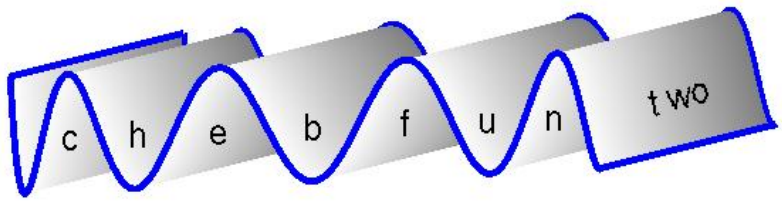
Chebfun2: Bivariate function approximation the ACA way

Alex Townsend
Supervised by Nick Trefethen

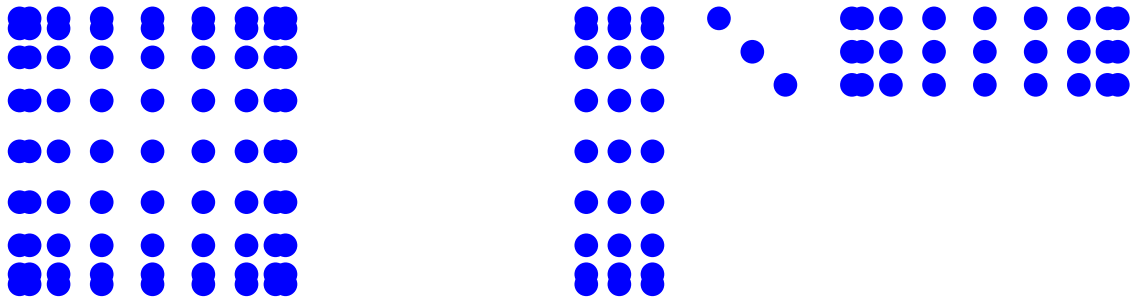
Chebfun and Beyond Workshop

What is Chebfun2?

- **Project:** An attempt to generalise Chebfun to rectangles.
- **Mathematical idea:** A continuous analogue of low-rank matrices.
- **Software:** 12,000 lines of MATLAB code, 203 m-files and 2 users.



Internally, we represent a function by a matrix of values on a Chebyshev tensor grid.



Definition

A rank-k matrix $A \in \mathbb{R}^{m\times n}$ is of **low rank** if

$$k \cdot (m+n) < m \cdot n$$

We're interested in low-rank functions, i.e. values from a Chebyshev tensor grid form a low rank matrix.

Approximate

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix} = \begin{pmatrix} 1 & & & \\ \frac{a_{21}}{a_{11}} & 1 & & \\ \vdots & & \ddots & \\ \frac{a_{n1}}{a_{11}} & & & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & a_{22}^{(1)} & \dots & a_{2n}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2}^{(1)} & \dots & a_{nn}^{(1)} \end{pmatrix}$$

$$a_{ij}^{(1)} = a_{ij} - \frac{a_{i1}}{a_{11}} a_{j1}, \qquad 2 \le i, j \le n.$$

Approximate

$$A = \begin{pmatrix} a_{11} & & & \\ a_{21} & 1 & & \\ \vdots & & \ddots & \\ a_{n1} & & & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{a_{11}} & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} \end{pmatrix}$$

Approximate

$$A = \begin{pmatrix} a_{11} & & \\ a_{21} & 1 & \\ \vdots & & \ddots \\ a_{n1} & & & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{a_{11}} & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} \end{pmatrix}$$

$$A = \begin{pmatrix} a_{11} & & \\ a_{21} & 1 & \\ \vdots & & \ddots \\ a_{n1} & & & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{a_{11}} & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} \end{pmatrix}$$

$$A - \frac{1}{a_{11}} \begin{pmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \end{pmatrix} = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} \end{pmatrix}$$

Approximate

$$A = \begin{pmatrix} a_{11} & & \\ a_{21} & 1 & \\ \vdots & & \ddots \\ a_{n1} & & & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{a_{11}} & & & \\ & 1 & & \\ & & \ddots & \\ & & & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} \end{pmatrix}$$

$$A - \frac{1}{a_{11}} \begin{pmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \end{pmatrix} = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\ \vdots & & \ddots & \vdots \\ 0 & a_{n2}^{(1)} & \cdots & a_{nn}^{(1)} \end{pmatrix}$$

Movie.

History of ACA by mugshots

Eugene Tyrtyshnikov

Bebendorf, Goreinov, Oseledets, Savostyanov, Zamarashkin.

