounds the *completely positive rank* of a matrix A:

$$\begin{array}{ll} \text{minimize } t & \\ \text{subject to} & \begin{pmatrix} t & \text{vect } A^T \\ \text{vect } A & X \end{pmatrix} \in \mathbb{S}^n_+ \\ & X_{ij,ij} \leq A_{ij}^2 \\ & (\text{ additional constraints}) \end{array}$$

i.e. it bounds smallest R for which

$$A = \sum_{i=1}^{R} v_i v_i^T \quad v_i \geq 0.$$

.

Strict feasibility fails if any A_{ij} is zero!

Example: strict feasibility can fail in SDP-based tests of polynomial non-negativity.

Let $p(x)$ be a vector of polynomials. Then, the polynomial $f(x)$ is a *sum-of-squares* if there exists Q that solves:

$$\begin{aligned} &\text{Find } Q \in \mathbb{S}^n_+ \\ &\text{subject to} \quad \underbrace{f(x) = p(x)^T Q p(x)}_{\text{Linear constraints}} \end{aligned}$$

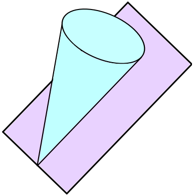
Example: strict feasibility can fail in SDP-based tests of polynomial non-negativity.

Let $p(x)$ be a vector of polynomials. Then, the polynomial $f(x)$ is a *sum-of-squares* if there exists Q that solves:

$$\text{Find } Q \in \mathbb{S}^n_+$$
$$\text{subject to} \quad \underbrace{f(x) = p(x)^T Q p(x)}_{\text{Linear constraints}}$$

Strict feasibility fails if $p(x) \neq 0$ at roots of $f(x)$.

If strict feasibility fails, SDPs can be simplified.

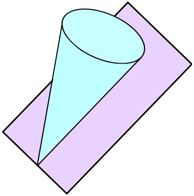


Find $x_1, x_2, x_3, x_4 \in \mathbb{R}$

subject to

$$X = \begin{pmatrix} x_1 & 0 & 0 & 0 \\ 0 & -x_1 & x_2 & 0 \\ 0 & x_2 & x_2 + x_3 & 0 \\ 0 & 0 & 0 & x_4 \end{pmatrix} \in \mathbb{S}^4_+$$

If strict feasibility fails, SDPs can be simplified.

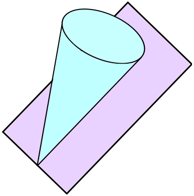


Find $x_1, x_2, x_3, x_4 \in \mathbb{R}$
subject to

$$X = \begin{pmatrix} x_1 & 0 & 0 & 0 \\ 0 & -x_1 & x_2 & 0 \\ 0 & x_2 & x_2 + x_3 & 0 \\ 0 & 0 & 0 & x_4 \end{pmatrix} \in \mathbb{S}^4_+$$

$v^T X v = 0$ for $v = (1, 1, 0, 0)^T$—i.e. strict feasibility fails.

If strict feasibility fails, SDPs can be simplified.



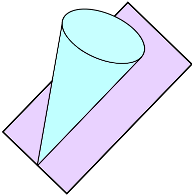
$$\begin{array}{ll} \text{Find} & x_1, x_2, x_3, x_4 \in \mathbb{R} \\ \text{subject to} & \\ & X = \begin{pmatrix} x_1 & 0 & 0 & 0 \\ 0 & -x_1 & x_2 & 0 \\ 0 & x_2 & x_2 + x_3 & 0 \\ 0 & 0 & 0 & x_4 \end{pmatrix} \in \mathbb{S}^4_+ \end{array}$$

$v^T X v = 0$ for $v = (1, 1, 0, 0)^T$—i.e. strict feasibility fails.

Equivalent reformulation:

$$\begin{array}{ll} \text{Find} & x_1, x_2, x_3, x_4 \in \mathbb{R} \\ \text{subject to} & \\ & x_1 = x_2 = 0, \qquad \begin{pmatrix} x_3 & 0 \\ 0 & x_4 \end{pmatrix} \in \mathbb{S}^2_+ \end{array}$$

If strict feasibility fails, SDPs can be simplified.



Find $x_1, x_2, x_3, x_4 \in \mathbb{R}$
subject to

$$X = \begin{pmatrix} x_1 & 0 & 0 & 0 \\ 0 & -x_1 & x_2 & 0 \\ 0 & x_2 & x_2 + x_3 & 0 \\ 0 & 0 & 0 & x_4 \end{pmatrix} \in \mathbb{S}^4_+$$

$v^T X v = 0$ for $v = (1, 1, 0, 0)^T$—i.e. strict feasibility fails.

