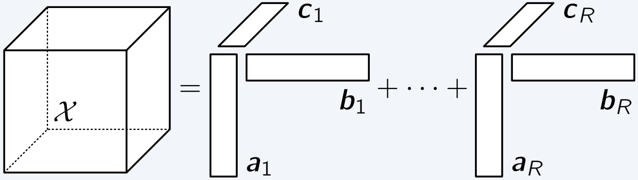
Computing tensor decompositions with complex and polynomial optimization

Laurent Sorber, Marc Van Barel and Lieven De Lathauwer

The canonical polyadic decomposition (CPD) decomposes a tensor into a minimal number of rank-one tensors R



The tensor's rank is defined as R

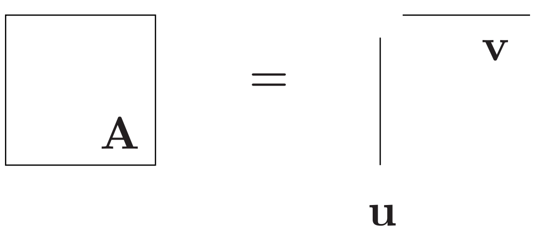
Rank-1 tensor

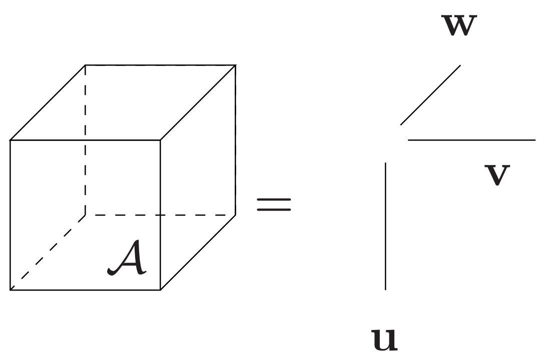
- Rank-1 matrix: outer product of 2 vectors $\mathbf{u}$, $\mathbf{v}$:

$$\begin{aligned} a_{ij} &= u_i v_j \\ \mathbf{A} &= \mathbf{u} \cdot \mathbf{v}^T \equiv \mathbf{u} \circ \mathbf{v} \end{aligned}$$

- Third-order rank-1 tensor: outer product of 3 vectors $\mathbf{u}$, $\mathbf{v}$, $\mathbf{w}$:

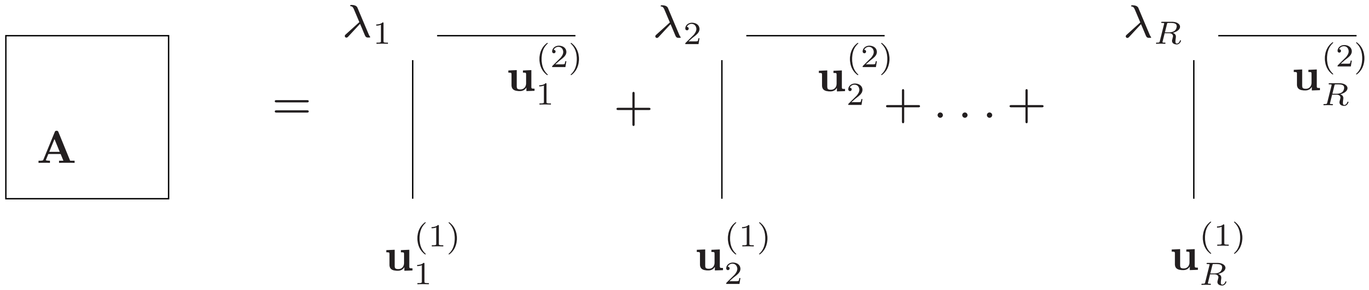
$$\begin{aligned} a_{ijk} &= u_i v_j w_k \\ \mathcal{A} &= \mathbf{u} \circ \mathbf{v} \circ \mathbf{w} \end{aligned}$$



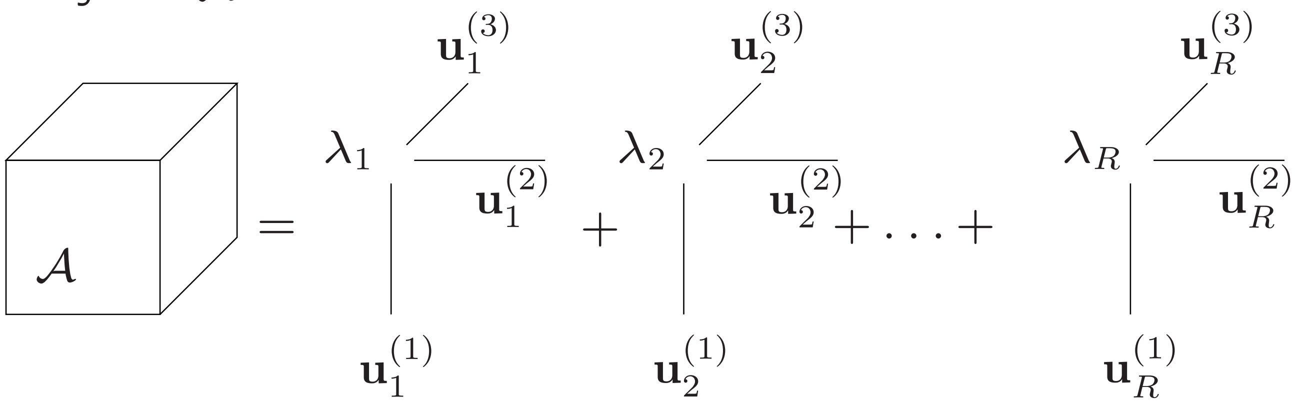


Rank of a tensor

- The rank R of a matrix $\mathbf{A}$ is minimal number of rank-1 matrices that yield $\mathbf{A}$ in a linear combination.



- The rank R of an Nth-order tensor $\mathcal{A}$ is the minimal number of rank-1 tensors that yield $\mathcal{A}$ in a linear combination.

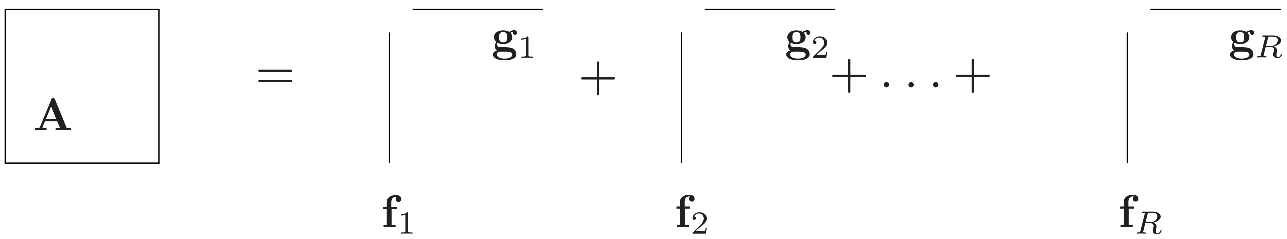


[*Hitchcock, 1927*]

Factor Analysis and Blind Source Separation

- Decompose a data matrix in rank-1 terms that can be interpreted
  E.g. statistics, telecommunication, biomedical applications, chemometrics, data analysis, . . .