Mario Bebendorf

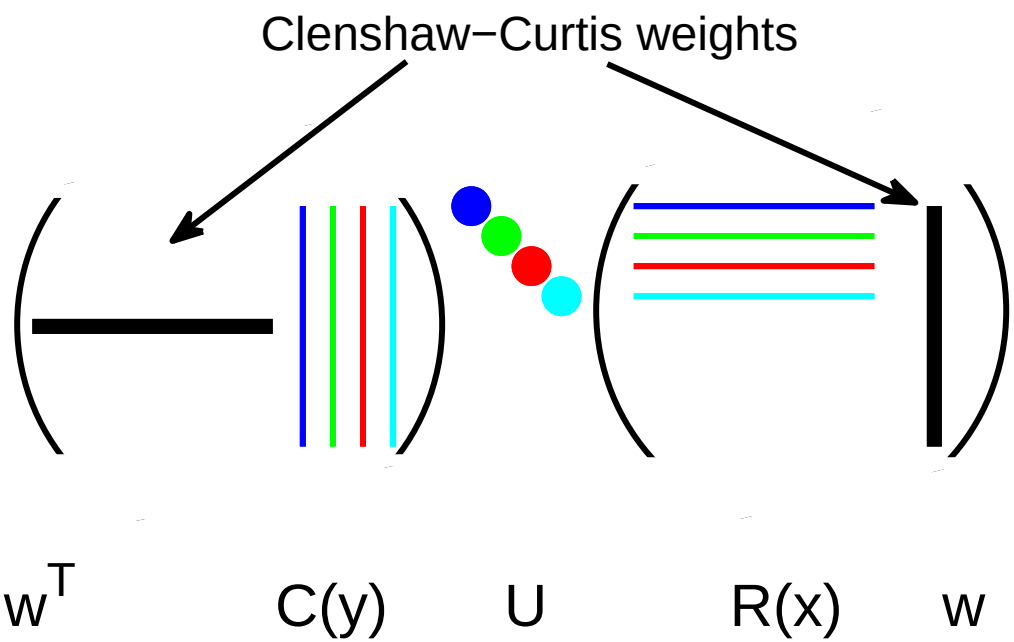
Gesenhues, Griebel, Hackbusch, Rjasanow.

Keith Geddes

Carvajal, Chapman.

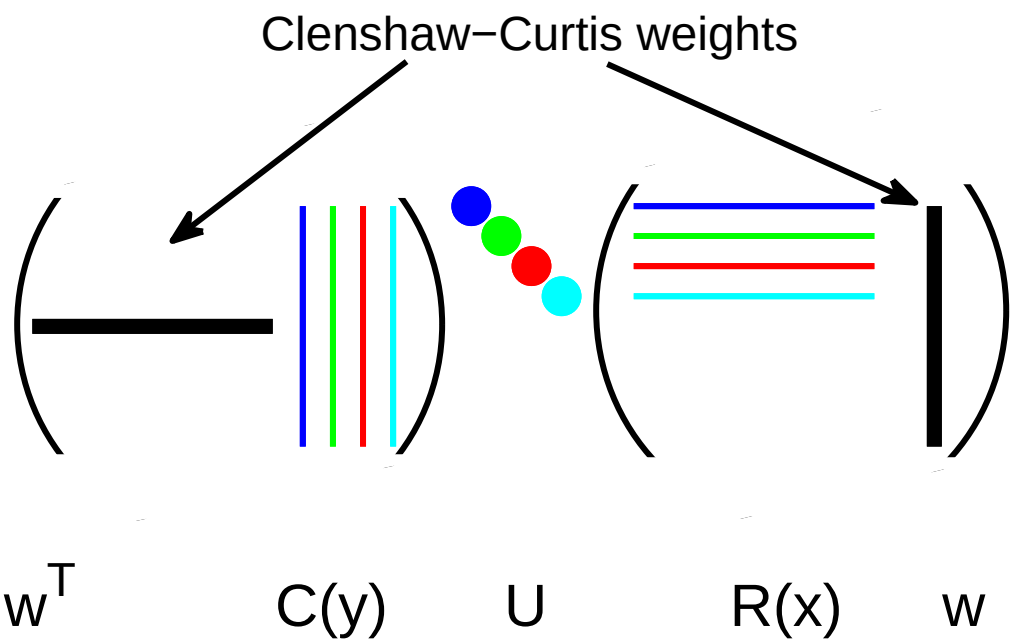
Integrate

$$\int_{-1}^{1}\int_{-1}^{1} f(x,y)dxdy.$$



Integrate

$$\int_{-1}^{1}\int_{-1}^{1} f(x,y)dxdy.$$



Integration takes $\mathcal{O}(n \log n + kn)$ operations. **See Waldvogel's talk for computing weights on Wednesday 10:30.**

Definition (Tensor product operators)

$$\mathcal{L} = \mathcal{L}_y \otimes \mathcal{L}_x, \qquad \mathcal{L}_x \text{ and } \mathcal{L}_y \text{ linear.}$$

Example of tensor product operators

1. Differentiation

$$\mathcal{L} = \frac{\partial}{\partial y}, \qquad \mathcal{L} = \mathcal{D} \otimes \mathcal{I}, \qquad \mathcal{O}(kn) \text{ operations.}$$

Definition (Tensor product operators)

$$\mathcal{L} = \mathcal{L}_y \otimes \mathcal{L}_x, \qquad \mathcal{L}_x \text{ and } \mathcal{L}_y \text{ linear.}$$

Example of tensor product operators

1. Differentiation

$$\mathcal{L} = \frac{\partial}{\partial y}, \qquad \mathcal{L} = \mathcal{D} \otimes \mathcal{I}, \qquad \mathcal{O}(kn) \text{ operations.}$$