Equivalent reformulation:

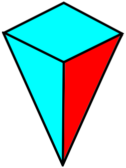
Find $x_1, x_2, x_3, x_4 \in \mathbb{R}$
subject to

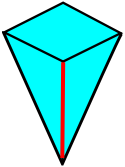
$$x_1 = x_2 = 0, \qquad \begin{pmatrix} x_3 & 0 \\ 0 & x_4 \end{pmatrix} \in \mathbb{S}^2_+$$

If strict feasibility fails, such a reformulation *always* exists.

Simplifications arise by reformulating SDP over a face containing feasible set.

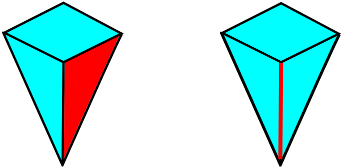
- What is a face? For a polyhedral cone:





Simplifications arise by reformulating SDP over a face containing feasible set.

- What is a face? For a polyhedral cone:

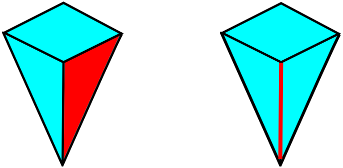


- For the PSD cone, a face is the subset of matrices with range contained in a given subspace S

$$\mathcal{F}_S := \left\{X \in \mathbb{S}^n_+ : \text{range}\, X \subseteq S\right\}$$

Simplifications arise by reformulating SDP over a face containing feasible set.

- What is a face? For a polyhedral cone:



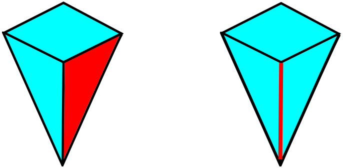
- For the PSD cone, a face is the subset of matrices with range contained in a given subspace S

$$\mathcal{F}_S := \{X \in \mathbb{S}^n_+ : \text{range}\, X \subseteq S\}$$

For subspaces A, B,
$A \subseteq B \Rightarrow \mathcal{F}_A \subseteq \mathcal{F}_B$.

Simplifications arise by reformulating SDP over a face containing feasible set.

- What is a face? For a polyhedral cone:



- For the PSD cone, a face is the subset of matrices with range contained in a given subspace S

$$\mathcal{F}_S := \left\{ X \in \mathbb{S}^n_+ : \text{range}\, X \subseteq S \right\}$$

For subspaces A, B,
$A \subseteq B \Rightarrow \mathcal{F}_A \subseteq \mathcal{F}_B$.

For $X \in \mathbb{S}^n_+$,
$\mathcal{F}_{\text{null}\, X} = X^\perp \cap \mathbb{S}^n_+$.

Faces can be parametrized using smaller PSD cones, which yields smaller SDPs.

- Fix $U \in \mathbb{R}^{n \times d}$. The following holds:

$$\begin{matrix} X \in \mathbb{S}^n_+ \\ \text{range } X \subseteq \text{range } U \end{matrix} \quad \Leftrightarrow \quad X = U \underbrace{\hat{X}}_{\in \mathbb{S}^d_+} U^T$$

Faces can be parametrized using smaller PSD cones, which yields smaller SDPs.

- Fix $U \in \mathbb{R}^{n \times d}$. The following holds:

$$\begin{array}{c} X \in \mathbb{S}^n_+ \\ \text{range}\, X \subseteq \text{range}\, U \end{array} \Leftrightarrow X = U \underbrace{\hat{X}}_{\in \mathbb{S}^d_+} U^T$$

- Containment of feasible set in a face yields reformulation

$$\begin{array}{ll} \text{minimize} & C \cdot X \\ \text{subject to} & A_i \cdot X = b_i \\ & X \in \mathbb{S}^n_+ \end{array} \quad \xleftrightarrow[\text{problems}]{\text{equivalent}} \quad \begin{array}{ll} \text{minimize} & C \cdot U\hat{X}U^T \\ \text{subject to} & A_i \cdot U\hat{X}U^T = b_i \\ & \hat{X} \in \mathbb{S}^d_+ \end{array}$$

Faces can be parametrized using smaller PSD cones, which yields smaller SDPs.

- Fix $U \in \mathbb{R}^{n \times d}$. The following holds:

$$\begin{array}{c} X \in \mathbb{S}^n_+ \\ \text{range}\, X \subseteq \text{range}\, U \end{array} \Leftrightarrow X = U \underbrace{\hat{X}}_{\in \mathbb{S}^d_+} U^T$$

- Containment of feasible set in a face yields reformulation

$$\begin{array}{ll} \text{minimize} & C \cdot X \\ \text{subject to} & A_i \cdot X = b_i \\ & X \in \mathbb{S}^n_+ \end{array} \quad \xleftrightarrow[\text{problems}]{\text{equivalent}} \quad \begin{array}{ll} \text{minimize} & C \cdot U\hat{X}U^T \\ \text{subject to} & A_i \cdot U\hat{X}U^T = b_i \\ & \hat{X} \in \mathbb{S}^d_+ \end{array}$$

How do you find a face containing feasible set?

Facial reduction is technique for finding a face.

