Solving linear and non-linear SDP by PENNON

Michal Kočvara

School of Mathematics, The University of Birmingham

University of Warwick, February 2010

Outline

- Why nonlinear SDP?
- PENNON – the new generation
- Example: Nearest correlation matrix
- Example: Approximation by nonnegative splines
- Example: Sensor network localization

Semidefinite programming (SDP)

“generalized” mathematical program

$$\min f(x)$$
$$\text{subject to}$$
$$g(x) \geq 0$$
$$\mathcal{A}(x) \succeq 0$$

$\mathcal{A}(x)$ — (non)linear matrix operator $\mathbb{R}^n \to \mathbb{S}^m$
$(\mathcal{A}(x) = A_0 + \sum x_i A_i)$

SDP notations

$\mathbb{S}^n$ ...symmetric matrices of order $n \times n$
$A \succeq 0$ ...A positive semidefinite
$A \succeq 0$ ...$A - B \succeq 0$
$\langle A, B \rangle := \mathrm{Tr}(AB)$...inner product on $\mathbb{S}^n$
$\mathcal{A}[\mathbb{R}^n \to \mathbb{S}^m]$...linear matrix operator defined by

$$\mathcal{A}(y) := \sum_{i=1}^{n} y_i A_i \qquad \text{with } A_i \in \mathbb{S}^m$$

$\mathcal{A}^*[\mathbb{S}^m \to \mathbb{R}^n]$...adjoint operator defined by

$$\mathcal{A}^*(X) := [\langle A_1, X \rangle, \ldots, \langle A_n, X \rangle]^T$$

and satisfying

$$\langle \mathcal{A}^*(X), y \rangle = \langle \mathcal{A}(y), X \rangle \qquad \text{for all } y \in \mathbb{R}^n$$

Primal-dual SDP pair

$$\inf_X \langle C, X\rangle := \mathrm{Tr}(CX) \qquad \text{(P)}$$
$$\text{s.t. } \mathcal{A}^*(X) = b \qquad [\langle A_i, X\rangle = b_i,\ i = 1, \ldots, n]$$
$$X \succeq 0$$

$$\sup_{y,S} \langle b, y\rangle := \sum b_i y_i \qquad \text{(D)}$$
$$\text{s.t. } \mathcal{A}(y) + S = C \qquad [\sum y_i A_i + S = C]$$
$$S \succeq 0$$

Weak duality: Feasible X, y, S satisfy

$$\langle C, X\rangle - \langle b, y\rangle = \langle \mathcal{A}(y) + S, X\rangle - \sum y_i \langle A_i, X\rangle = \langle S, X\rangle \geq 0$$

duality gap nonnegative for feasible points

Linear Semidefinite Programming

Vast area of applications. . .

- LP and CQP is SDP
- eigenvalue optimisation
- robust programming
- control theory
- relaxations of integer optimisation problems
- approximations to combinatorial optimisation problems
- structural optimisation
- chemical engineering
- machine learning
- many many others. . .

Why nonlinear SDP?

Problems from

- Structural optimization
- Control theory
- Mathematical Programming with Equilibrium Constraints
- Examples below

There are more but the researchers just don't know about. . .

Nonlinear SDP?