2. Discrete Cosine Transform

$$\mathcal{F}_2 : \mathit{vals} \to \mathit{coeffs}, \qquad \mathcal{F}_2 = \mathcal{F} \otimes \mathcal{F}, \qquad \mathcal{O}(kn \log n) \text{ operations.}$$

Definition (Tensor product operators)

$$\mathcal{L} = \mathcal{L}_y \otimes \mathcal{L}_x, \qquad \mathcal{L}_x \text{ and } \mathcal{L}_y \text{ linear.}$$

Example of tensor product operators

1. Differentiation

$$\mathcal{L} = \frac{\partial}{\partial y}, \qquad \mathcal{L} = \mathcal{D} \otimes \mathcal{I}, \qquad \mathcal{O}(kn) \text{ operations.}$$

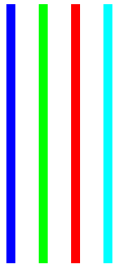
2. Discrete Cosine Transform

$$\mathcal{F}_2 : \mathit{vals} \to \mathit{coeffs}, \qquad \mathcal{F}_2 = \mathcal{F} \otimes \mathcal{F}, \qquad \mathcal{O}(kn \log n) \text{ operations.}$$

3. Evaluation

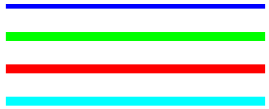
$$\mathcal{E} f = f(x, y), \qquad \mathcal{E} = \mathcal{E}_y \otimes \mathcal{E}_x, \qquad \mathcal{O}(kn^2) \text{ operations.}$$

Orthogonalise: Singular Value Decomposition



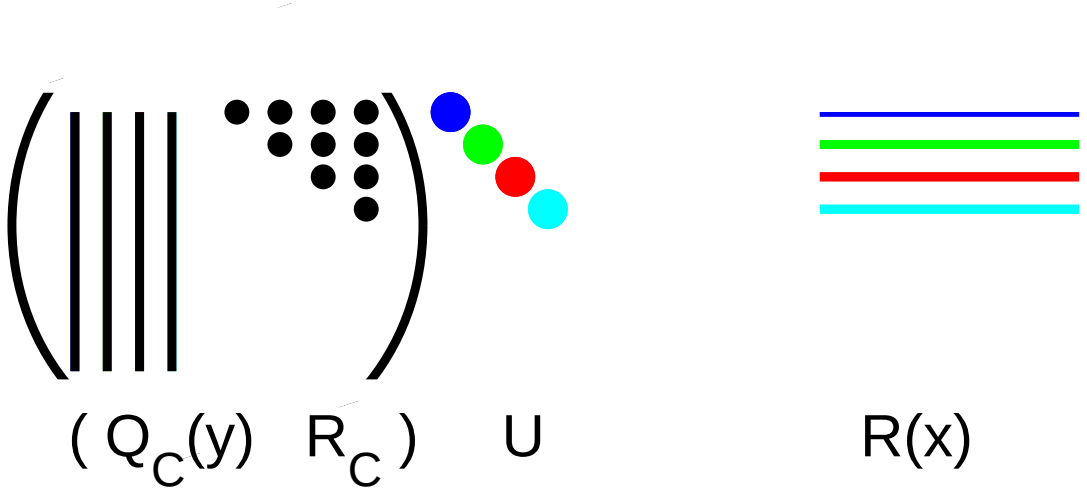
C(y)

U



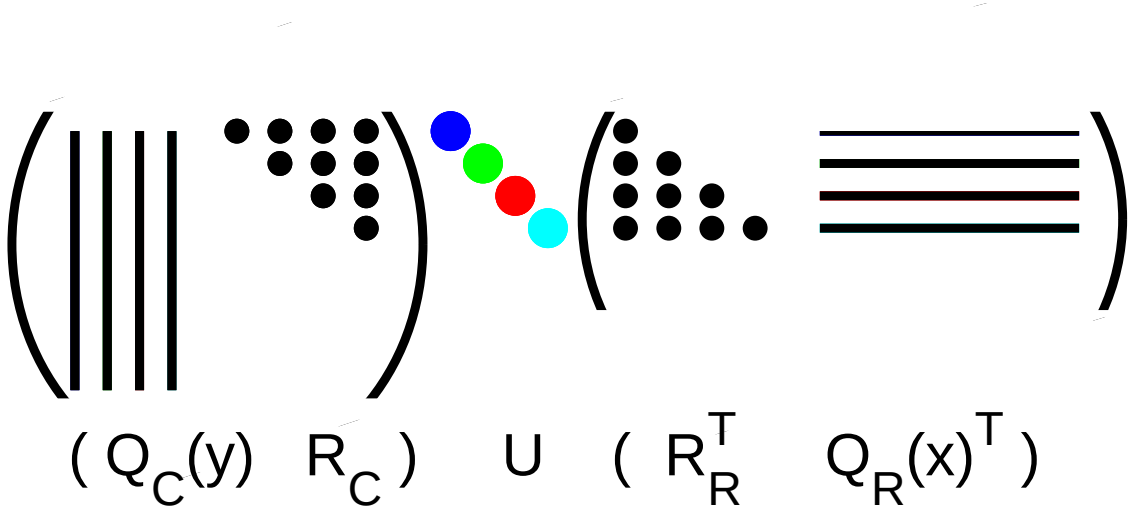
R(x)