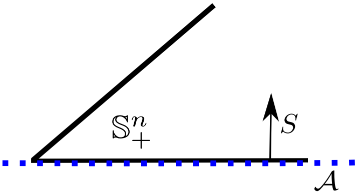
Approaches:

- Borwein and Wolkowicz ’81. Original algorithm.
- Ramana ’97. Generalized SDP dual.
- Pataki ’13. Simplifies ’81, generalizes ’97 to other cones.
- Waki and Muramatsu ’13. Simplifies ’81.
- Cheung and Wolkowicz ’13. Numerical stability.
- Other application specific methods (e.g. Krislock et al. ’10)

Finding faces is a search problem over the dual cone.

- Let $\mathcal{A}$ denote solutions to $A_i \cdot X = b_i$ and let S solve:

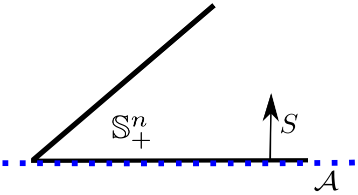
$$\begin{array}{ll} \text{Find} & S \in (\mathbb{S}^n_+)^* \\ \text{subject to} & S^\perp \text{ contains } \mathcal{A} \end{array}$$



Finding faces is a search problem over the dual cone.

- Let $\mathcal{A}$ denote solutions to $A_i \cdot X = b_i$ and let S solve:

$$\begin{array}{cr} \text{Find} & S \in (\mathbb{S}^n_+)^* \\ \text{subject to} & S^\perp \text{ contains } \mathcal{A} \end{array}$$

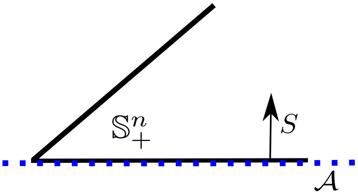


- Then, the face $\mathbb{S}^n_+ \cap S^\perp$ contains feasible set $\mathcal{A} \cap \mathbb{S}^n_+$.

Finding faces is a search problem over the dual cone.

- Let $\mathcal{A}$ denote solutions to $A_i \cdot X = b_i$ and let S solve:

$$\begin{array}{cr} \text{Find} & S \in (\mathbb{S}^n_+)^* \\ \text{subject to} & S^\perp \text{ contains } \mathcal{A} \end{array}$$

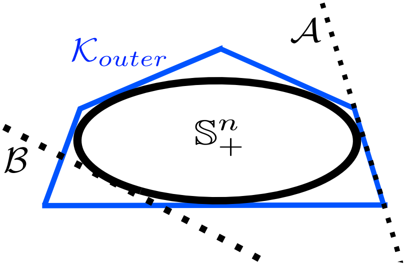


- Then, the face $\mathbb{S}^n_+ \cap S^\perp$ contains feasible set $\mathcal{A} \cap \mathbb{S}^n_+$.

Finding a face is an SDP!

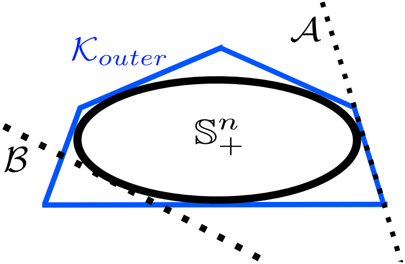
Our approach: simplify search by approximating $\mathbb{S}^n_+$.

- Using a user-specified *outer approximation* $\mathcal{K}^*_{outer}$,



Our approach: simplify search by approximating $\mathbb{S}^n_+$.

- Using a user-specified *outer approximation* $\mathcal{K}^*_{outer}$,

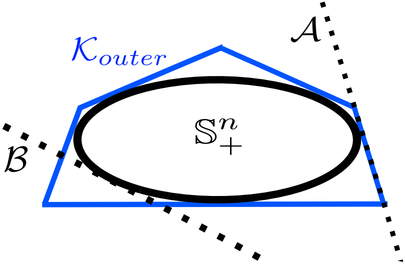


we find a face by solving easier optimization problem—e.g. an LP or SOCP:

$$\begin{array}{ll} \text{Find} & S \in \cancel{(\mathbb{S}^n_+)^*}\ \mathcal{K}^*_{outer} \\ \text{subject to} & S^\perp \text{ contains } \mathcal{A} \end{array}$$

Our approach: simplify search by approximating $\mathbb{S}^n_+$.

- Using a user-specified *outer approximation* $\mathcal{K}^*_{outer}$,



we find a face by solving easier optimization problem—e.g. an LP or SOCP:

Find	$S \in$ ~~$(\mathbb{S}^n_+)^*$~~ $\mathcal{K}^*_{outer}$
subject to	$S^\perp$ contains $\mathcal{A}$

- Since $\mathcal{K}^*_{outer} \subseteq (\mathbb{S}^n_+)^*$, the set $\mathbb{S}^n_+ \cap S^\perp$ is a face of $\mathbb{S}^n_+$.