The general nonlinear SDP (NSDP) problem

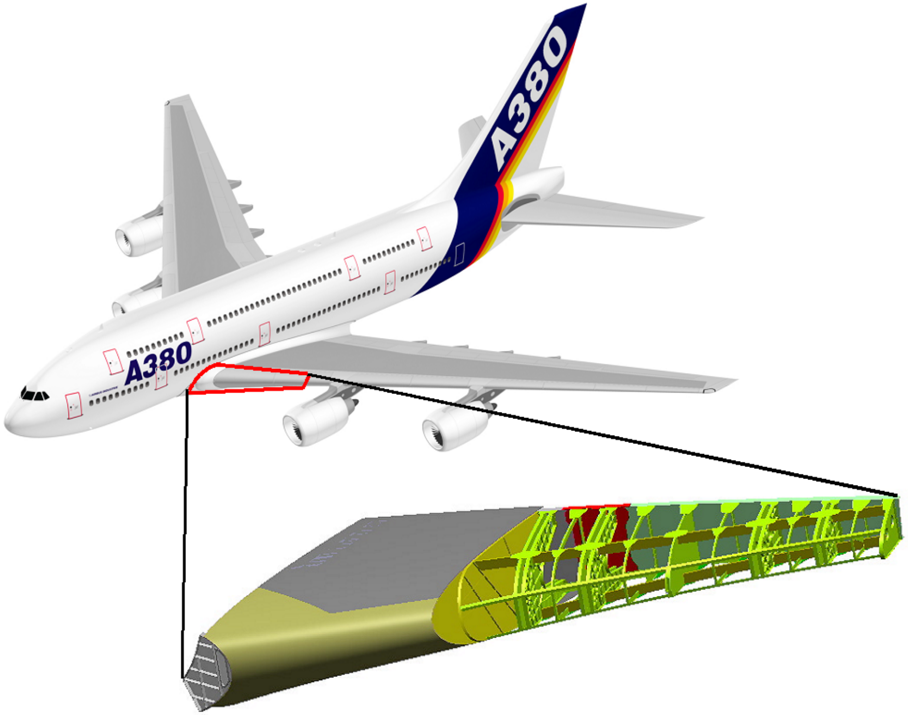
$$\min_{x \in \mathbb{R}^n} f(x)$$

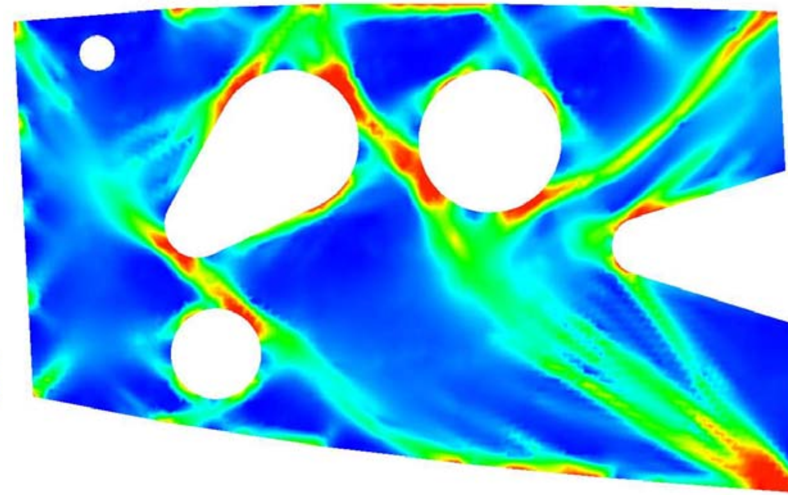
subject to

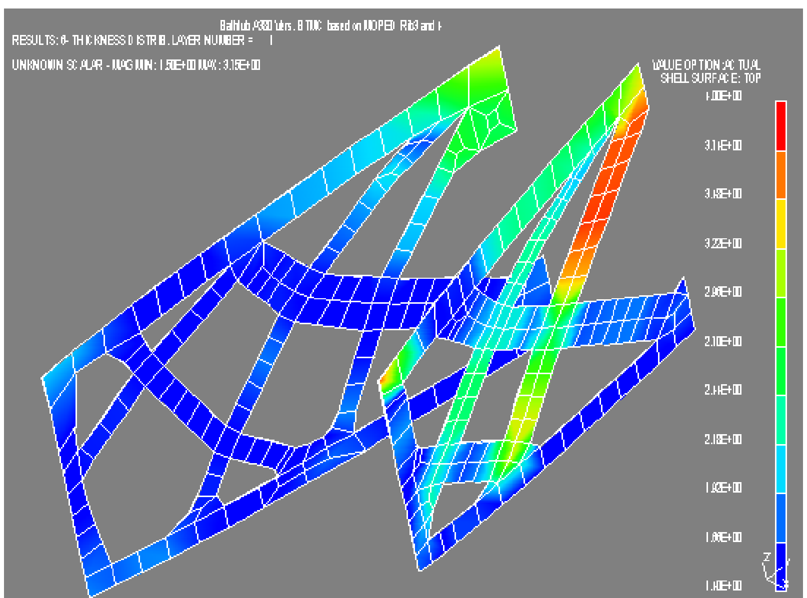
$$g_i(x) \leq 0, \qquad i = 1, \ldots, n_g$$
$$h_i(x) = 0, \qquad i = 1, \ldots, n_h$$
$$\mathcal{A}(x) \preccurlyeq 0$$

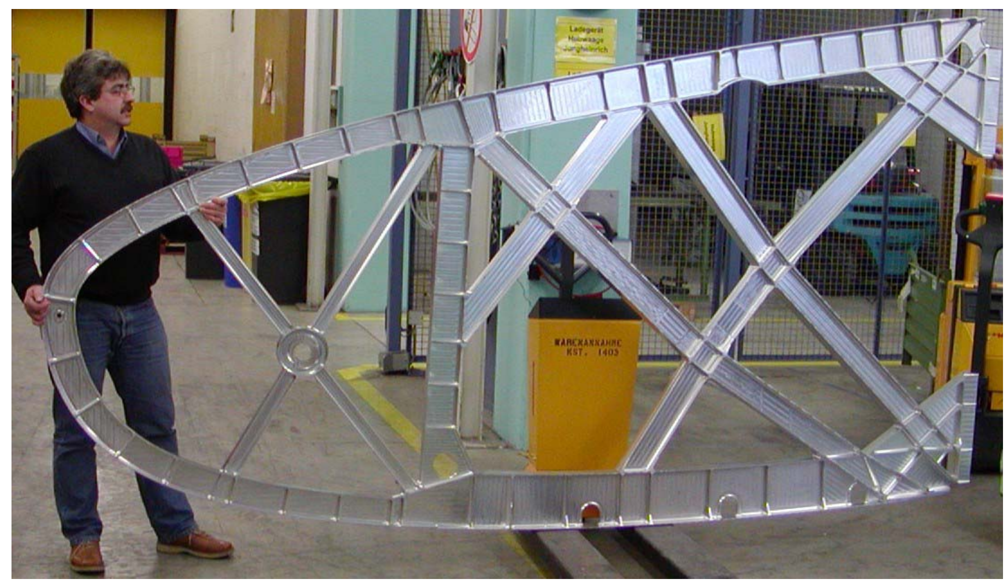
$b \in \mathbb{R}^n$ and $\mathcal{A} : \mathbb{R}^n \to \mathbb{S}^m$ nonlinear, nonconvex