Orthogonalise: Singular Value Decomposition



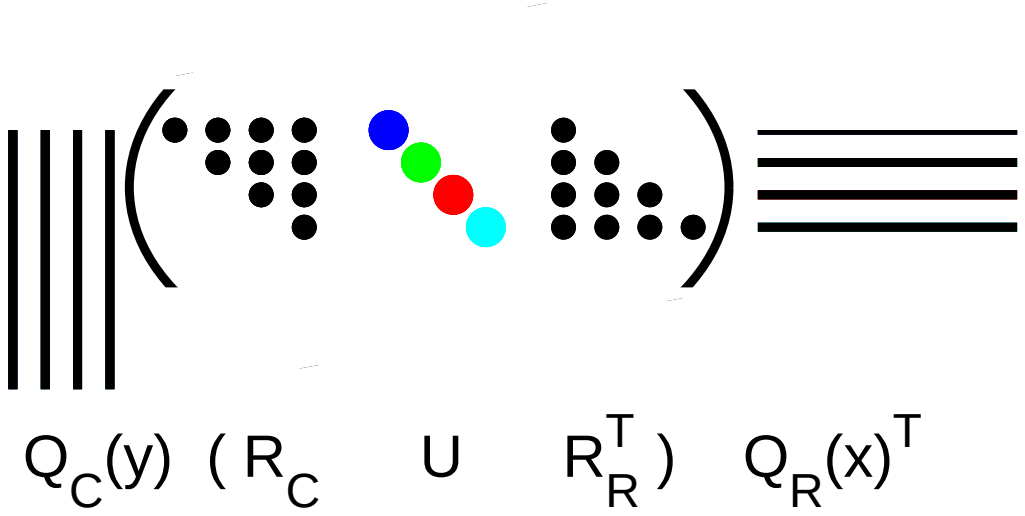
1. Reduced QR factorisation of columns (Trefethen 2009, Gonnet).

Orthogonalise: Singular Value Decomposition



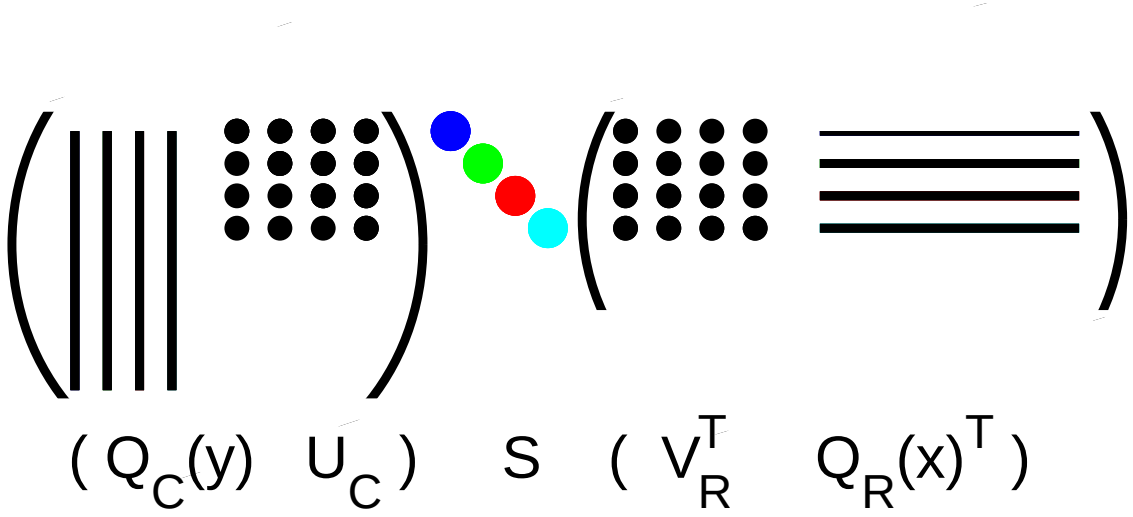
1. Reduced QR factorisation of columns (Trefethen 2009, Gonnet).
2. Reduced QR factorisation of rows.

Orthogonalise: Singular Value Decomposition



1. Reduced QR factorisation of columns (Trefethen 2009, Gonnet).
2. Reduced QR factorisation of rows.
3. Discrete singular value decomposition.

