Fast polynomial transforms by low-rank approximation of Hankel matrices and the FFT

Marcus Webb
KU Leuven

Joint work with
Alex Townsend (Cornell) and Sheehan Olver (Imperial)

NUMA internal seminar
9 March 2017

Overview

- Motivation
- The algorithm (for Legendre-to-Chebyshev)
- Low rank matrix approximations
- Generalise to other polynomial bases

Motivation: Chebfun technology

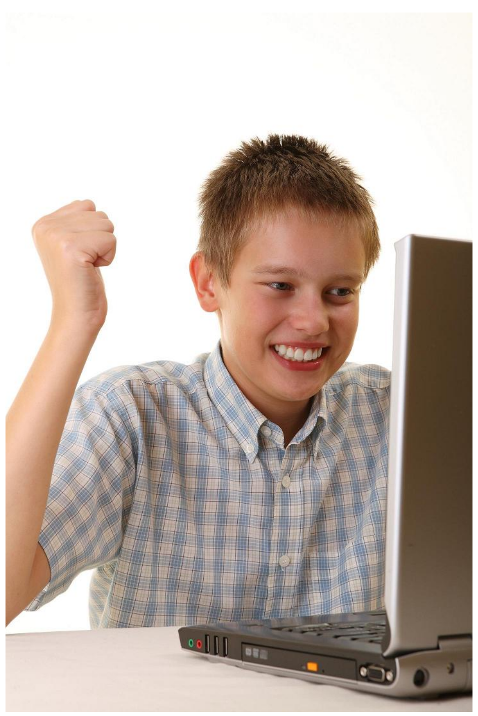
- In 2003 Battles and Trefethen invented chebfun

For the user

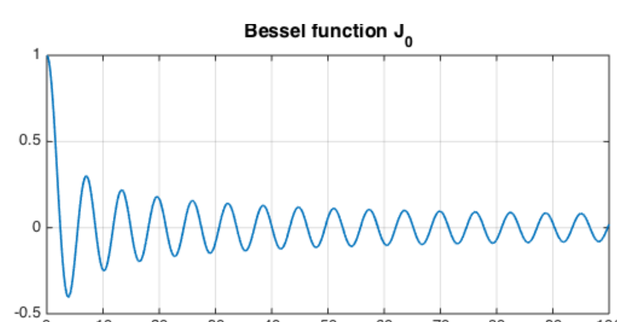
- Feels like symbolic computation
- It's actually fun!

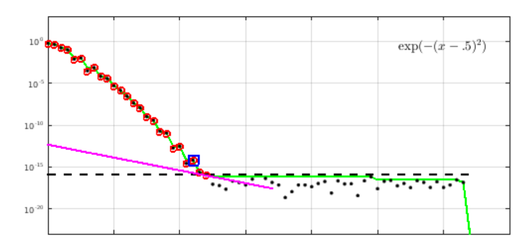
In the code

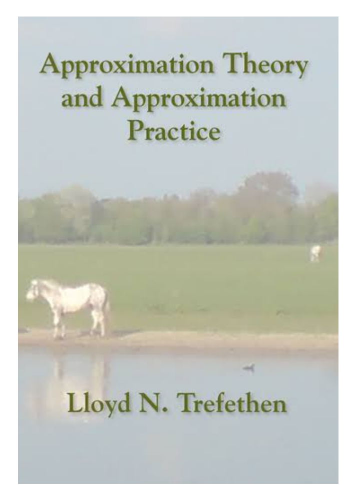
- Robust, automatic polynomial approximation
- Rigorous theory



```
>> J0 = chebfun(@(x) besselj(0,x),[0 100])

J0 =

   chebfun column (1 smooth piece)
       interval       length     endpoint values
[       0,    1e+02]       89          1      0.02
vertical scale =   1

>> diff(J0)

ans =

   chebfun column (1 smooth piece)
       interval       length     endpoint values
[       0,    1e+02]       88   -5.1e-14     0.077
vertical scale = 0.57

>> plot(J0)
```







SIAM J. Sci. Comput.
Vol. 25, No. 5, pp. 1743–1770

AN EXTENSION OF MATLAB TO CONTINUOUS FUNCTIONS AND OPERATORS*

ZACHARY BATTLES† AND LLOYD N. TREFETHEN†

Abstract. An object-oriented MATLAB system is described for performing numerical linear algebra on continuous functions and operators rather than the usual discrete vectors and matrices. About eighty MATLAB functions from `plot` and `sum` to `svd` and `cond` have been overloaded so that one can work with our "chebfun" objects using almost exactly the usual MATLAB syntax. All functions live on $[-1,1]$ and are represented by values at sufficiently many Chebyshev points for the polynomial interpolant to be accurate to close to machine precision. Each of our overloaded operations raises questions about the proper generalization of familiar notions to the continuous context and about appropriate methods of interpolation, differentiation, integration, zerofinding, or transforms. Applications in approximation theory and numerical analysis are explored, and possible extensions for more substantial problems of scientific computing are mentioned.