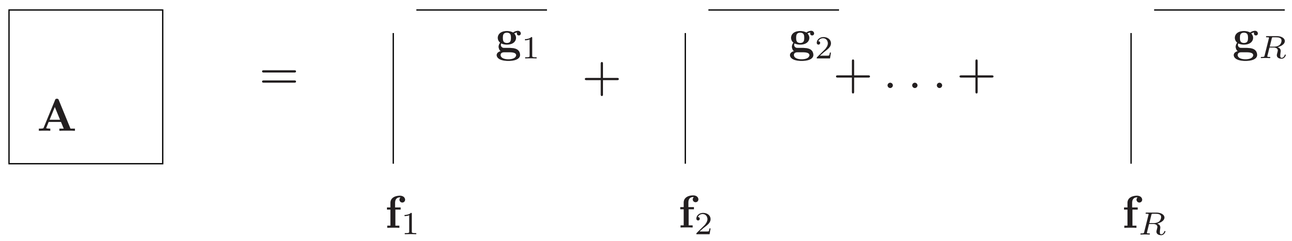
$$\mathbf{A} = \mathbf{F} \cdot \mathbf{G}^T$$



- $\mathbf{F}$: mixing matrix
  $\mathbf{G}$: source signals

- Decompose a data matrix in rank-1 terms that can be interpreted

$$\mathbf{A} = \mathbf{F} \cdot \mathbf{G}^T$$



- Problem: decomposition in rank-1 terms is not unique

$$\begin{aligned} \mathbf{A} &= (\mathbf{F}\mathbf{M}) \cdot (\mathbf{M}^{-1}\mathbf{G}^T) \\ &= \tilde{\mathbf{F}} \cdot \tilde{\mathbf{G}}^T \end{aligned}$$

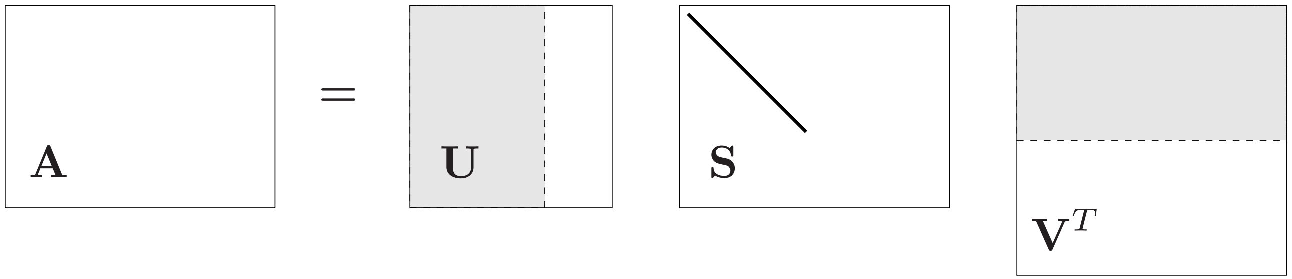
What about SVD?

- SVD is unique
- . . . thanks to orthogonality constraints

$$\mathbf{A} = \mathbf{U} \cdot \mathbf{S} \cdot \mathbf{V}^T = \sum_{r=1}^{R} s_{rr} \mathbf{u}_r \mathbf{v}_r^T$$

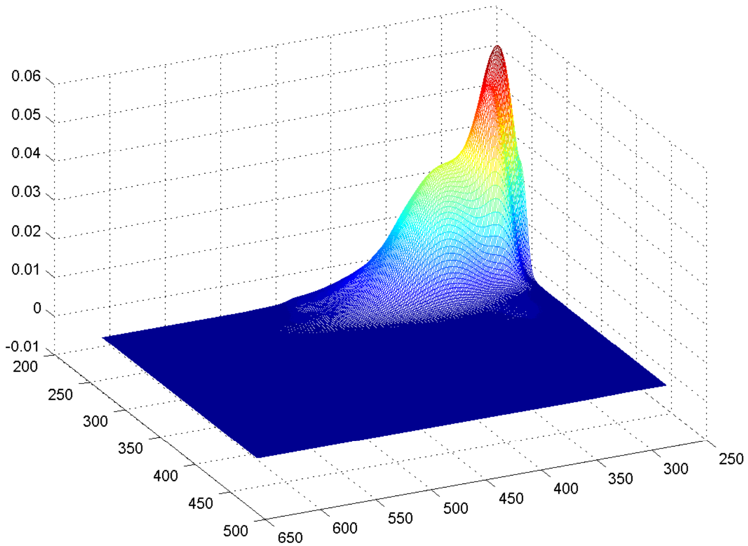
$\mathbf{U}$, $\mathbf{V}$ orthogonal, $\mathbf{S}$ diagonal

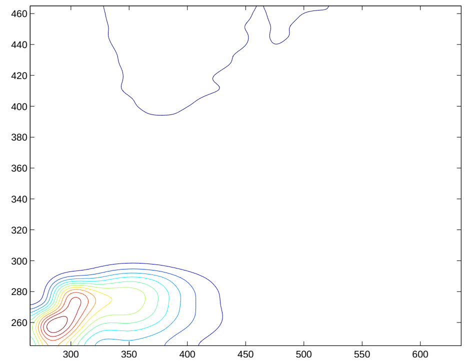
- Whether these constraints make sense, depends on the application
- SVD is great for dimensionality reduction
  best rank-R approximation $\leftarrow$ truncated SVD



An example where matrices fail

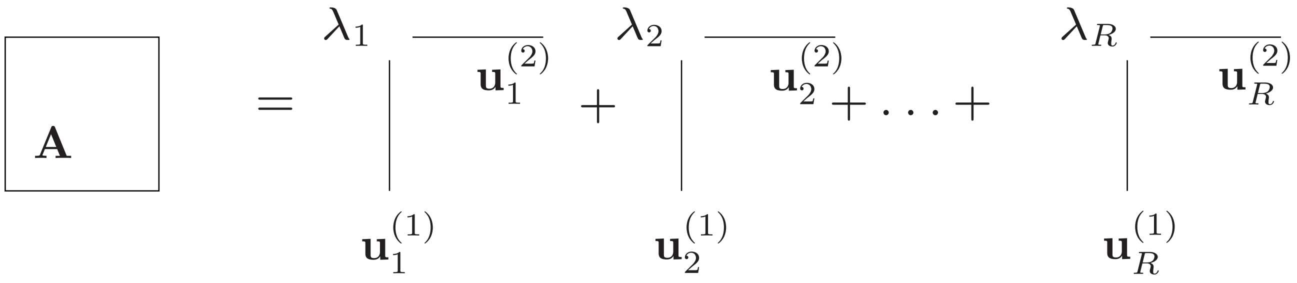
Excitation-emission spectroscopy





Excitation-emission spectroscopy

row vector $\sim$ excitation spectrum
column vector $\sim$ emission spectrum
coefficients $\sim$ concentrations