A380
A380 AIRBUS
SINGAPORE AIRLINES


A380
A380



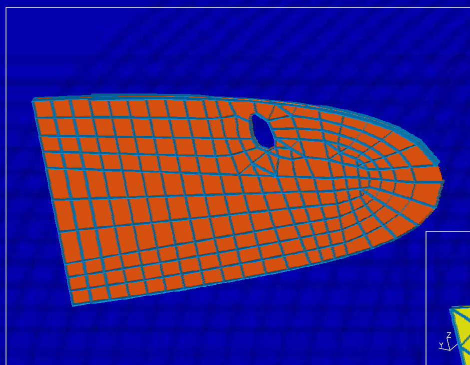

RESULTS: 6- THICKNESS DISTRIB. LAYER NUMBER = 1
VALUE OPTION:ACTUAL
SHELL SURFACE: TOP



A380 Inboard Inner Leading Edge Ribs

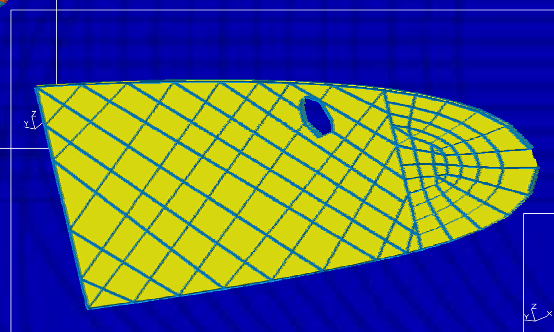
Impact of topological decisions

all designs
sizing-optimized



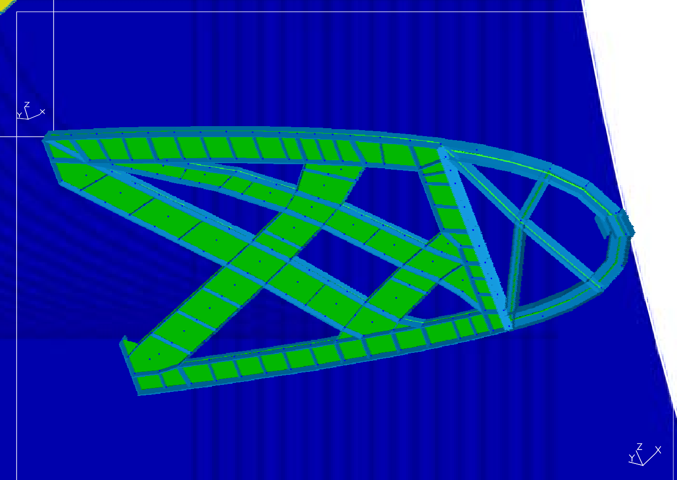
Baseline

-13%



Stiffeners at 45 °

-33%



Optimized topology

Free Material Optimization

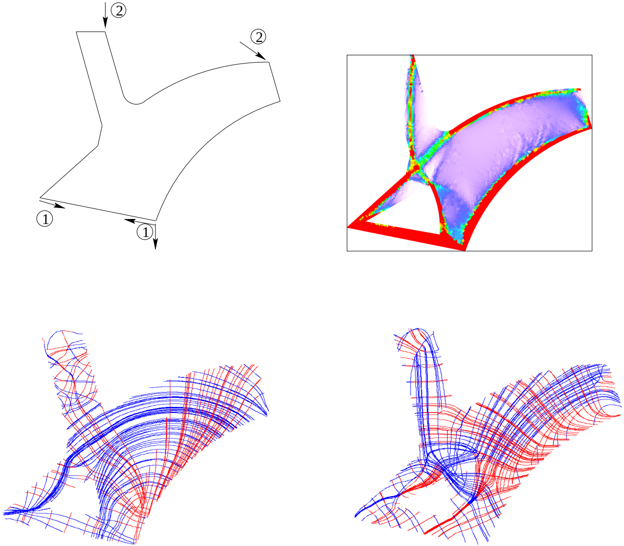
Aim:

Given an amount of material, boundary conditions and external load f, find the material (distribution) so that the body is as stiff as possible under f.