Key words. MATLAB, Chebyshev points, interpolation, barycentric formula, spectral methods, FFT

AMS subject classifications. 41-04, 65D05

DOI. 10.1137/S1064827503430126

1. Introduction. Numerical linear algebra and functional analysis are two faces of the same subject, the study of linear mappings from one vector space to another. But it could not be said that mathematicians have settled on a language and notation that blend the discrete and continuous worlds gracefully. Numerical analysts favor a concrete, basis-dependent matrix-vector notation that may be quite foreign to the functional analysts. Sometimes the difference may seem very minor between, say, expressing an inner product as (u,v) or as u^Tv. At other times it seems more substantial, as, for example, in the case of Gram–Schmidt orthogonalization, which a numerical analyst would interpret as an algorithm, and not necessarily the best one, for computing a matrix factorization $A = QR$. Though experts see the links, the discrete and continuous worlds have remained superficially quite separate; and, of course, sometimes there are good mathematical reasons for this, such as the distinction between spectrum and eigenvalues that arises for operators but not matrices.

The purpose of this article is to explore some bridges that may be built between discrete and continuous linear algebra. In particular we describe the "chebfun" software system in object-oriented MATLAB, which extends many MATLAB operations on vectors and matrices to functions and operators. This system consists of about eighty M-files taking up about 100KB of storage. It can be downloaded from http://www.comlab.ox.ac.uk/oucl/work/nick.trefethen/, and we assure the reader that going through this paper with a computer at hand is much more fun.

Core MATLAB contains hundreds of functions. We have found that this collection has an extraordinary power to focus the imagination. We simply asked ourselves, for one MATLAB operation after another, what is the "right" analogue of this operation in the continuous case? The question comes in two parts, conceptual and algorithmic. What should each operation mean? And how should one compute it?

*Received by the editors June 18, 2003; accepted for publication (in revised form) November 16, 2003; published electronically May 20, 2004.
http://www.siam.org/journals/sisc/25-5/43012.html
†Computing Laboratory, Oxford University, Wolfson Building, Parks Road, Oxford OX13QD, England (battles@comlab.ox.ac.uk, LNT@comlab.ox.ac.uk).

Descendents: Chebfun2, ApproxFun (in Julia), RKToolbox…

Chebyshev vs Legendre

- To approximate a function, we can expand in a Chebyshev or Legendre polynomial expansion:

$$f_N(x) = \sum_{k=0}^{N} a_k^{\text{cheb}} T_k(x) = \sum_{k=0}^{N} a_k^{\text{leg}} P_k(x)$$

Chebyshev polynomials

$$T_k(x) = \cos(k \cos^{-1}(x)) \qquad x_k = \cos\left(\frac{k\pi}{N}\right)$$

- Change of variables from Cosine series, so

$$(f_N(x_0), f_N(x_1), \ldots, f_N(x_N))$$

$$DCT \updownarrow \mathcal{O}(N \log N)$$

$$(a_0^{\text{cheb}}, a_1^{\text{cheb}}, \ldots, a_N^{\text{cheb}})$$

- Many nice results inherited from Fourier series

Legendre polynomials

- Orthogonal: $\int_{-1}^{1} P_j(x) P_k(x)\, \mathrm{d}x = 0 \text{ if } j \neq k$
- Fourier transform is nice: $\hat{P}_k(\xi) = 2(-i)^k j_k(\xi)$
- Fast $\mathcal{O}(N^2)$ convolution algorithms (Hale-Townsend 2014)
- Cauchy transform has rapidly decaying series (Olver 2012). Riemann-Hilbert problems.
- Connections to spherical harmonics

- Both have fast, accurate algorithms for **derivatives**, **integration**, **root finding**, **optimisation** (but Chebyshev is often faster)

State of the art conversion algorithms

- Timeline for Chebyshev—Legendre conversion

Year	Authors	Complexity	Comments
$\leq$1970s	Piessens, Gallagher, Wise, Allen	$\mathcal{O}(N^2)$	Direct
1986	Orszag	$\mathcal{O}(N\log(N)^2/\log\log N)$	Slow asymptotic expansion
1991	Alpert, Rokhlin	$\mathcal{O}(N\log(N)^2)$	Hierarchical data structures
1998	Potts, Steidl, Tasche	$\mathcal{O}(N\log(N)^2)$	Divide-and-conquer
1999	Mori, Suda, Sugihara	$\mathcal{O}(N\log N)$	Unstable for large N
2011	Iserles	$\mathcal{O}(N\log N)$	Values in the complex plane
2013	Hale, Townsend	$\mathcal{O}(N\log(N)^2/\log\log N)$	Fast asymptotic expansion

- Fast algorithms for **ultrapherical**, **Jacobi** polynomials: Cantero-Iserles 2012, Wang-Huybrechs 2014, Slevinsky 2016

- **First, we tackle Leg-to-Cheb**. Then generalise.