Interpretation: for polyhedral approximations, we find a face by identifying always-active constraints.

- Polyhedral $\mathcal{K}_{outer}$ yields LP relaxation of SDP:

$$\begin{array}{lll} \text{minimize} & C \cdot X & \\ \text{subject to} & A_i \cdot X = b_i & i.e. \;\; X \in \mathcal{A} \\ & \cancel{X \in \mathbb{S}^n_+} & \\ & v_j^T X v_j \geq 0 \;\; \forall j \in \mathcal{I}, & i.e. \;\; X \in \mathcal{K}_{outer} \end{array}$$

Interpretation: for polyhedral approximations, we find a face by identifying always-active constraints.

- Polyhedral $\mathcal{K}_{outer}$ yields LP relaxation of SDP:

$$
\begin{array}{lll}
\text{minimize} & C \cdot X & \\
\text{subject to} & A_i \cdot X = b_i & i.e.\ \ X \in \mathcal{A} \\
& \cancel{X \in \mathbb{S}^n_+} & \\
& v_j^T X v_j \geq 0 \ \ \forall j \in \mathcal{I}, & i.e.\ \ X \in \mathcal{K}_{outer}
\end{array}
$$

- In this LP, some inequalities are *always active*:

Interpretation: for polyhedral approximations, we find a face by identifying always-active constraints.

- Polyhedral $\mathcal{K}_{outer}$ yields LP relaxation of SDP:

$$
\begin{array}{lll}
\text{minimize} & C \cdot X & \\
\text{subject to} & A_i \cdot X = b_i & i.e. \;\; X \in \mathcal{A} \\
 & \cancel{X \in \mathbb{S}^n_+} & \\
 & v_j^T X v_j \geq 0 \;\; \forall j \in \mathcal{I}, & i.e. \;\; X \in \mathcal{K}_{outer}
\end{array}
$$

- In this LP, some inequalities are *always active*:

$$\mathcal{A} \cap \mathcal{K}_{outer} \subseteq \left\{ X : v_k^T X v_k = 0 \;\; \forall k \in \mathcal{I}_{act} \subseteq \mathcal{I} \right\}$$

Interpretation: for polyhedral approximations, we find a face by identifying always-active constraints.

- Polyhedral $\mathcal{K}_{outer}$ yields LP relaxation of SDP:

$$\begin{array}{lll} \text{minimize} & C \cdot X & \\ \text{subject to} & A_i \cdot X = b_i & i.e.\ X \in \mathcal{A} \\ & \cancel{X \in \mathbb{S}^n_+} & \\ & v_j^T X v_j \geq 0 \ \ \forall j \in \mathcal{I}, & i.e.\ X \in \mathcal{K}_{outer} \end{array}$$

- In this LP, some inequalities are *always active*:

$$\mathcal{A} \cap \mathcal{K}_{outer} \subseteq \left\{ X : v_k^T X v_k = 0 \ \ \forall k \in \mathcal{I}_{act} \subseteq \mathcal{I} \right\}$$

- These inequalities identify a face of $\mathbb{S}^n_+$

$$\mathcal{A} \cap \mathbb{S}^n_+ \ \subseteq \ \mathbb{S}^n_+ \cap (\textstyle\sum_{k \in \mathcal{I}_{act}} v_k v_k^T)^{\perp}$$

Example choices for PSD approximation.

Choices for $\mathcal{K}_{outer}$ (in terms of its dual cone $\mathcal{K}^*_{outer}$):

$\mathcal{K}^*_{outer}$	Search	Size
Non-negative diagonal	LP	$O(n)$
Diagonally-dominant	LP	$O(n^2)$
Scaled diagonally-dominant	SOCP	$O(n^2)$
Factor width-k	SDP ($k \times k$)	$O(\binom{n}{k})$

Can choose $\mathcal{K}_{outer}$ to

- set pre-processing effort,
- enable use of *exact arithmetic*,
- ensure reformulation preserves sparsity.

Sparsity of reformulation is sensitive to chosen approximation.

To reformulate the SDP over $\mathbb{S}^n_+ \cap S^\perp$, one applies $U^T(\cdot)U$ to problem data, where range $U = \text{null}\, S$:

$$\begin{array}{ll} \text{minimize} & U^T C U \cdot X \\ \text{subject to} & U^T A_i U \cdot \hat{X} = b_i \\ & X \in \mathbb{S}^d_+ \end{array}$$

Sparsity of reformulation is sensitive to chosen approximation.