The design variables are the material properties at each point of the structure.

M. P. Bendsøe, J.M. Guades, R.B. Haber, P. Pedersen and J. E. Taylor: An analytical model to predict optimal material properties in the context of optimal structural design. *J. Applied Mechanics*, 61 (1994) 930–937

Free Material Optimization



FMO SL primal formulation

FMO-problem (minimum volume formulation)

$$\min_{u \in \mathbb{R}^n, E_1, \dots, E_m} \sum_{i=1}^{m} \mathrm{Tr} E_i$$

subject to

$$E_i \succeq 0, \underline{\rho} \leq \mathrm{Tr} E_i \leq \overline{\rho}, \quad i = 1, \dots, m$$

$$f^\top u \leq C$$

$$A(E) u = f$$

FMO SL primal formulation

FMO-problem with vibration/buckling constraint

$$\min_{u \in \mathbb{R}^n, E_1, \ldots, E_m} \sum_{i=1}^{m} \mathrm{Tr} E_i$$

subject to

$$E_i \succeq 0, \ \underline{\rho} \leq \mathrm{Tr} E_i \leq \overline{\rho}, \quad i = 1, \ldots, m$$

$$f^\top u \leq C$$

$$A(E)u = f$$

$$A(E) + G(E, u) \succeq 0$$

FMO SL primal formulation

FMO-problem with vibration/buckling constraint

$$\min_{u \in \mathbb{R}^n, E_1, \ldots, E_m} \sum_{i=1}^{m} \mathrm{Tr} E_i$$

subject to

$$E_i \succeq 0, \; \underline{\rho} \leq \mathrm{Tr} E_i \leq \overline{\rho}, \quad i = 1, \ldots, m$$

$$f^\top u \leq C$$

$$A(E)u = f$$

$$A(E) + G(E, u) \succeq 0$$

…nonlinear, non-convex semidefinite problem

PENNON collection

PENNON (PENalty methods for NONlinear optimization)
a collection of codes for NLP, (linear) SDP and BMI

– one algorithm to rule them all –

READY

- PENNLP AMPL, MATLAB, C/Fortran
- PENSDP MATLAB/YALMIP, SDPA, C/Fortran
- PENBMI MATLAB/YALMIP, C/Fortran

NEW

- PENNON (NLP + SDP) extended AMPL, MATLAB

The problem

Optimization problems with nonlinear objective subject to nonlinear inequality and equality constraints and semidefinite bound constraints:

$$\begin{aligned} \min_{x\in ,Y_1\in\mathbb{S}^{p_1},\ldots,Y_k\in\mathbb{S}^{p_k}} \quad & f(x,Y) \\ \text{subject to} \quad & g_i(x,Y)\le 0, && i=1,\ldots,m_g \\ & h_i(x,Y)=0, && i=1,\ldots,m_h \qquad \text{(NLP-SDP)} \\ & \underline{\lambda}_i I \preceq Y_i \preceq \overline{\lambda}_i I, && i=1,\ldots,k\,. \end{aligned}$$

The problem

Here

- $x \in \mathbb{R}^n$ is the vector variable
- $Y_1 \in \mathbb{S}^{p_1}, \ldots, Y_k \in \mathbb{S}^{p_k}$ are the matrix variables, k symmetric matrices of dimensions $p_1 \times p_1, \ldots, p_k \times p_k$
- we denote $Y = (Y_1, \ldots, Y_k)$
- f, g_i and h_i are C^2 functions from $\mathbb{R}^n \times \mathbb{S}^{p_1} \times \ldots \times \mathbb{S}^{p_k}$ to $\mathbb{R}$
- $\underline{\lambda}_i$ and $\overline{\lambda}_i$ are the lower and upper bounds, respectively, on the eigenvalues of Y_i, $i = 1, \ldots, k$

The problem

Any nonlinear SDP problem can be furmulated as NLP-SDP, using slack variables and (NLP) equality constraints:

$$g(X) \succeq 0$$

write as

$$\begin{aligned} &g(X) = S \quad \text{element-wise} \\ &S \succeq 0 \end{aligned}$$

The algorithm

Based on penalty/barrier functions $\varphi_g : \mathbb{R} \to \mathbb{R}$ and $\Phi_P : \mathbb{S}^p \to \mathbb{S}^p$:

$$g_i(x) \leq 0 \iff p_i \varphi_g(g_i(x)/p_i) \leq 0, \quad i = 1, \ldots, m$$
$$Z \preceq 0 \iff \Phi_P(Z) \preceq 0, \quad Z \in \mathbb{S}^p .$$

Augmented Lagrangian of (NLP-SDP):

$$F(x,Y,u,\underline{U},\overline{U},p) = f(x,Y) + \sum_{i=1}^{m_g} u_i p_i \varphi_g(g_i(x,Y)/p_i) + \sum_{i=1}^{k} \langle \underline{U}_i, \Phi_P(\underline{\lambda}_i I - Y_i) \rangle + \sum_{i=1}^{k} \langle \overline{U}_i, \Phi_P(Y_i - \overline{\lambda}_i I) \rangle ;$$

here $u \in \mathbb{R}^{m_g}$ and $\underline{U}_i, \overline{U}_i$ are Lagrange multipliers.