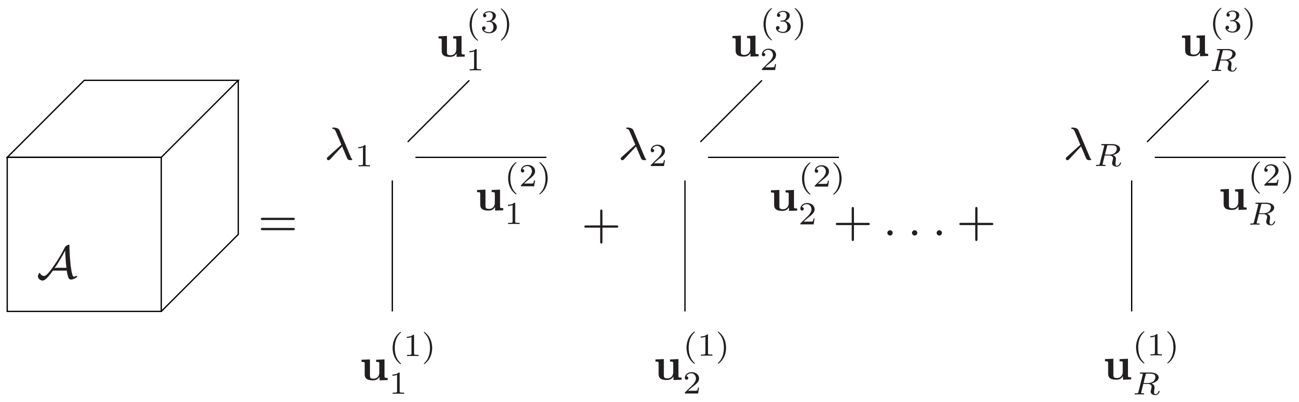


Spectra are nonnegative (and not orthogonal)

Nonnegative Matrix Factorization not unique in general

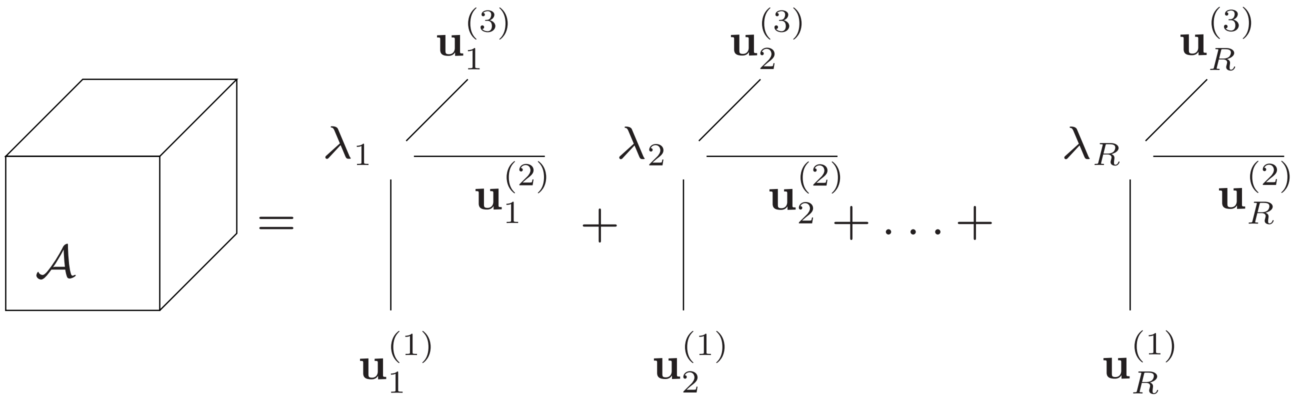
Preview: the tensor approach

row vector $\sim$ excitation spectrum
column vector $\sim$ emission spectrum
coefficients $\sim$ concentrations



Canonical Polyadic Decomposition

Canonical Polyadic Decomposition (CPD): decomposition in minimal number of rank-1 terms [*Harshman '70*], [*Carroll and Chang '70*]



- Unique under mild conditions on number of terms and differences between terms
- Orthogonality (triangularity, ...) not required (but may be imposed)
- Fundamental tool for signal separation

Uniqueness

Deterministic bound: Uniqueness if:

- columns of $\mathbf{U}^{(1)}$: linearly independent
- columns of $\mathbf{U}^{(2)}$: linearly independent
- columns of $\mathbf{U}^{(3)}$: no proportional pair

Generic version:

$$I \geqslant R \qquad J \geqslant R \qquad K \geqslant 2$$

one matrix $\rightarrow$ two (or more) matrices

Indeterminacies: permutation and scaling

Computation: via matrix EVD

[*Sanchez and Kowalski '90*], [*Leurgans et al. '93*], [*Faber et al. '94*]

Toolbox: `cpd_gevd`

Uniqueness (2)

Deterministic bound: Uniqueness if:

- columns of $\mathbf{U}^{(1)}$: linearly independent ("sample mode")
- columns of $\mathbf{C}_2(\mathbf{U}^{(2)} \odot \mathbf{U}^{(3)})$: linearly independent

Generic version:

$$\frac{I(I-1)}{2}\frac{J(J-1)}{2} \geqslant \frac{R(R-1)}{2} \qquad K \geqslant R$$

Computation: via matrix EVD

[*Jiang and Sidiropoulos '06*], [*DL '06*], [*Domanov and DL '12*]

Toolbox: `cpd_sd`

Uniqueness (3)

The k-rank of a matrix **A** is the maximal number such that any set of k columns of **A** is linearly independent.

Deterministic bound: For $\mathcal{A} \in \mathbb{C}^{I \times J \times K}$ uniqueness if

$$k(\mathbf{U}^{(1)}) + k(\mathbf{U}^{(2)}) + k(\mathbf{U}^{(3)}) \geqslant 2R + 2$$

[*Kruskal '77*], [*Sidiropoulos '00*], [*Stegeman and Sidiropoulos '06*], [*Domanov and DL '12*]

Generic version:

$$\text{min}(I, R) + \text{min}(J, R) + \text{min}(K, R) \geqslant 2R + 2$$

Computation: no dedicated algorithm

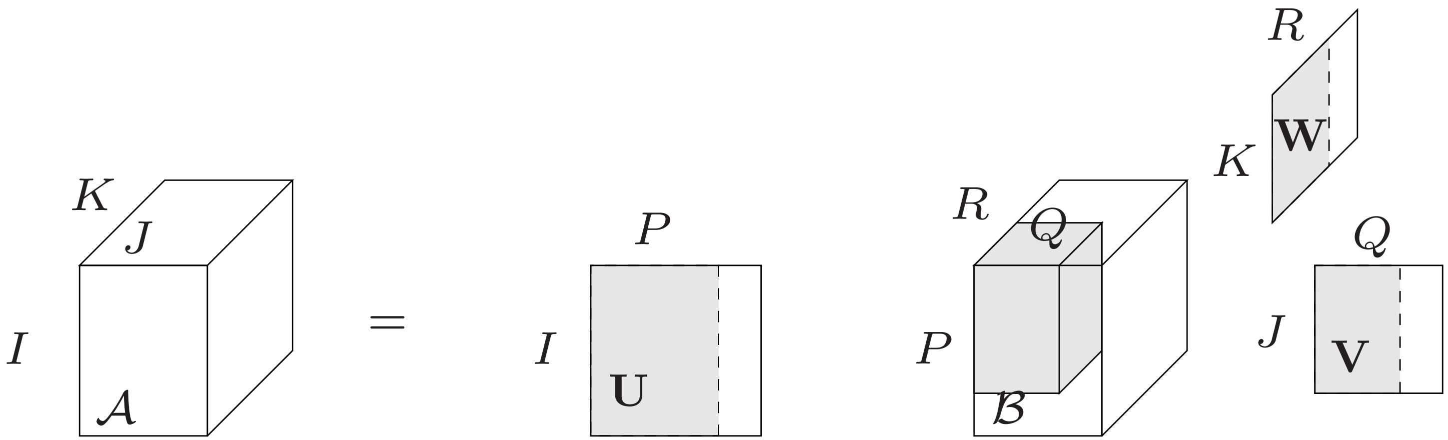
Matrix multiplication

Matrix:

$$\mathbf{A} = \mathbf{U} \cdot \mathbf{B} \cdot \mathbf{V}^T = \mathbf{B} \cdot_1 \mathbf{U} \cdot_2 \mathbf{V} \Leftrightarrow a_{ij} = \sum_{p=1}^{P} \sum_{q=1}^{Q} u_{ip} v_{jq} b_{pq}$$

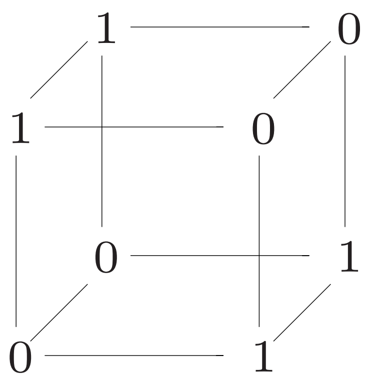
Tensor:

$$\mathcal{A} = \mathcal{B} \cdot_1 \mathbf{U} \cdot_2 \mathbf{V} \cdot_3 \mathbf{W} \Leftrightarrow a_{ijk} = \sum_{p=1}^{P} \sum_{q=1}^{Q} \sum_{r=1}^{R} u_{ip} v_{jq} w_{kr} b_{pqr}$$



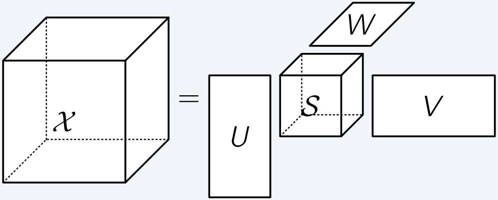
Multilinear rank of a tensor

- Column (row) rank of a matrix: dimension of column (row) space
- Column (row, . . .) rank of a tensor: dimension of column (row, . . .) space
- Tensors: column rank, row rank, . . . can be mutually different
- Rank-(R_1, R_2, R_3) tensor: $\text{rank}_1(\mathcal{A}) = R_1$, $\text{rank}_2(\mathcal{A}) = R_2$, $\text{rank}_3(\mathcal{A}) = R_3$
- Multilinear rank: $\text{rank}_{\boxplus}(\mathcal{A}) = (R_1, R_2, R_3)$

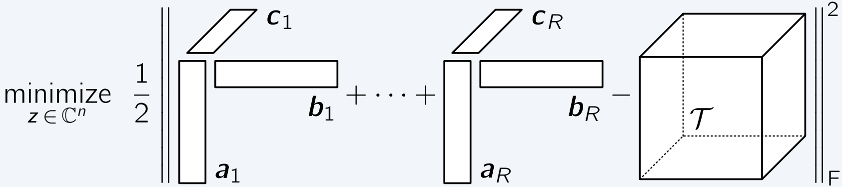


$R_1 = R_2 = 2 \quad R_3 = 1$

A low multilinear rank approximation (LMLRA) decomposes a tensor into a core tensor $\mathcal{S}$ and matrices U, V and W



The tensor's multilinear rank is defined as the triplet $(\text{rank}(U), \text{rank}(V), \text{rank}(W))$



where $z^\top := \begin{bmatrix} a_1^\top & \cdots & a_R^\top & b_1^\top & \cdots & b_R^\top & c_1^\top & \cdots & c_R^\top \end{bmatrix}$

$$\underset{\boldsymbol{x} \in \mathbb{R}^n}{\text{minimize}} \quad f(\boldsymbol{x})$$

$$\underset{\mathbf{z} \in \mathbb{C}^n}{\text{minimize}} \quad f(\mathbf{z}, \overline{\mathbf{z}})$$

$$\underset{z \in \mathbb{C}^n}{\text{minimize}} \quad f(z, \overline{z})$$

- f is not differentiable w.r.t. z
  No real-valued functions are analytic in complex z!

$$\underset{z \in \mathbb{C}^n}{\text{minimize}} \quad f(z, \overline{z})$$

- f is not differentiable w.r.t. z
  No real-valued functions are analytic in complex z!
- Defacto solution is to minimize $f(z_R)$ where $z_R := \begin{bmatrix} \text{Re}\{z\} \\ \text{Im}\{z\} \end{bmatrix}$

$$\underset{z \in \mathbb{C}^n}{\text{minimize}} \quad f(z, \overline{z})$$

- f is not differentiable w.r.t. z
  No real-valued functions are analytic in complex z!
- Defacto solution is to minimize $f(z_R)$ where $z_R := \begin{bmatrix} \mathrm{Re}\{z\} \\ \mathrm{Im}\{z\} \end{bmatrix}$
- Alternatively, use complex optimization [S,VB,DL]

Consider

$$\begin{bmatrix} z \\ \overline{z} \end{bmatrix} = \begin{bmatrix} \mathbb{I} & \mathbb{I}i \\ \mathbb{I} & -\mathbb{I}i \end{bmatrix} \cdot \begin{bmatrix} \mathrm{Re}\{z\} \\ \mathrm{Im}\{z\} \end{bmatrix}$$

$$z_C = J \cdot z_R$$

Consider

$$\begin{bmatrix} z \\ \overline{z} \end{bmatrix} = \begin{bmatrix} \mathbb{I} & \mathbb{I}i \\ \mathbb{I} & -\mathbb{I}i \end{bmatrix} \cdot \begin{bmatrix} \mathrm{Re}\{z\} \\ \mathrm{Im}\{z\} \end{bmatrix}$$

$$z_C = J \cdot z_R$$

and define the complex gradient as

$$\frac{\partial f}{\partial z_C} := J^{-\mathsf{T}} \cdot \frac{\partial f}{\partial z_R} = \frac{1}{2} \begin{bmatrix} \frac{\partial f}{\partial \mathrm{Re}\{z\}} - \frac{\partial f}{\partial \mathrm{Im}\{z\}} i \\ \frac{\partial f}{\partial \mathrm{Re}\{z\}} + \frac{\partial f}{\partial \mathrm{Im}\{z\}} i \end{bmatrix} =: \begin{bmatrix} \frac{\partial f}{\partial z} \\ \frac{\partial f}{\partial \overline{z}} \end{bmatrix}$$