To reformulate the SDP over $\mathbb{S}^n_+ \cap S^\perp$, one applies $U^T(\cdot)U$ to problem data, where range $U = \text{null}\, S$:

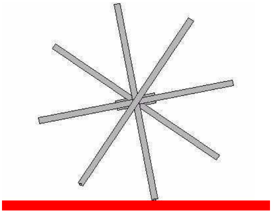
$$\begin{array}{ll} \text{minimize} & U^T C U \cdot X \\ \text{subject to} & U^T A_i U \cdot \hat{X} = b_i \\ & X \in \mathbb{S}^d_+ \end{array}$$

For $S \in \mathcal{K}^*_{outer}$,

$\mathcal{K}^*_{outer}$	$U^T(\cdot)U$
Non-negative diagonal	deletes rows/cols
Diagonally-dominant (rank one)	replaces two rows/cols with their sum/difference
Scaled diagonally-dominant (rank one)	replaces two rows/cols with a linear combination

Example #1 - SDP from Posa, Tedrake '13.

- Lyapunov analysis of *rimless wheel*, a simple walking model and hybrid system.



- Problem has 13000 variables and takes 105s to solve. With reductions...

$\mathcal{K}^*_{outer}$	Num. Vars.	Find Face (sec.)	Solve (sec.)
Diagonal	4500	.1	3.70
Diag. Dom.,	2300	.5	1.1

Example #2 - SDPs from Boyd, Mueller, et al. '12.

- SDP-based lower bounds of 4 optimal controllers.

	Before	After	Find face
1	$\mathbb{S}_+^{90} \times 100$	$\mathbb{S}_+^{60} \times 100$	3 sec
2	$\mathbb{S}_+^{120} \times 100$	$\mathbb{S}_+^{60} \times 100$	4 sec
3	$\mathbb{S}_+^{120} \times 100$	$\mathbb{S}_+^{60} \times 100$	5 sec
4	$\mathbb{S}_+^{150} \times 100$	$\mathbb{S}_+^{60} \times 100$	7 sec

Example #2 - SDPs from Boyd, Mueller, et al. ’12.

- SDP-based lower bounds of 4 optimal controllers.

	Before	After	Find face
1	$\mathbb{S}^{90}_{+} \times 100$	$\mathbb{S}^{60}_{+} \times 100$	3 sec
2	$\mathbb{S}^{120}_{+} \times 100$	$\mathbb{S}^{60}_{+} \times 100$	4 sec
3	$\mathbb{S}^{120}_{+} \times 100$	$\mathbb{S}^{60}_{+} \times 100$	5 sec
4	$\mathbb{S}^{150}_{+} \times 100$	$\mathbb{S}^{60}_{+} \times 100$	7 sec

- Solve times (sec)

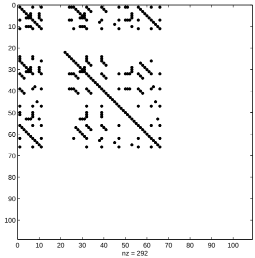
	Before (SeDuMi)	After (SeDuMi)	Before (Mosek)	After (Mosek)
1	949	727	246	158
2	795	593	281	151
3	617	507	230	189
4	1270	648	234	170

$\mathcal{K}^*_{outer}$ is set of non-negative diagonal matrices.

Simple approximations identify "trivial degeneracy"—this is the job of a pre-processor.

- In previous examples, strict feasibility failed for "trivial" reason.

$$\underbrace{\begin{pmatrix} 3x_1 & * & \cdots \\ * & -x_1 & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix} \in \mathbb{S}^n_+}_{\text{example constraint}}$$



imposed sparsity

- Identifying this structure is "due diligence"—analogous to removing columns of zeros from $Ax = b$.

Facial reduction also improves solution accuracy.

Considering the following SDP:

$$\text{Find } x_{ij} \text{ s.t. } \begin{pmatrix} 100 & x_{12} & x_{13} & x_{14} & x_{15} \\ x_{12} & x_{22} & x_{23} & x_{24} & x_{25} \\ x_{13} & x_{23} & x_{33} & x_{34} & x_{35} \\ x_{14} & x_{24} & x_{34} & x_{44} & x_{45} \\ x_{15} & x_{25} & x_{35} & x_{45} & x_{55} \end{pmatrix} \in \mathbb{S}^5_+$$

$$\sum \text{red} = 0$$
$$\sum \text{cyan} = 0$$
$$\sum \text{blue} = 0$$
$$\sum \text{mag.} = 0$$

Facial reduction also improves solution accuracy.

Considering the following SDP:

$$\text{Find } x_{ij} \text{ s.t. } \begin{pmatrix} 100 & x_{12} & \color{blue}{x_{13}} & x_{14} & \color{cyan}{x_{15}} \\ x_{12} & \color{red}{x_{22}} & x_{23} & x_{24} & \color{cyan}{x_{25}} \\ \color{blue}{x_{13}} & x_{23} & \color{cyan}{x_{33}} & x_{34} & x_{35} \\ x_{14} & x_{24} & x_{34} & \color{blue}{x_{44}} & \color{magenta}{x_{45}} \\ x_{15} & \color{cyan}{x_{25}} & x_{35} & \color{magenta}{x_{45}} & \color{magenta}{x_{55}} \end{pmatrix} \in \mathbb{S}^5_+ \qquad \begin{array}{l} \sum \color{red}{\text{red}} = 0 \\ \sum \color{cyan}{\text{cyan}} = 0 \\ \sum \color{blue}{\text{blue}} = 0 \\ \sum \color{magenta}{\text{mag.}} = 0 \end{array}$$

- It has a unique solution:

$$X^* = \begin{pmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Facial reduction also improves solution accuracy.