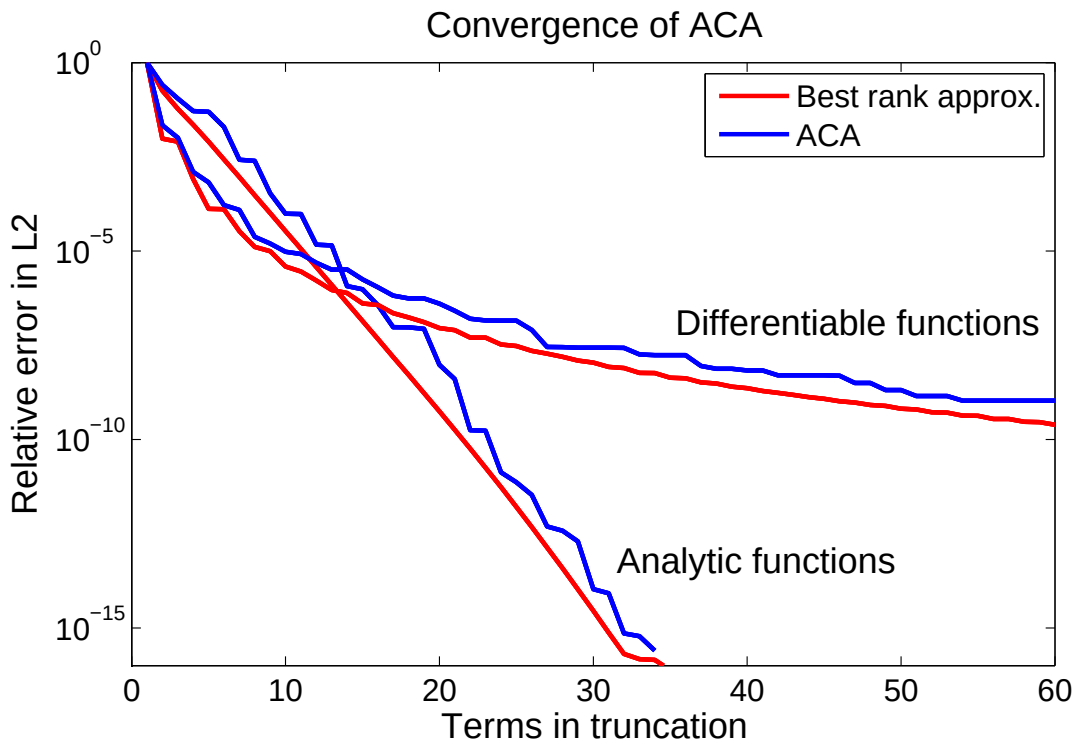
Orthogonalise: Singular Value Decomposition



1. Reduced QR factorisation of columns (Trefethen 2009, Gonnet).
2. Reduced QR factorisation of rows.
3. Discrete singular value decomposition.

Total Operations $\sim 12k^2n + 20k^3$.

Analyse



In practice, ACA computes the **near-optimal** rank approximation to a smooth function.

Optimise: Global Maximum

$$\max\{f(x,y) : x,y \in [-1,1]\}.$$

1. Find the maximum on a Chebyshev tensor grid.
2. Constrained Newton iteration.

Optimise: Global Maximum

$$\max\{f(x,y) : x,y \in [-1,1]\}.$$

1. Find the maximum on a Chebyshev tensor grid.
2. Constrained Newton iteration.

Combinatorial Optimisation: If $c \in \mathbb{R}^k$ and $S \subset \mathbb{R}^k$ then

$$\max\left\{c^\top x : x \in S\right\} = \max\left\{c^\top x : x \in \operatorname{conv}(S)\right\}.$$

Optimise: Global Maximum

$$\max\{f(x,y) : x,y \in [-1,1]\}.$$

1. Find the maximum on a Chebyshev tensor grid.
2. Constrained Newton iteration.

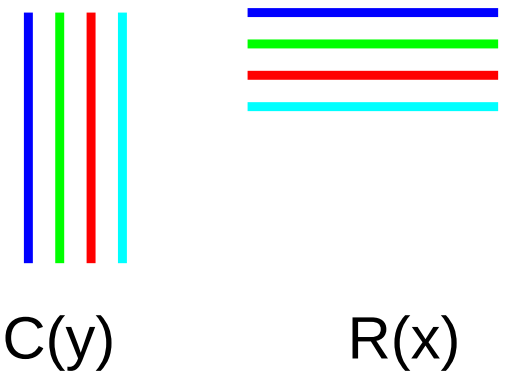
Combinatorial Optimisation: If $c \in \mathbb{R}^k$ and $S \subset \mathbb{R}^k$ then

$$\max\left\{c^\top x : x \in S\right\} = \max\left\{c^\top x : x \in \operatorname{conv}(S)\right\}.$$

Moreover, for any $S, T \subset \mathbb{R}^k$

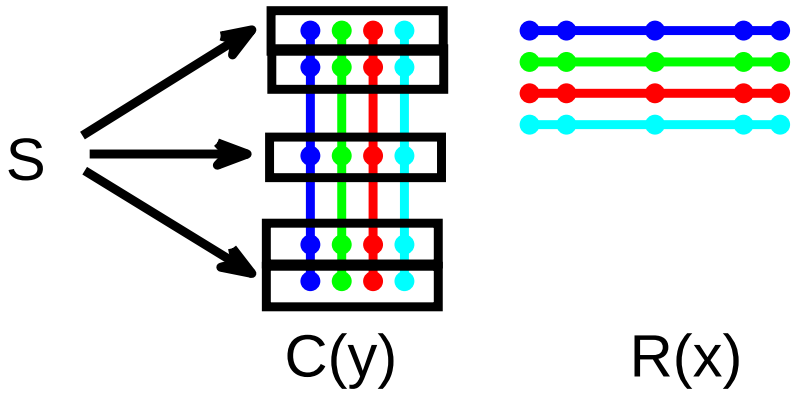
$$\max\left\{y^\top x : x \in S, y \in T\right\} = \max\left\{y^\top x : x \in \operatorname{conv}(S), y \in \operatorname{conv}(T)\right\}.$$

Optimise: Global Maximum



- If $k = 1, 2, 3$ then computing convex hull requires $\mathcal{O}(n \log n)$ operations.
- If $k \geq 4$ then it is more efficient to just compute the discrete maximum in $\mathcal{O}(n^2)$.