Consider

$$\begin{bmatrix} z \\ \overline{z} \end{bmatrix} = \begin{bmatrix} \mathbb{I} & \mathbb{I}i \\ \mathbb{I} & -\mathbb{I}i \end{bmatrix} \cdot \begin{bmatrix} \mathrm{Re}\{z\} \\ \mathrm{Im}\{z\} \end{bmatrix}$$

$$z_C = J \cdot z_R$$

and define the complex gradient as

$$\frac{\partial f}{\partial z_C} := J^{-\mathsf{T}} \cdot \frac{\partial f}{\partial z_R} = \frac{1}{2} \begin{bmatrix} \frac{\partial f}{\partial \mathrm{Re}\{z\}} - \frac{\partial f}{\partial \mathrm{Im}\{z\}} i \\ \frac{\partial f}{\partial \mathrm{Re}\{z\}} + \frac{\partial f}{\partial \mathrm{Im}\{z\}} i \end{bmatrix} =: \begin{bmatrix} \frac{\partial f}{\partial z} \\ \frac{\partial f}{\partial \overline{z}} \end{bmatrix}$$

Real Taylor series

$$f(\boldsymbol{z}^{(k)}) + \boldsymbol{p}_R^\mathsf{T} \qquad \frac{\partial f(\boldsymbol{z}^{(k)})}{\partial \boldsymbol{z}_R} + \boldsymbol{p}_R^\mathsf{T} \qquad \frac{\partial^2 f(\boldsymbol{z}^{(k)})}{\partial \boldsymbol{z}_R \partial \boldsymbol{z}_R^\mathsf{T}} \qquad \boldsymbol{p}_R$$

Real Taylor series

$$f(\boldsymbol{z}^{(k)}) + \boldsymbol{p}_R^{\mathsf{T}} \cdot J^{\mathsf{T}} J^{-\mathsf{T}} \cdot \frac{\partial f(\boldsymbol{z}^{(k)})}{\partial \boldsymbol{z}_R} + \boldsymbol{p}_R^{\mathsf{T}} \cdot J^{\mathsf{T}} J^{-\mathsf{T}} \cdot \frac{\partial^2 f(\boldsymbol{z}^{(k)})}{\partial \boldsymbol{z}_R \partial \boldsymbol{z}_R^{\mathsf{T}}} \cdot J^{\mathsf{T}} J^{-\mathsf{T}} \cdot \boldsymbol{p}_R$$

Real Taylor series

$$f(z^{(k)}) + \boldsymbol{p}_R^\top \cdot J^\top J^{-\top} \cdot \frac{\partial f(\boldsymbol{z}^{(k)})}{\partial \boldsymbol{z}_R} + \boldsymbol{p}_R^\top \cdot J^\top J^{-\top} \cdot \frac{\partial^2 f(\boldsymbol{z}^{(k)})}{\partial \boldsymbol{z}_R \partial \boldsymbol{z}_R^\top} \cdot J^\top J^{-\top} \cdot \boldsymbol{p}_R$$

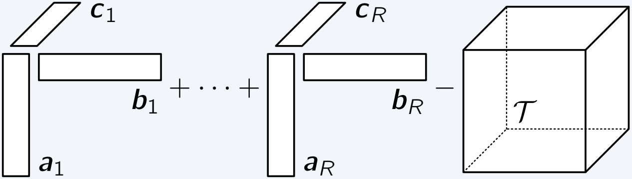
Complex Taylor series

$$f(z^{(k)}) + \boldsymbol{p}_C^\top \cdot \frac{\partial f(\boldsymbol{z}^{(k)})}{\partial \boldsymbol{z}_C} + \boldsymbol{p}_C^\top \cdot \frac{\partial^2 f(\boldsymbol{z}^{(k)})}{\partial \boldsymbol{z}_C \partial \boldsymbol{z}_C^\top} \cdot \boldsymbol{p}_C$$

Complex Optimization Toolbox (COT) for MATLAB
esat.kuleuven.be/sista/cot

- Generalized nonlinear optimization
  `minf_lbfgs`, `minf_lbfgsdl`, `minf_ncg`
- Generalized nonlinear least squares
  `nls_gndl`, `nls_lm`, `nls_gncgs`, `nlsb_gndl`
- Complex differentiation and Moré–Thuente line search
  `deriv`, `ls_mt`

$\underset{\boldsymbol{z}\in\mathbb{C}^n}{\text{minimize}}\quad \frac{1}{2}\Big\|$



$\Big\|_{\mathsf{F}}^{2}$

where $\boldsymbol{z}^\top := \begin{bmatrix}\boldsymbol{a}_1^\top & \cdots & \boldsymbol{a}_R^\top & \boldsymbol{b}_1^\top & \cdots & \boldsymbol{b}_R^\top & \boldsymbol{c}_1^\top & \cdots & \boldsymbol{c}_R^\top\end{bmatrix}$

$$\underset{z \in \mathbb{C}^n}{\text{minimize}} \quad \frac{1}{2}\|\mathcal{M}(z) - \mathcal{T}\|_{\mathsf{F}}^2$$

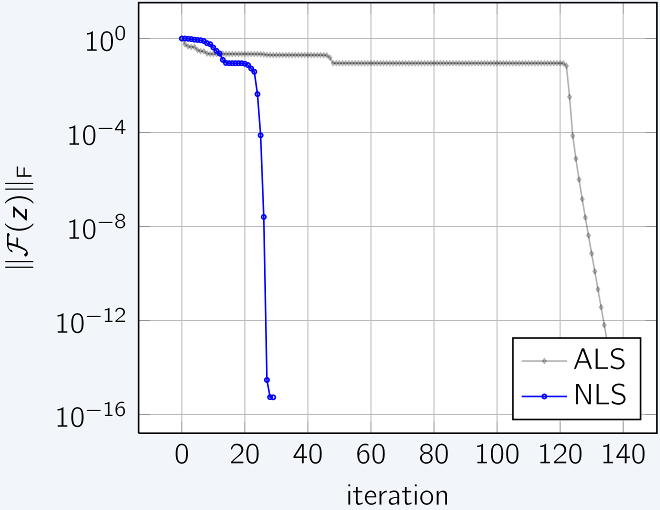
where $\mathcal{M}$ is multilinear

$$\underset{z \in \mathbb{C}^n}{\text{minimize}} \quad \frac{1}{2}\|\mathcal{F}(z)\|_{\mathsf{F}}^2$$

where $\mathcal{F}$ is multilinear

- canonical polyadic decomposition (CPD),
- low multilinear rank approximation (LMLRA),
- block term decompositions (BTD),
- support tensor machines (STM),
- coupled tensor-matrix factorizations (CTMF),
- ...