Considering the following SDP:

$$\text{Find } x_{ij} \text{ s.t. } \begin{pmatrix} 100 & x_{12} & x_{13} & x_{14} & x_{15} \\ x_{12} & x_{22} & x_{23} & x_{24} & x_{25} \\ x_{13} & x_{23} & x_{33} & x_{34} & x_{35} \\ x_{14} & x_{24} & x_{34} & x_{44} & x_{45} \\ x_{15} & x_{25} & x_{35} & x_{45} & x_{55} \end{pmatrix} \in \mathbb{S}^5_+ \qquad \begin{array}{l} \sum \text{red} = 0 \\ \sum \text{cyan} = 0 \\ \sum \text{blue} = 0 \\ \sum \text{mag.} = 0 \end{array}$$

- It has a unique solution:

$$X^* = \begin{pmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

- Facial reduction converts to a 1×1 SDP, easily solved.

Without facial reduction, error is large even though primal residual is small.

Solution found by solver (no reductions):

$$X = \begin{pmatrix} 100.000 & -0.0000 & -0.0585 & -0.0000 & 0.0000 \\ -0.0000 & 0.0000 & 0.0000 & 0.0000 & -0.0000 \\ -0.0585 & 0.0000 & 0.0001 & 0.0000 & -0.0000 \\ -0.0000 & 0.0000 & 0.0000 & 0.1171 & -0.1916 \\ 0.0000 & -0.0000 & -0.0000 & -0.1916 & 0.3832 \end{pmatrix}$$

$A(X) = b$:

$$\sum \text{red} = 0$$
$$\sum \text{cyan} = 0$$
$$\sum \text{blue} = 0$$
$$\sum \text{mag.} = 0$$

Without facial reduction, error is large even though primal residual is small.

Solution found by solver (no reductions):

$$X = \begin{pmatrix} 100.000 & -0.0000 & -0.0585 & -0.0000 & 0.0000 \\ -0.0000 & 0.0000 & 0.0000 & 0.0000 & -0.0000 \\ -0.0585 & 0.0000 & 0.0001 & 0.0000 & -0.0000 \\ -0.0000 & 0.0000 & 0.0000 & 0.1171 & -0.1916 \\ 0.0000 & -0.0000 & -0.0000 & -0.1916 & 0.3832 \end{pmatrix}$$

$A(X) = b$:

$$\sum \text{red} = 0$$
$$\sum \text{cyan} = 0$$
$$\sum \text{blue} = 0$$
$$\sum \text{mag.} = 0$$

Residuals:

$$\begin{aligned} \|A(X) - b\| &= 4.54 \cdot 10^{-9} \\ \lambda_{min}(X) &= 2.98 \cdot 10^{-10} \end{aligned}$$

Without facial reduction, error is large even though primal residual is small.

Solution found by solver (no reductions):

$$X = \begin{pmatrix} 100.000 & -0.0000 & -0.0585 & -0.0000 & 0.0000 \\ -0.0000 & 0.0000 & 0.0000 & 0.0000 & -0.0000 \\ -0.0585 & 0.0000 & 0.0001 & 0.0000 & -0.0000 \\ -0.0000 & 0.0000 & 0.0000 & 0.1171 & -0.1916 \\ 0.0000 & -0.0000 & -0.0000 & -0.1916 & 0.3832 \end{pmatrix}$$

$A(X) = b$:

$$\sum \text{red} = 0$$
$$\sum \text{cyan} = 0$$
$$\sum \text{blue} = 0$$
$$\sum \text{mag.} = 0$$

Residuals:

$$\begin{aligned} ||A(X) - b|| &= 4.54 \cdot 10^{-9} \\ \lambda_{min}(X) &= 2.98 \cdot 10^{-10} \end{aligned}$$

True error:

$$||X - X^*|| = \mathbf{0.4907}$$

Without facial reduction, residuals can be (arbitrarily) small even if problem is infeasible.