Optimise: Global Maximum



- If $k = 1, 2, 3$ then computing convex hull requires $\mathcal{O}(n \log n)$ operations.
- If $k \geq 4$ then it is more efficient to just compute the discrete maximum in $\mathcal{O}(n^2)$.

Associate, illustrate and demonstrate

Let's play. Demo time.

So many questions...

- Is Chebfun2 a good starting point? **Discussion group at 4:30**
- What about representations away from the rectangle? **Huybrechs at 11:30**
- Can we extend things to Partial Differential Equations? **Driscoll at 9am and Olver at 10am tomorrow**
- How to find roots, extrema and contours of bivariate polynomials? **Boyd now**

References

- M. Bebendorf, *Hierarchical Matrices*, Springer, 2008.
- O. A. Carvajal, F. W. Chapman and K. O. Geddes, *Hybrid symbolic-numeric integration in multiple dimensions via tensor-product series*, ISSAC '05 Proceedings, (2005), pp. 84–91.
- S. A. Goreinov, E. E. Tyrtyshnikov and N. L. Zamarashkin, *A theory of pseudo-skeleton approximations*, Linear Algebra Appl., 261 (1997), pp. 1–21.

Approximate

Singular Value Decomposition:
$A = \sum_{j=1}^{n} \sigma_j u_j v_j^\top$

Best rank-r approximation:
$\|A - A_r\|_F^2 = \sum_{j=r+1}^{n} \sigma_j^2$, where
$\|A\|_F^2 = \sum_{i,j=1}^{n} |a_{ij}|^2$.

Karhunen–Loève Expansion:
$f(x,y) = \sum_{j=1}^{\infty} \sigma_j \psi_j(y) \phi_j(x)$

Best rank-r approximation:
$\int\int (f - f_r)^2 = \sum_{j=r+1}^{\infty} \sigma_j^2$.

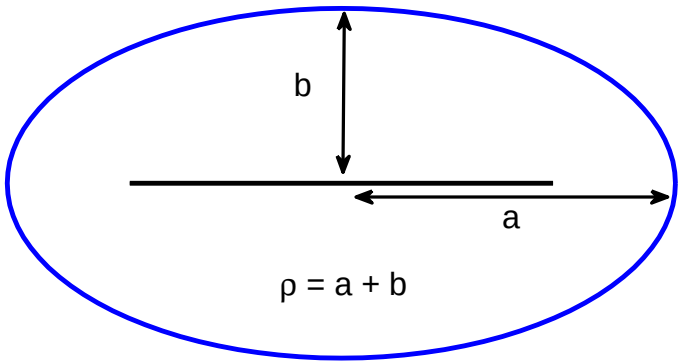
- Compute truncated K-L expansion.
- Singular Value Decomposition is expensive $\mathcal{O}(n^3)$!
- We use Adaptive Cross Approximation with complete pivoting (k steps of Gaussian elimination with complete pivoting)

Theorem (Little and Reade, 1984)

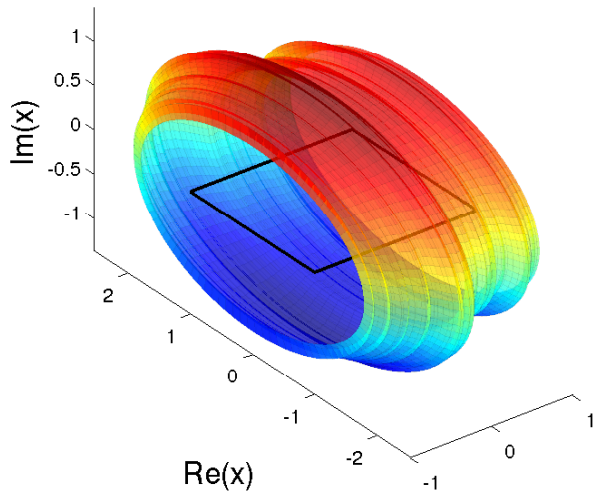
If $f(x,y)$ is symmetric, analytic, bounded by M and for each $y_0 \in [-1,1]$ the function $f(x,y_0)$ is analytically continuable to the open Bernstein ellipse E_ρ. Then we have

$$\left\| f - \sum_{j=1}^{N} \sigma_j \phi_j \otimes \psi_j \right\|_{L_2([-1,1]^2)} \leq \frac{2M\rho^{-N-1}}{1-\rho}.$$

Bernstein ellipse of cos(10x)