CPD of a $9 \times 9 \times 9 \times 9 \times 9$ tensor of rank 11



The step is computed as

$$\boldsymbol{p}^* = -H^{-1}\boldsymbol{g}$$

$f(\boldsymbol{z}, \overline{\boldsymbol{z}}) := \frac{1}{2}\|\mathcal{F}(\boldsymbol{z})\|_\mathsf{F}^2$ is the objective function
$\boldsymbol{g} := 2\frac{\partial f}{\partial \overline{\boldsymbol{z}}}$ is the scaled conjugate cogradient
H := is (an approximation of) the complex Hessian

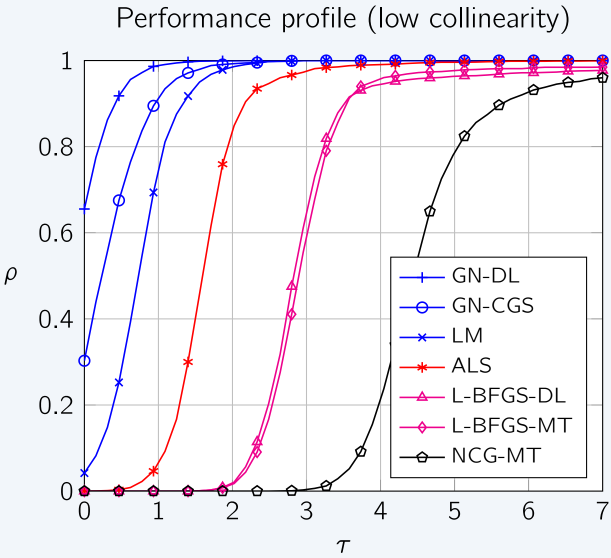
Where H is

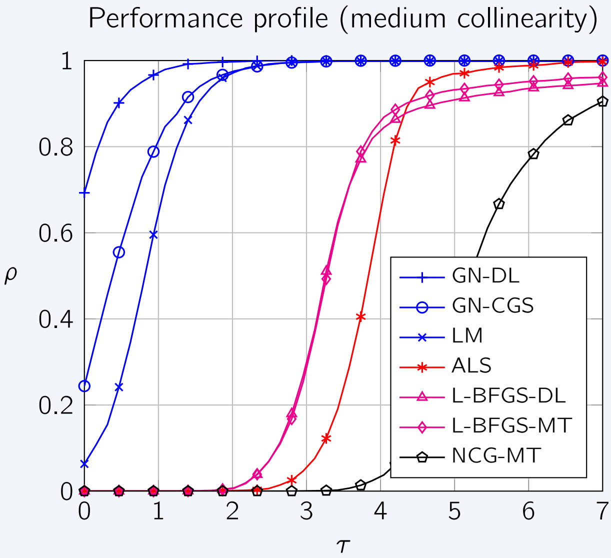
- a diagonal plus low-rank matrix in quasi-Newton
- $J^\mathsf{H} J$ in NLS and $J := \frac{\partial \mathcal{F}}{\partial \boldsymbol{z}^\mathsf{T}}$

- However, NLS is expensive in both memory and flop/iteration
  - NI^2 times more memory than ALS
  - $N^2 R^2$ times more flop/iteration than ALS

- However, NLS is expensive in both memory and flop/iteration
  - NI^2 times more memory than ALS
  - N^2R^2 times more flop/iteration than ALS
- Exploit rank-one and diagonal block structure in $J^{\mathrm{H}}J$ to obtain a fast inexact NLS algorithm [S,VB,DL]
  - Same memory cost as ALS
  - Same flop/iteration as ALS for large tensors

- However, NLS is expensive in both memory and flop/iteration
  - NI^2 times more memory than ALS
  - N^2R^2 times more flop/iteration than ALS
- Exploit rank-one and diagonal block structure in $J^{\mathrm{H}} J$ to obtain a fast inexact NLS algorithm [S,VB,DL]
  - Same memory cost as ALS
  - Same flop/iteration as ALS for large tensors
- Additional benefits (compared to ALS)
  - Almost "embarrassingly" parallel
    Can theoretically achieve peak performance on GPUs
  - Robust performance on difficult decompositions


Performance profile (low collinearity)
1
0.8
0.6
0.4
0.2
0
0
1
2
3
4
5
6
7
ρ
τ
GN-DL
GN-CGS
LM
ALS
L-BFGS-DL
L-BFGS-MT
NCG-MT


Performance profile (medium collinearity)
1
0.8
0.6
0.4
0.2
0
ρ
0
1
2
3
4
5
6
7
τ
GN-DL
GN-CGS
LM
ALS
L-BFGS-DL
L-BFGS-MT
NCG-MT

Tensorlab — a MATLAB toolbox for tensor decompositions
`esat.kuleuven.be/sista/tensorlab`

- Elementary operations on tensors
  Multicore-aware and profiler tuned
- Tensor decompositions with structure and/or symmetry
  CPD, LMLRA, MLSVD, block term decompositions
- Global minimization of bivariate polynomials
  Exact line and plane search for tensor optimization
- Cumulants, tensor visualization, estimating a tensor's rank or multilinear rank, ...

Tensorlab — a MATLAB toolbox for tensor decompositions
`esat.kuleuven.be/sista/tensorlab`

- Elementary operations on tensors
  Multicore-aware and profiler tuned
- Tensor decompositions with structure and/or symmetry
  CPD, LMLRA, MLSVD, block term decompositions
- Global minimization of bivariate polynomials
  Exact line and plane search for tensor optimization
- Cumulants, tensor visualization, estimating a tensor's rank or multilinear rank, . . .

$$\underset{\alpha}{\text{minimize}} \quad \frac{1}{2}\|\mathcal{M}(\mathbf{z} + \alpha\Delta\mathbf{z}) - \mathcal{T}\|_\mathsf{F}^2 \tag{LS}$$

$$\underset{\alpha,\gamma}{\text{minimize}} \quad \frac{1}{2}\|\mathcal{M}(\gamma\mathbf{z} + \alpha\Delta\mathbf{z}) - \mathcal{T}\|_\mathsf{F}^2 \tag{SLS}$$

$$\underset{\alpha,\beta}{\text{minimize}} \quad \frac{1}{2}\|\mathcal{M}(\mathbf{z} + \alpha\Delta\mathbf{z}_1 + \beta\Delta\mathbf{z}_2) - \mathcal{T}\|_\mathsf{F}^2 \tag{PS}$$

$$\underset{\alpha,\beta,\gamma}{\text{minimize}} \quad \frac{1}{2}\|\mathcal{M}(\gamma\mathbf{z} + \alpha\Delta\mathbf{z}_1 + \beta\Delta\mathbf{z}_2) - \mathcal{T}\|_\mathsf{F}^2 \tag{SPS}$$

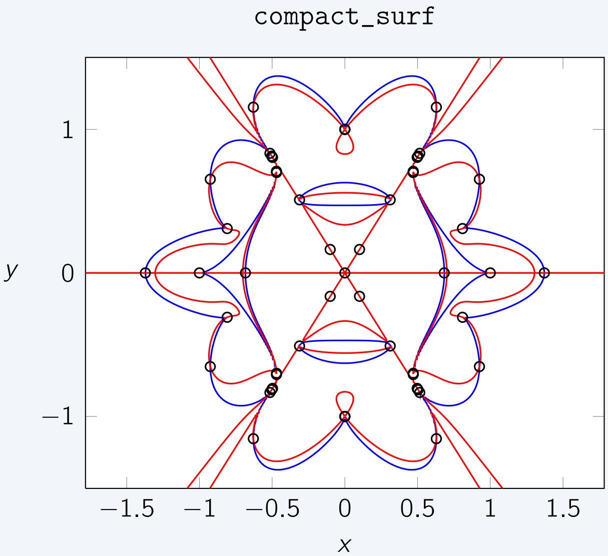
Problem \ Field	$\mathbb{R}$	$\mathbb{C}$
LS	degree $2N$ analytic univariate polynomial	coordinate degree N polyanalytic univariate polynomial
SLS	degree $2N$ analytic univariate rational function	coordinate degree N polyanalytic univariate rational function
PS	total degree $2N$ bivariate polynomial	—
SPS	total degree $2N$ bivariate rational function	—