The algorithm

A generalized Augmented Lagrangian algorithm (based on R. Polyak '92, Ben-Tal–Zibulevsky '94, Stingl '05):

Given $x^1, Y^1, u^1, \underline{U}^1, \overline{U}^1; p_i^1 > 0,\ i = 1, \ldots, m_g$ and $P > 0$.
For $k = 1, 2, \ldots$ repeat till a stopping criterium is reached:

$$
\begin{array}{ll}
(i) & \text{Find } x^{k+1} \text{ and } Y^{k+1} \text{ s.t. } \|\nabla_x F(x^{k+1}, Y^{k+1}, u^k, \underline{U}^k, \overline{U}^k, p^k)\| \le K \\
(ii) & u_i^{k+1} = u_i^k \varphi_g'(g_i(x^{k+1})/p_i^k), \quad i = 1, \ldots, m_g \\
 & \underline{U}_i^{k+1} = D_{\mathcal{A}} \Phi_P((\underline{\lambda}_i I - Y_i); \underline{U}_i^k), \quad i = 1, \ldots, k \\
 & \overline{U}_i^{k+1} = D_{\mathcal{A}} \Phi_P((Y_i - \overline{\lambda}_i I); \overline{U}_i^k), \quad i = 1, \ldots, k \\
(iii) & p_i^{k+1} < p_i^k,\ i = 1, \ldots, m_g \\
 & P^{k+1} < P^k\ .
\end{array}
$$

Interfaces

How to enter the data – the functions and their derivatives?

- Matlab interface
- AMPL interface

Matlab interface

User provides six MATLAB functions:

`f` ...evaluates the objective function

`df` ...evaluates the gradient of objective function

`hf` ...evaluates the Hessian of objective function

`g` ...evaluates the constraints

`dg` ...evaluates the gradient of constraints

`hg` ...evaluates the Hessian of constraints

Matlab interface

Matrix variables are treated as vectors, using the function $\mathrm{svec} : \mathbb{S}^m \to \mathbb{R}^{(m+1)m/2}$ defined by

$$\mathrm{svec}\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ & a_{22} & \dots & a_{2m} \\ & & \ddots & \vdots \\ sym & & & a_{mm} \end{pmatrix} = (a_{11}, a_{12}, a_{22}, \dots, a_{1m}, a_{2m}, a_{mm})^T$$

Matlab interface

Matrix variables are treated as vectors, using the function $\text{svec} : \mathbb{S}^m \to \mathbb{R}^{(m+1)m/2}$ defined by

$$\text{svec} \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ & a_{22} & \dots & a_{2m} \\ & & \ddots & \vdots \\ sym & & & a_{mm} \end{pmatrix}$$

$$= (a_{11}, a_{12}, a_{22}, \dots, a_{1m}, a_{2m}, a_{mm})^T$$

Keep a specific order of variables, to recognize which are matrices and which vectors. Add lower/upper bounds on matrix eigenvalues.

Sparse matrices available, sparsity maintained in the user defined functions.

AMPL interface

AMPL does not support SDP variables and constraints. Use the same trick:
Matrix variables are treated as vectors, using the function $\text{svec} : \mathbb{S}^m \to \mathbb{R}^{(m+1)m/2}$ defined by

$$\text{svec}\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ & a_{22} & \dots & a_{2m} \\ & & \ddots & \vdots \\ sym & & & a_{mm} \end{pmatrix} = (a_{11}, a_{12}, a_{22}, \dots, a_{1m}, a_{2m}, a_{mm})^T$$

Need additional input file specifying the matrix sizes and lower/upper eigenvalue bounds.

Example: nearest correlation matrix

Find a nearest correlation matrix:

$$\min_X \sum_{i,j=1}^{n} (X_{ij} - H_{ij})^2 \tag{1}$$

subject to

$$X_{ii} = 1, \quad i = 1, \ldots, n$$

$$X \succeq 0$$

Example: nearest correlation matrix

AMPL code:

```
param h{1..21};
set ind within {1..21};

# Nonlinear SDP variables
var x{1..21} default 0;

minimize Obj: sum{i in 1..21} (x[i]-h[i])^2;

subject to
   l1{i in ind}:
x[i] = 1;

data;
param h:=
   1  1.0000  2 -0.4400  3  1.0000  4 -0.2000  5  0
   8 -0.3800  9 -0.1700 10  1.0000 11 -0.4600 12  0
```