Solution found by solver for perturbed, **infeasible** problem:

$$X = \begin{pmatrix} 100.000 & -0.0000 & -0.3044 & -0.0000 & 0.0000 \\ -0.0000 & 0.0000 & 0.0000 & 0.0004 & -0.0005 \\ -0.3044 & 0.0000 & 0.0010 & 0.0000 & -0.0000 \\ -0.0000 & 0.0004 & 0.0000 & 0.6088 & -0.6963 \\ 0.0000 & -0.0005 & -0.0000 & -0.6963 & 0.8926 \end{pmatrix}$$

$A(X) = b$:

Residuals:

$$\sum \text{red} = 0$$

$$\sum \text{cyan} = 0$$

$$\sum \text{blue} = 0$$

$$\underbrace{\sum \text{mag.} = -.5}_{\text{perturbed}}$$

Without facial reduction, residuals can be (arbitrarily) small even if problem is infeasible.

Solution found by solver for perturbed, **infeasible** problem:

$$X = \begin{pmatrix} 100.000 & -0.0000 & -0.3044 & -0.0000 & 0.0000 \\ -0.0000 & 0.0000 & 0.0000 & 0.0004 & -0.0005 \\ -0.3044 & 0.0000 & 0.0010 & 0.0000 & -0.0000 \\ -0.0000 & 0.0004 & 0.0000 & 0.6088 & -0.6963 \\ 0.0000 & -0.0005 & -0.0000 & -0.6963 & 0.8926 \end{pmatrix}$$

$A(X) = b$:

$$\sum \text{red} = 0$$
$$\sum \text{cyan} = 0$$
$$\sum \text{blue} = 0$$
$$\underbrace{\sum \text{mag.} = -.5}_{\text{perturbed}}$$

Residuals:

$$\begin{aligned} \|A(X) - b\| &= 7.54 \cdot 10^{-7} \\ \lambda_{min}(X) &= 4.82 \cdot 10^{-8} \end{aligned}$$

Without facial reduction, residuals can be (arbitrarily) small even if problem is infeasible.

Solution found by solver for perturbed, **infeasible** problem:

$$X = \begin{pmatrix} 100.000 & -0.0000 & -0.3044 & -0.0000 & 0.0000 \\ -0.0000 & 0.0000 & 0.0000 & 0.0004 & -0.0005 \\ -0.3044 & 0.0000 & 0.0010 & 0.0000 & -0.0000 \\ -0.0000 & 0.0004 & 0.0000 & 0.6088 & -0.6963 \\ 0.0000 & -0.0005 & -0.0000 & -0.6963 & 0.8926 \end{pmatrix}$$

$A(X) = b$:

$$\sum \text{red} = 0$$
$$\sum \text{cyan} = 0$$
$$\sum \text{blue} = 0$$
$$\underbrace{\sum \text{mag.} = -.5}_{\text{perturbed}}$$

Residuals:

$$\begin{aligned} \|A(X) - b\| &= 7.54 \cdot 10^{-7} \\ \lambda_{min}(X) &= 4.82 \cdot 10^{-8} \end{aligned}$$

True error:

$$\|X - X^*\| = \text{undefined}$$

Part II: Dual solution recovery.

Facial reduction *restricts* the primal and *relaxes* the dual:

$$\begin{array}{ll} \text{minimize} & C \cdot X \\ \text{subject to} & A_i \cdot X = b_i \\ & \cancel{X \in \mathbb{S}^n_+} \\ & X \in \mathbb{S}^n_+ \cap S^\perp \end{array} \qquad \begin{array}{ll} \text{maximize} & b^T y \\ \text{subject to} & \cancel{C - \sum_i y_i A_i \in \mathbb{S}^n_+} \\ & C - \sum_i y_i A_i \in \overline{\mathbb{S}^n_+ + \operatorname{span} S} \end{array}$$

Part II: Dual solution recovery.

Facial reduction *restricts* the primal and *relaxes* the dual:

$$\begin{array}{ll} \text{minimize} & C \cdot X \\ \text{subject to} & A_i \cdot X = b_i \\ & \cancel{X \in \mathbb{S}^n_+} \\ & X \in \mathbb{S}^n_+ \cap S^\perp \end{array} \qquad \begin{array}{ll} \text{maximize} & b^T y \\ \text{subject to} & \cancel{C - \sum_i y_i A_i \in \mathbb{S}^n_+} \\ & C - \sum_i y_i A_i \in \overline{\mathbb{S}^n_+ + \text{span } S} \end{array}$$

Solution recovery: (using fact $S = \sum_i d_i A_i, b^T d = 0$):

$$\boxed{\text{Find } \alpha \text{ such that } C - \sum_i y_i A_i + \alpha S \in \mathbb{S}^n_+}$$

Part II: Dual solution recovery.