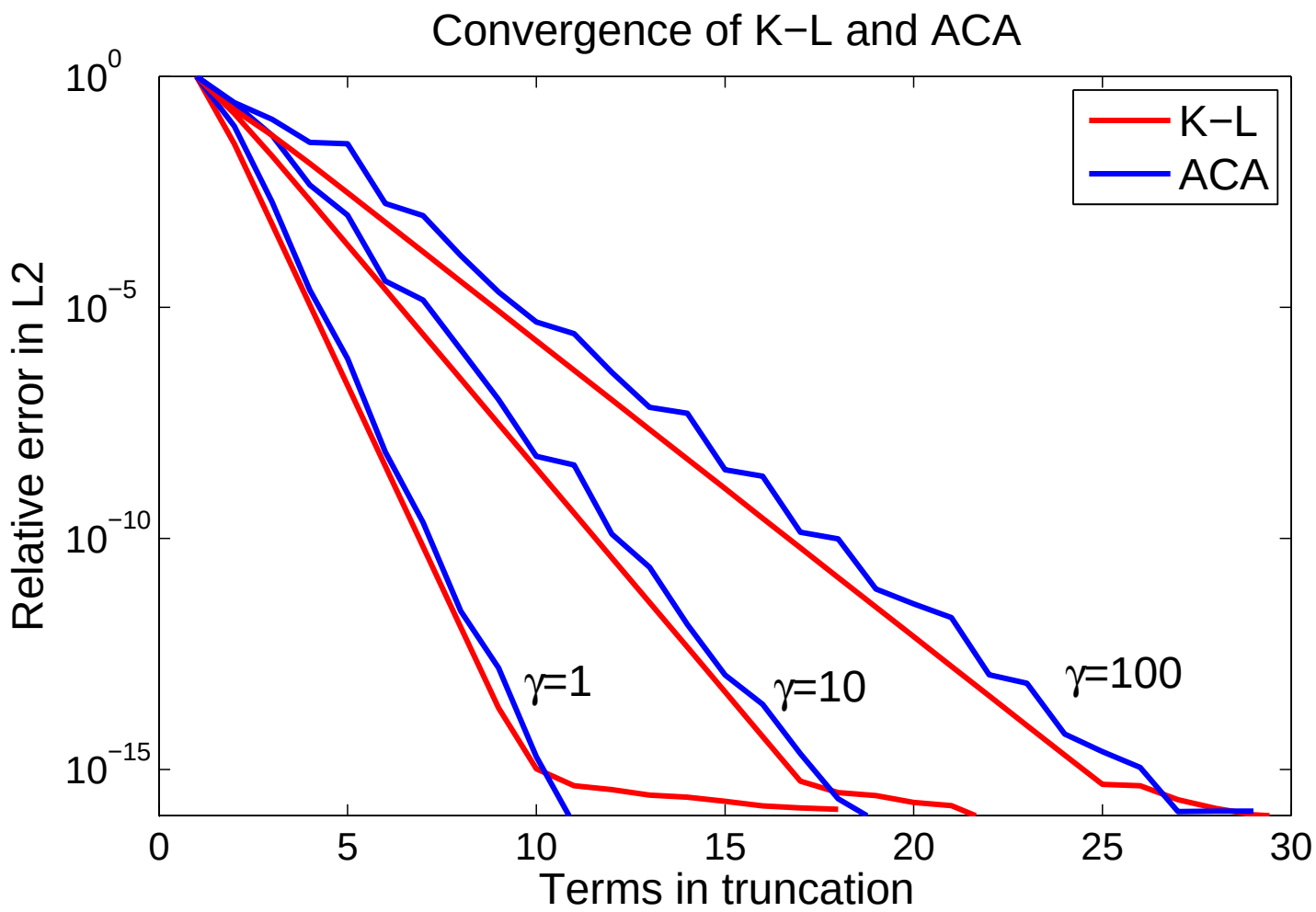


Bernstein Ellipsoid for bivariate function



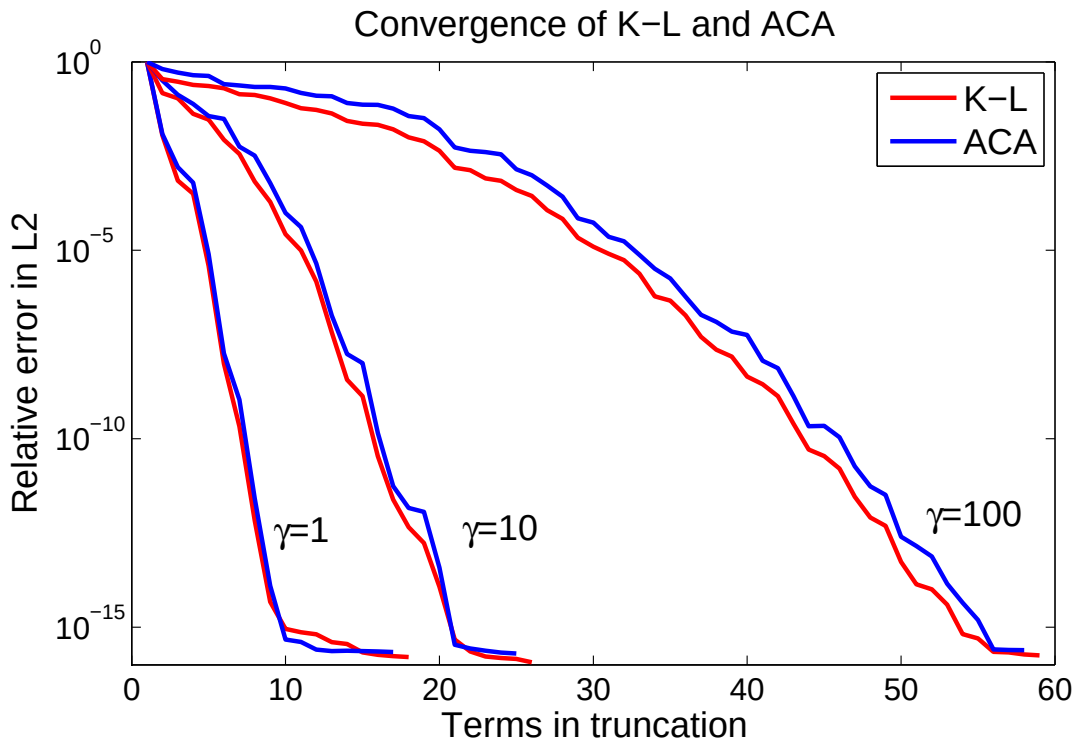
Analyse: Symmetric Analytic functions

$$f(x,y) = \frac{1}{1+\gamma(x^2+y^2)}, \qquad \text{2D Runge function.}$$



Analyse: Supergeometric convergence

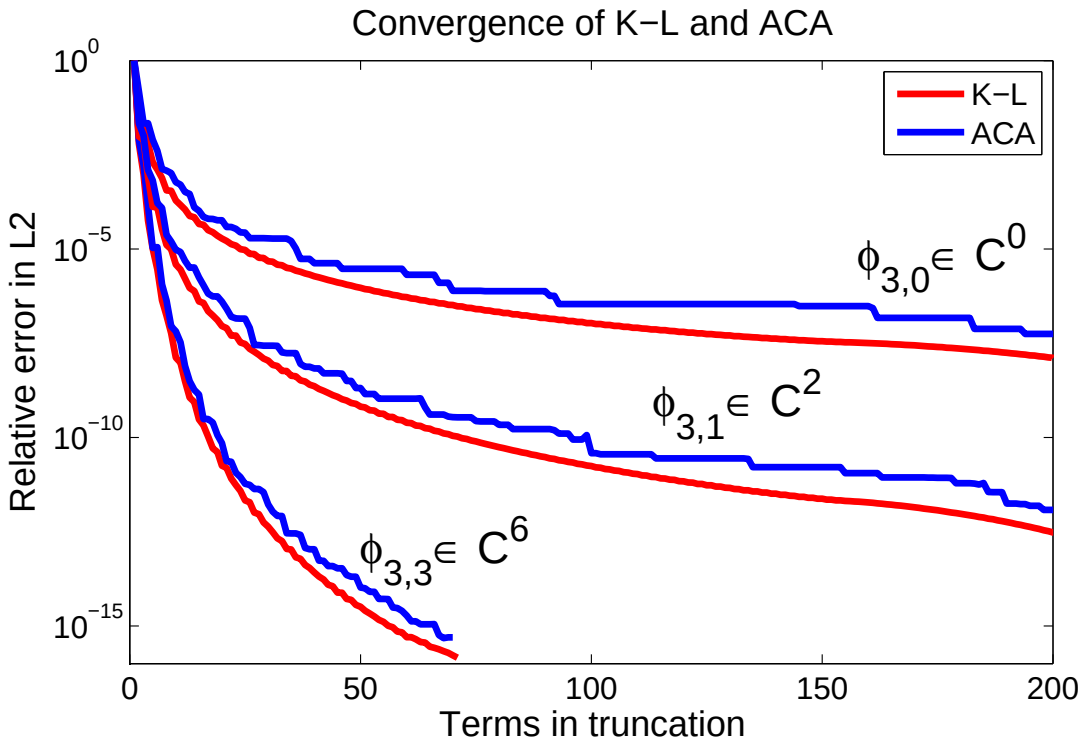
$$f(x,y) = \sum_{k=1}^{100} q_k e^{-\gamma\left((x-x_k)^2+(y-y_k)^2\right)}, \qquad x_k, y_k \in [-1,1]^2, \text{ arbitrary.}$$



$$\left\| f - \sum_{j=1}^{N} \sigma_j \phi_j \otimes \psi_j \right\|_{L_2([-1,1]^2)} \leq \mathcal{O}\left(e^{-\frac{N}{2}\log N}\right).$$

Analyse: Symmetric Differentiable functions

$$f_k(x,y) = \phi_{3,k}(\|x-y\|_2) \in \mathcal{C}^{2k}, \qquad \text{Wendland's CSRBFs}$$



$$\left\| f_k - \sum_{j=1}^{N} \sigma_j \phi_j \otimes \psi_j \right\|_{L_2} \leq \mathcal{O}(N^{-2k+\frac{1}{2}}), k \geq 1.$$