Example: nearest correlation matrix

For

$$H_{\text{ext}} = \begin{pmatrix} 1 & -0.44 & -0.20 & 0.81 & -0.46 & -0.05 \\ -0.44 & 1 & 0.87 & -0.38 & 0.81 & -0.58 \\ -0.20 & .87 & 1 & -0.17 & 0.65 & -0.56 \\ 0.81 & -0.38 & -0.17 & 1 & -0.37 & -0.15 \\ -0.46 & 0.81 & 0.65 & -0.37 & 1 & -0.08 \\ -0.05 & -0.58 & -0.56 & -0.15 & 0.08 & 1 \end{pmatrix}$$

the eigenvalues of the correlation matrix are

```
eigen =
    0.0000 0.1163 0.2120 0.7827 1.7132 3.1757
```

Example: nearest correlation matrix

The condition number of the nearest correlation matrix must be bounded.
Add new variables $y, z \in \mathbb{R}$ and new cosntraints

$$X \succeq zI \tag{2}$$
$$X \preceq yI \tag{3}$$
$$y \leq \kappa z \tag{4}$$

where κ is the required condition number.
The constraints do not fit into our required NLP-SDP problem structure. Introduce two new (slack) matrix variables, say, P and Q, and replace (2) and (3) by

$$\begin{aligned} X - zI - P &= 0 \\ X - yI - Q &= 0 \\ P &\succeq 0 \\ Q &\preceq 0 \end{aligned}$$

Example: nearest correlation matrix

More ellegant way: rewrite constraints (2)–(3) as

$$I \preceq \widetilde{X} \preceq \kappa I \tag{5}$$

assuming that $y = \kappa z$ and using the transormation of the variable X:

$$z\widetilde{X} = X\,.$$

The new problem:

$$\min_{z,\widetilde{X}} \sum_{i,j=1}^{n} (z\widetilde{X}_{ij} - H_{ij})^2 \tag{6}$$

subject to

$$z\widetilde{X}_{ii} = 1, \quad i = 1, \ldots, n$$

$$I \preceq \widetilde{X} \preceq \kappa I$$

Example: nearest correlation matrix

For

```
X =
    1.0000 -0.3775 -0.2230  0.7098 -0.4272 -0.0704
   -0.3775  1.0000  0.6930 -0.3155  0.5998 -0.4218
   -0.2230  0.6930  1.0000 -0.1546  0.5523 -0.4914
    0.7098 -0.3155 -0.1546  1.0000 -0.3857 -0.1294
   -0.4272  0.5998  0.5523 -0.3857  1.0000 -0.0576
   -0.0704 -0.4218 -0.4914 -0.1294 -0.0576  1.0000
```

the eigenvalues of the correlation matrix are

```
eigen =
  0.2866   0.2866   0.2867   0.6717   1.6019   2.8664
```

Example: nearest correlation matrix

Cooperation with Allianz SE, Munich:

Matrices of size up to 3500×3500

Code PENCOR:
C code, data in xml format
feasibility analysis
sensitivity analysis w.r.t. bounds on matrix elements

Example: Approximation by nonnegative splines

Let $f : [0,1] \to \mathbb{R}$. Given its (noisy) function values b_i, $i = 1, \ldots, n$ at points $t_i \in (0,1)$.

Find a smooth approximation of f by a cubic spline:

$$P(t) = P^{(i)}(t) = \sum_{k=1}^{3} P(i)_k (t - a_{i-1})^k$$

for a point $t \in [a_{i-1}, a_i]$, where $0 = a_0 < a_1 < \ldots < a_m = 1$ are the knots and $P_k^{(i)} (i = 1, \ldots, m,\ k = 0, 1, 2, 3)$ the coefficients of the spline.

Spline property: for $i = 1, \ldots, m-1$

$$P_0^{(i+1)} - P_0^{(i)} - P_1^{(i)}(a_i - a_{i-1}) - P_2^{(i)}(a_i - a_{i-1})^2 - P_3^{(i)}(a_i - a_{i-1})^3 = 0 \quad (7)$$

$$P_1^{(i+1)} - P_1^{(i)} - 2P_2^{(i)}(a_i - a_{i-1}) - 3P_3^{(i)}(a_i - a_{i-1})^2 = 0 \quad (8)$$

$$2P_2^{(i+1)} - 2P_2^{(i)} - 6P_3^{(i)}(a_i - a_{i-1}) = 0 \,. \quad (9)$$

Example: Approximation by nonnegative splines

The function f will be approximated by P in the least square sense: minimize

$$\sum_{j=1}^{n}(P(t_j) - b_j)^2$$

subject to (7),(8),(9).

Now, f is assumed to be nonnegative, so $P \geq 0$ is required.

Example: Approximation by nonnegative splines

de Boor and Daniel '74: while approximation of a nonnegative function by nonnegative splines of order k gives errors of order h^k, approximation by a subclass of nonnegative splines of order k consisting of all those whose B-spline coefficients are nonnegative may yield only errors of order h^2.