Facial reduction *restricts* the primal and *relaxes* the dual:

minimize	$C \cdot X$	maximize	$b^T y$
subject to	$A_i \cdot X = b_i$	subject to	~~$C - \sum_i y_i A_i \in \mathbb{S}^n_+$~~
	~~$X \in \mathbb{S}^n_+$~~		$C - \sum_i y_i A_i \in \overline{\mathbb{S}^n_+ + \text{span } S}$
	$X \in \mathbb{S}^n_+ \cap S^\perp$		

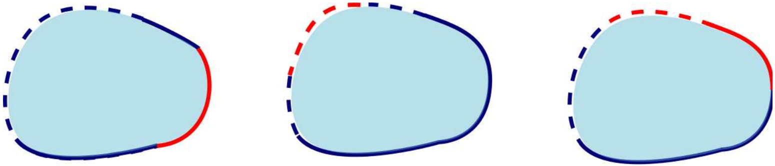
Solution recovery: (using fact $S = \sum_i d_i A_i, b^T d = 0$):

$$\boxed{\text{Find } \alpha \text{ such that } C - \sum_i y_i A_i + \alpha S \in \mathbb{S}^n_+}$$

Is dual solution recovery possible? Equivalent to asking:

$$\boxed{\text{Is } C - \sum_i y_i A_i \text{ in } \mathbb{S}^n_+ + \text{span } S?}$$

Is dual recovery possible? There are three possibilities.



Yes.

No (duality gap).

Maybe.

The set $\mathbb{S}^n_+ + \text{span}\, S$ and set of optimal slacks, $C - \sum_i y_i A_i$.

Can determine if recovery will succeed by comparing nullspaces.

Pick orthogonal (U, V) satisfying range $V =$ range S and change coordinates:

$$\begin{pmatrix} W_{11} & W_{21}^T \\ W_{21} & W_{22} \end{pmatrix} := (U, V)^T (C - \sum_i y_i A_i)(U, V)$$

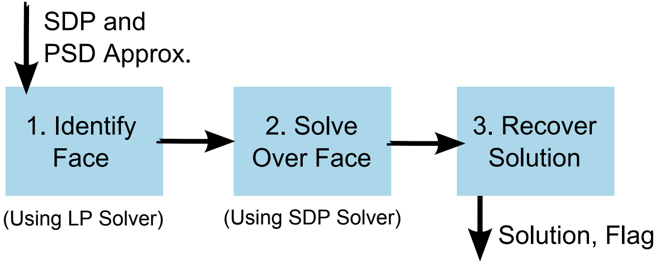
The following holds:

$$C - \textstyle\sum_i y_i A_i \in \overline{\mathbb{S}^n_+ + \text{span}\, S} \quad \Leftrightarrow \quad W_{11} \in \mathbb{S}^d_+$$

$$C - \textstyle\sum_i y_i A_i \in \mathbb{S}^n_+ + \text{span}\, S \quad \Leftrightarrow \quad W_{11} \in \mathbb{S}^d_+, \ \underbrace{\text{null}\, W_{11} \subseteq \text{null}\, W_{21}}_{\text{Recovery succeeds.}}$$

Part III: `frlib` is a MATLAB-based tool implementing these ideas.

Basic flow:



Inputs:

1. SDP primal-dual pair
2. PSD approximation (e.g non-negative diagonal matrices)

Outputs:

1. Solution to primal-dual pair
2. Flag indicating successful dual recovery

Using `frlib` (direct interface).

Calling directly using diagonal ('d') approximations:

```
prg = frlibPrg(A,b,c,K);
prgR = prg.ReducePrimal('d');
[xR,yR] = sedumi(prgR.A, prgR.b, ...
                 prgR.c, prgR.K);
[x,y,dual_recovered] = prgR.Recover(xR,yR);
```

What do these functions do?

- `frlibPrg`: reads in SDP in SeDuMi format `A b c K`
- `ReducePrimal`: finds a face by solving LPs.
- `Recover`: converts to original coordinates, attempts dual recovery.

Using `frlib` via YALMIP—a parser by Johan Löfberg:

To use, specify as solver and set options in YALMIP script:

```
sdpvar x y z
p = 12+y^2-2*x^3*y+2*y*z^2+x^6-2*x^3*z^2+z^4...
      +x^2*y^2;
ops = sdpsettings('solver','frlib');
ops = sdpsettings(ops,'frlib.approx','dd');
[sol] = solvesos(sos(p),[],ops);
```

Using `frlib` via YALMIP—a parser by Johan Löfberg:

To use, specify as solver and set options in YALMIP script:

```
sdpvar x y z
p = 12+y^2-2*x^3*y+2*y*z^2+x^6-2*x^3*z^2+z^4...
      +x^2*y^2;
ops = sdpsettings('solver','frlib');
ops = sdpsettings(ops,'frlib.approx','dd');
[sol] = solvesos(sos(p),[],ops);
```

Produces output:

```
---------------------------------
frlib: reductions found!
---------------------------------
  Dim PSD constraint(s) (original):  7 2
  Dim PSD constraint(s) (reduced):   3 0
```

Summary

- Facial reduction-based pre-processing allowing you to specify pre-processing effort.
- Dual solution recovery: not always possible!
- Software/paper:

`www.github.com/frankpermenter/frlib`

`http://arxiv.org/abs/1408.4685`