(S)LS-$\mathbb{C}$ and (S)PS-$\mathbb{R}$ are equivalent to solving

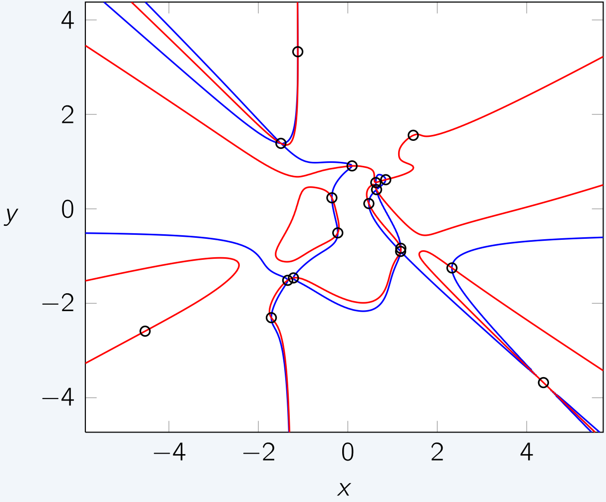
$$\begin{cases} p(x, y) = 0 \\ q(x, y) = 0 \end{cases} \text{ where } x, y \in \mathbb{R}$$

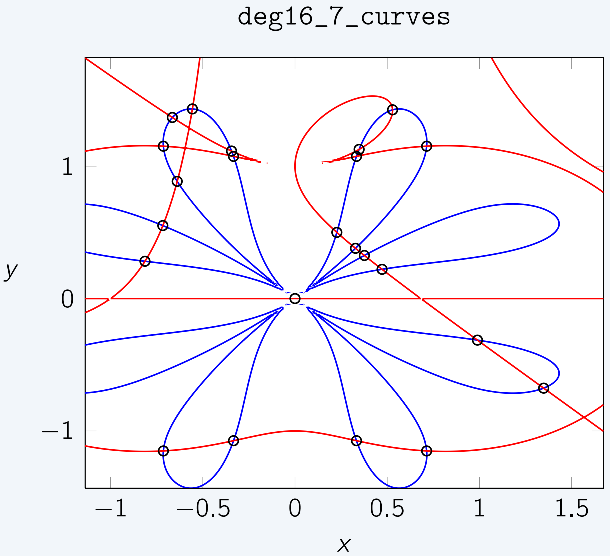
for some polynomials p and q

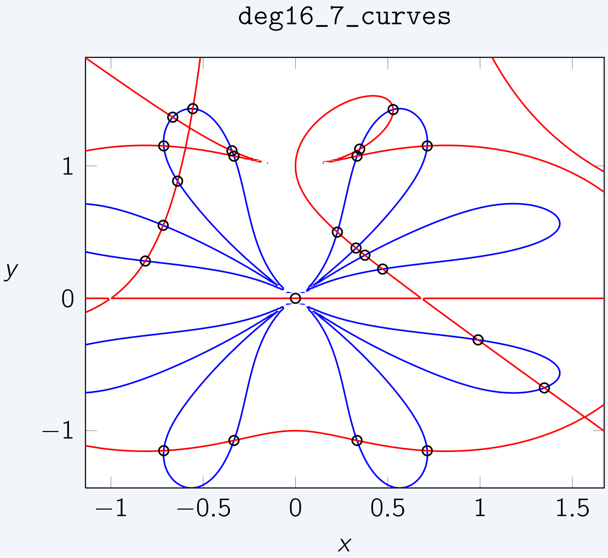
How? Newton's method, interval methods, semidefinite programming, Gröbner bases, resultants, homotopy continuation, . . .