Nesterov 2000: $P^{(i)}(t)$ nonnegative $\Leftrightarrow$ there exist two symmetric matrices

$$X^{(i)} = \begin{pmatrix} x_i & y_i \\ y_i & z_i \end{pmatrix}, \qquad S^{(i)} = \begin{pmatrix} s_i & v_i \\ v_i & w_i \end{pmatrix}$$

such that

$$P_0^{(i)} = (a_i - a_{i-1})s_i \tag{10}$$

$$P_1^{(i)} = x_i - s_i + 2(a_i - a_{i-1})v_i \tag{11}$$

$$P_2^{(i)} = 2y_i - 2v_i + (a_i - a_{i-1})w_i \tag{12}$$

$$P_3^{(i)} = z_i - w_i \tag{13}$$

$$X^{(i)} \succeq 0, \quad S^{(i)} \succeq 0\,. \tag{14}$$

Example: Approximation by nonnegative splines

We want to solve an NLP-SDP problem

$$\min_{\substack{P_k^{(i)} \in \mathbb{R} \\ i=1,\dots,m,\ k=0,1,2,3}} \sum_{j=1}^{n} (P(t_j) - b_j)^2 \tag{15}$$

subject to

$$(7), (8), (9), \quad i = 1, \dots, m$$

$$(10) - (14), \quad i = 1, \dots, m$$

Example: Approximation by nonnegative splines

Example, $n = 500, m = 7$, noisy data:

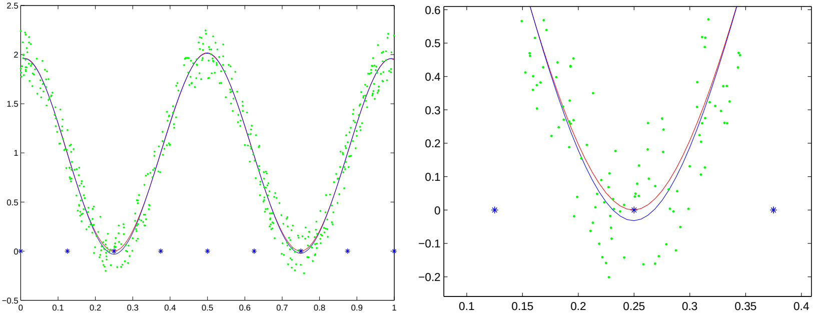


Figure: Approximation by nonnegative splines: noisy data given in green, optimal nonnegative spline in red and an optimal spline ignoring the nonnegativity constraint in blue. The right-hand side figure zooms on the left valey.

Sensor network localization

(Euclidean distance matrix completion, Graph realization)

We have (in $\mathbb{R}^2$ (or $\mathbb{R}^d$))

n points a_i, anchors with known location

m points x_i, sensors with unknown location

d_{ij} known Euclidean distance between "close" points

$$d_{ij} = \|x_i - x_j\|, \; (i,j) \in \mathcal{I}_x$$

$$\bar{d}_{kj} = \|a_k - x_j\|, \; (k,j) \in \mathcal{I}_a$$

Goal: Find the positions of the sensors!

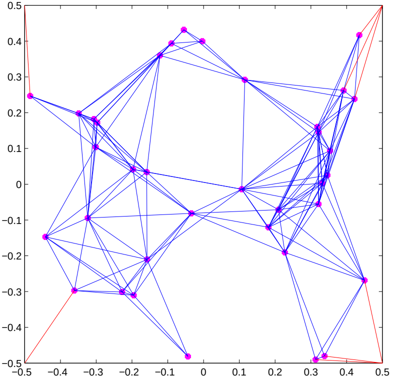
Find $x \in \mathbb{R}^{2\times m}$ such that

$$\|x_i - x_j\|^2 = d_{ij}^2, \qquad (i,j) \in \mathcal{I}_x$$

$$\|a_k - x_j\|^2 = \bar{d}_{kj}^2, \qquad (k,j) \in \mathcal{I}_a$$

Sensor network localization

Example, 4 anchors, 36 sensors



Sensor network localization

Applications

- Wireless sensor network localization
  - habitat monitoring system in the Great Duck Island
  - detecting volcano eruptions
  - industrial control in semiconductor manufacturing plants
  - structural health monitoring
  - military and civilian surveillance
  - moving object tracking
  - asset location
- Molecule conformation
- . . .

Sensor network localization

Solve the least-square problem

$$\min_{x_1,\ldots,x_m} \sum_{(i,j)\in\mathcal{I}_x} \left| \|x_i - x_j\|^2 - d_{ij}^2 \right| + \sum_{(i,j)\in\mathcal{I}_a} \left| \|a_k - x_j\|^2 - \overline{d}_{kj}^2 \right|$$

to global minimum. This is an NP-hard problem.

SDP relaxation

(P. Biswas and Y. Ye, ’04)

Let $X = [x_1\ x_2\ \dots\ x_n]$ be a $d \times n$ unknown matrix. Then

$$\|x_i - x_j\|^2 = (e_i - e_j)^T X^T X (e_i - e_j)$$

$$\|a_k - x_j\|^2 = (a_k; -e_j)^T \begin{bmatrix} I_d \\ X^T \end{bmatrix} [I_d\ X](a_k; -e_j)$$

and the problem becomes

$$(e_i - e_j)^T X^T X (e_i - e_j) = d_{ij}^2$$

$$(a_k; -e_j)^T \begin{pmatrix} I_d & X \\ X^T & Y \end{pmatrix} (a_k; -e_j) = \overline{d}_{kj}^2$$

$$Y = X^T X$$

SDP relaxation

Now relax $Y = X^T X$ to $Y \succeq X^T X$, equivalent to

$$Z = \begin{pmatrix} I_d & X \\ X^T & Y \end{pmatrix} \succeq 0$$

Relaxed problem:

$$\begin{aligned} & \min 0 \\ & \text{subject to} \\ & \quad Z_{1:d,1:d} = I_d \\ & \quad (0; e_i - e_j)^T Z (0; e_i - e_j) = d_{ij}^2 \quad \forall (i,j) \in \mathcal{I}_x \\ & \quad (a_k; -e_j)^T Z (a_k; -e_j) = \overline{d}_{kj}^2 \quad \forall (k,j) \in \mathcal{I}_a \\ & \quad Z \succeq 0 \end{aligned}$$

Full SDP relaxation, FSDP (linear SDP)

SDP relaxation

Equivalent formulation:

$$\min \sum_{(i,j)\in\mathcal{I}_x} \left((0;e_i - e_j)^T Z (0;e_i - e_j) - d_{ij}^2\right)^2 + \sum_{(k,j)\in\mathcal{I}_a} \left((a_k;-e_j)^T Z (a_k;-e_j) - \bar{d}_{kj}^2\right)^2$$

subject to

$$Z_{1:d,1:d} = I_d$$

$$Z \succeq 0$$

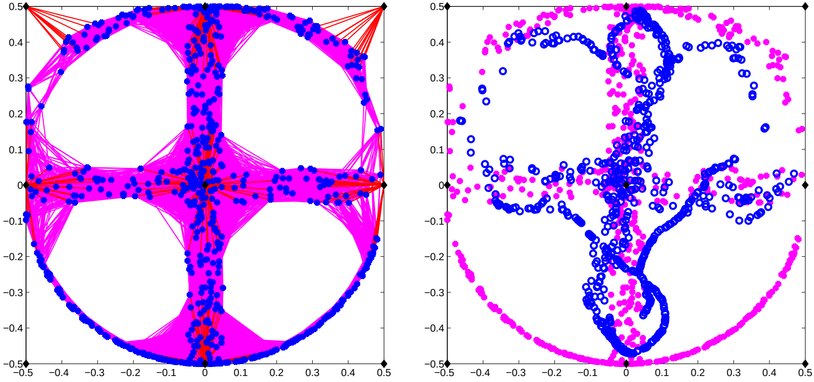
Full SDP relaxation, FSDP (nonlinear SDP)

SDP relaxation

Take the SDP solution as initial approximation for the original unconstrained nonconvex problem. Solve both by PENNON.

Sensor network localization

Example, 9 anchors, 720 sensors



Sensor network localization

Example, 9 anchors, 720 sensors

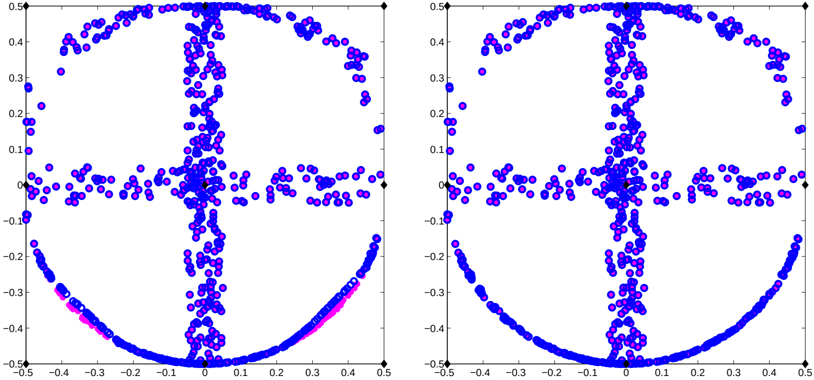


Figure: SDP: 36494 variables, 34334 (4×4) LMIs

Other Applications, Availability

- polynomial matrix inequalities (with Didier Henrion)
- financial mathematics (with Ralf Werner)
- structural optimization with matrix variables and nonlinear matrix constraints (PLATO-N EU FP6 project)
- approximation by nonnegative splines
- approximation of arrival rate function of a non-homogeneous Poisson process (F. Alizadeh, J. Eckstein)
- sensor network localization (with Houduo Xi)
- detection of definite pairs of matrices (with F. Tisseur)

Many other applications. . . . . . any hint welcome

Free academic version of the code available
Free downloadable MATLAB version available soon