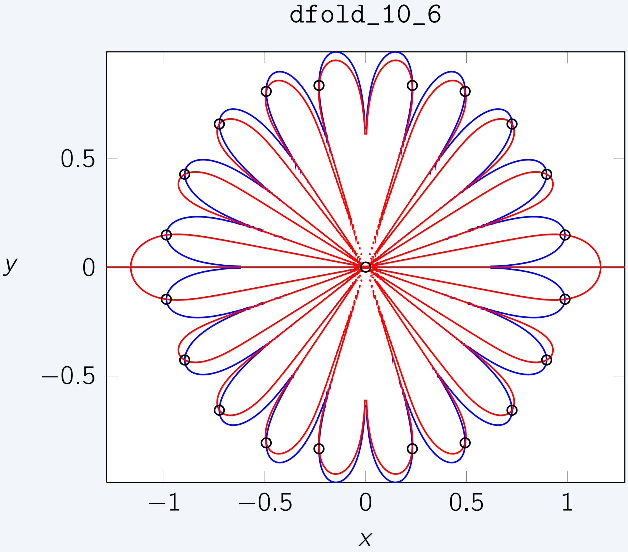

compact_surf
y
x
1
0
−1
−1.5
−1
−0.5
0
0.5
1
1.5

`curve_issac`



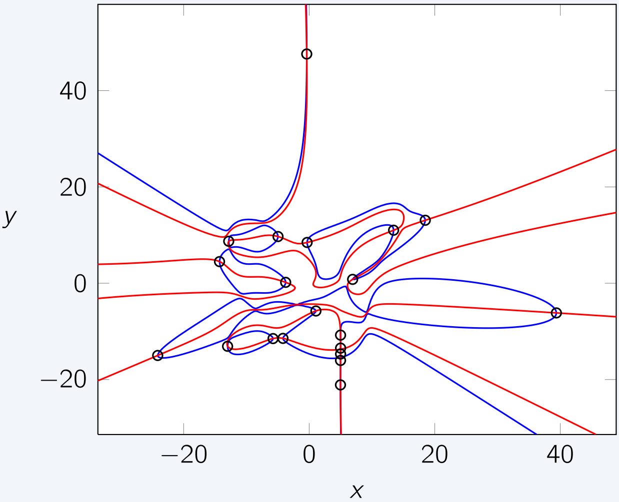

deg16_7_curves
y
x
1
0
−1
−1
−0.5
0
0.5
1
1.5

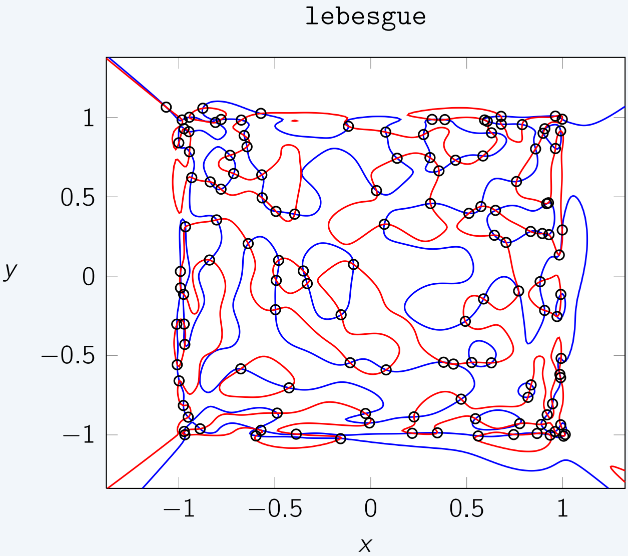

deg16_7_curves
1
0
−1
y
−1
−0.5
0
0.5
1
1.5
x



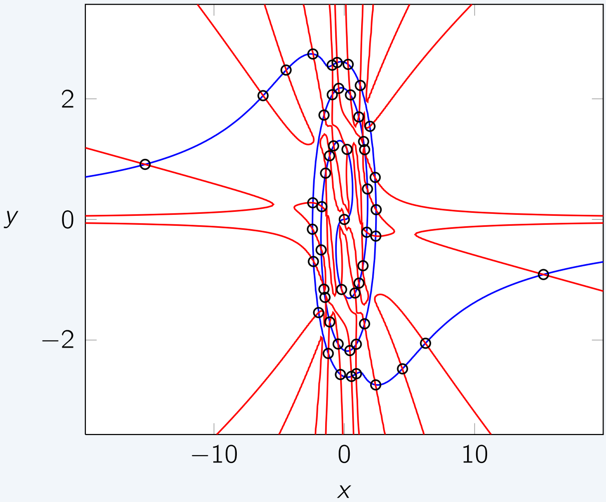
dfold_10_6
0.5
0
−0.5
y
−1
−0.5
0
0.5
1
x

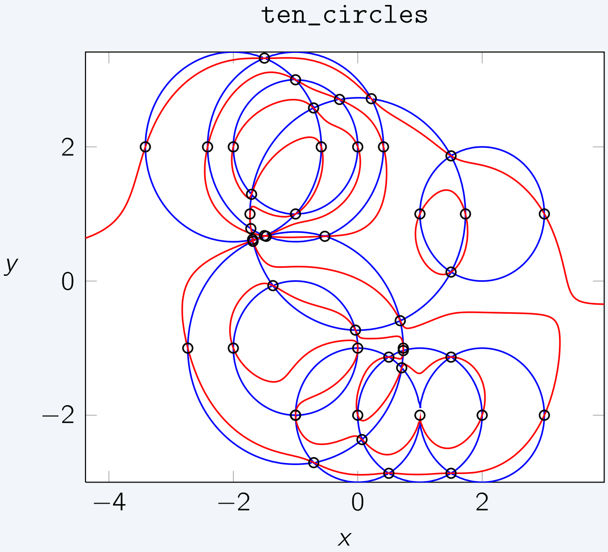
grid_deg_10

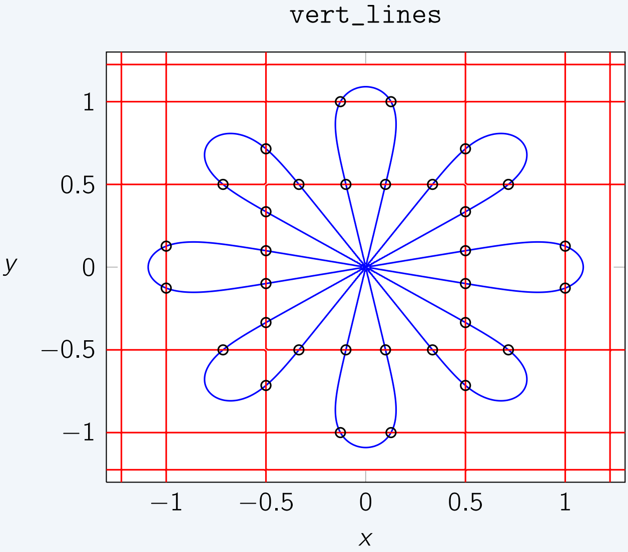


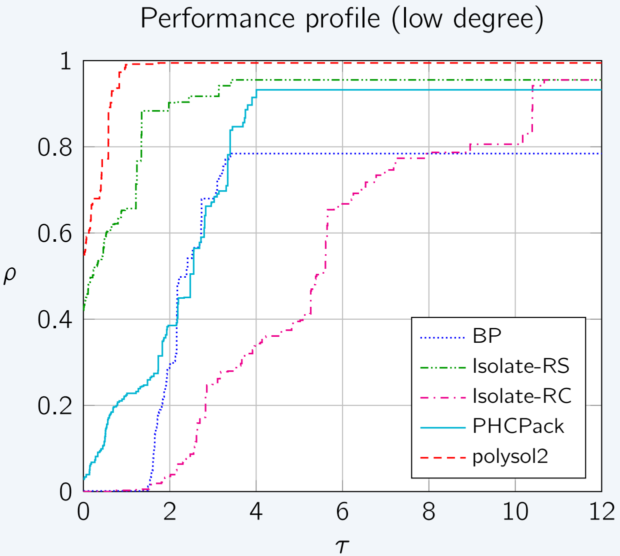

lebesgue
1
0.5
0
−0.5
−1
y
−1
−0.5
0
0.5
1
x

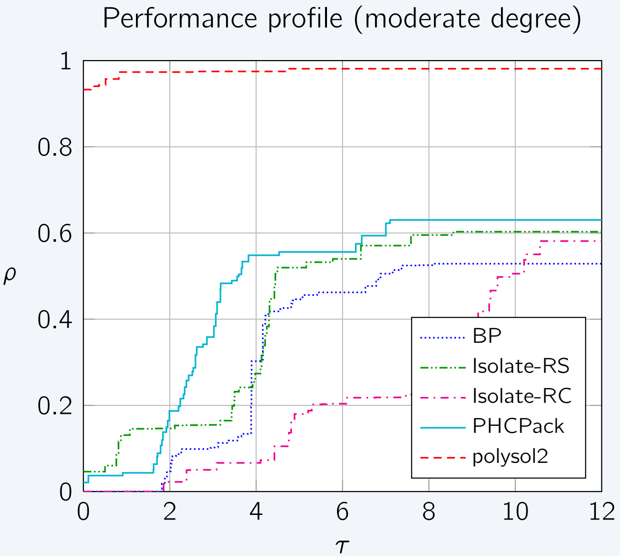
spiral29_24




ten_circles
y
x
2
0
−2
−4
−2
0
2


vert_lines
1
0.5
0
−0.5
−1
y
−1
−0.5
0
0.5
1
x


Performance profile (low degree)
1
0.8
0.6
0.4
0.2
0
ρ
0
2
4
6
8
10
12
τ
BP
Isolate-RS
Isolate-RC
PHCPack
polysol2


Performance profile (moderate degree)
1
0.8
0.6
0.4
0.2
0
ρ
0
2
4
6
8
10
12
τ
BP
Isolate-RS
Isolate-RC
PHCPack
polysol2

- Matrix rank disjoins into two concepts for tensors
- Uniqueness leads to interpretability leads to applications
- Complex Optimization Toolbox & Tensorlab
- Search problems in tensor optimization are polynomial systems
- Bivariate and polyanalytic polynomial systems are GEPs