- New method is $\mathcal{O}(N\log(N)^2)$, and has added benefits. Hence now used in Chebfun and ApproxFun

Connection coefficient matrix

- For any two polynomial bases (degree-graded) there is a connection coefficients matrix,

$$\begin{pmatrix} a_0^{\text{cheb}} \\ a_1^{\text{cheb}} \\ a_2^{\text{cheb}} \\ \vdots \end{pmatrix} = \underbrace{\begin{pmatrix} c_{00} & c_{01} & c_{02} & \cdots \\ 0 & c_{11} & c_{12} & \cdots \\ 0 & 0 & c_{22} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix}}_{C} \begin{pmatrix} a_0^{\text{leg}} \\ a_1^{\text{leg}} \\ a_2^{\text{leg}} \\ \vdots \end{pmatrix}$$

- The entries satisfy $P_k(x) = \sum_{j=0}^{k} c_{jk} T_k(x)$
- The problem is reduced to computing $\underline{b} = C\underline{a}, \quad \underline{a} \in \mathbb{C}^{N+1}$
- Naïve method is $\mathcal{O}(N^2)$. Best for $N < 1000$

Leg-to-Cheb matrix

- The connection coefficients are (Gegenbauer 1884):

$$c_{jk} = \frac{2}{\pi}\frac{\Gamma\left(\frac{k-j}{2}+\frac{1}{2}\right)}{\Gamma\left(\frac{k-j}{2}+1\right)}\frac{\Gamma\left(\frac{k+j}{2}+\frac{1}{2}\right)}{\Gamma\left(\frac{k+j}{2}+1\right)}, \text{ if } 0 \le j \le k \le N,\ j-k \text{ even}$$

and the first row is halved. Other entries =0.

- This is a Hadamard product $C = D(T \circ H)$

$$D = \frac{1}{\pi}\begin{pmatrix} 1 & & & & \\ & 2 & & & \\ & & 2 & & \\ & & & 2 & \\ & & & & 2 \end{pmatrix}, \quad T = \begin{pmatrix} \gamma_0 & 0 & \gamma_2 & 0 & \gamma_4 \\ & \gamma_0 & 0 & \gamma_2 & 0 \\ & & \gamma_0 & 0 & \gamma_2 \\ & & & \gamma_0 & 0 \\ & & & & \gamma_0 \end{pmatrix}, \quad H = \begin{pmatrix} \gamma_0 & \gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 \\ \gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 & \gamma_5 \\ \gamma_2 & \gamma_3 & \gamma_4 & \gamma_5 & \gamma_6 \\ \gamma_3 & \gamma_4 & \gamma_5 & \gamma_6 & \gamma_7 \\ \gamma_4 & \gamma_5 & \gamma_6 & \gamma_7 & \gamma_8 \end{pmatrix}$$

$$\gamma_k = \frac{\Gamma\left(\frac{k}{2}+\frac{1}{2}\right)}{\Gamma\left(\frac{k}{2}+1\right)}$$

Toeplitz matrix

Hankel matrix

Hadamard products and low-rank matrices

- A-dot-rank-1:

$$A \circ \underline{v}\underline{w}^T = \begin{pmatrix} a_{00}\, v_0 w_0 & a_{01}\, v_0 w_1 & a_{02}\, v_0 w_2 \\ a_{10}\, v_1 w_0 & a_{11}\, v_1 w_1 & a_{12}\, v_1 w_2 \\ a_{20}\, v_2 w_0 & a_{21}\, v_2 w_1 & a_{22}\, v_2 w_2 \end{pmatrix}$$

$$D_{\underline{v}} A D_{\underline{w}} = \begin{pmatrix} v_0 & & \\ & v_1 & \\ & & v_2 \end{pmatrix} \begin{pmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{pmatrix} \begin{pmatrix} w_0 & & \\ & w_1 & \\ & & w_2 \end{pmatrix}$$

- A-dot-rank-R: $A \circ \left(\sum_{k=1}^{R} \underline{v}_k \underline{w}_k^T \right) = \sum_{k=1}^{R} \left(A \circ \underline{v}_k \underline{w}_k^T \right) = \sum_{k=1}^{R} D_{\underline{v}_k} A D_{\underline{w}_k}$
- Toeplitz matrix can be applied in $\mathcal{O}(N \log(N))$ operations using Fast Fourier Transform (FFT)

- Toeplitz-dot-rank-R can be applied in $\mathcal{O}(RN \log(N))$ operations.

The algorithm

Input: $\underline{a}^{\mathrm{leg}} \in \mathbb{C}^{N+1}$

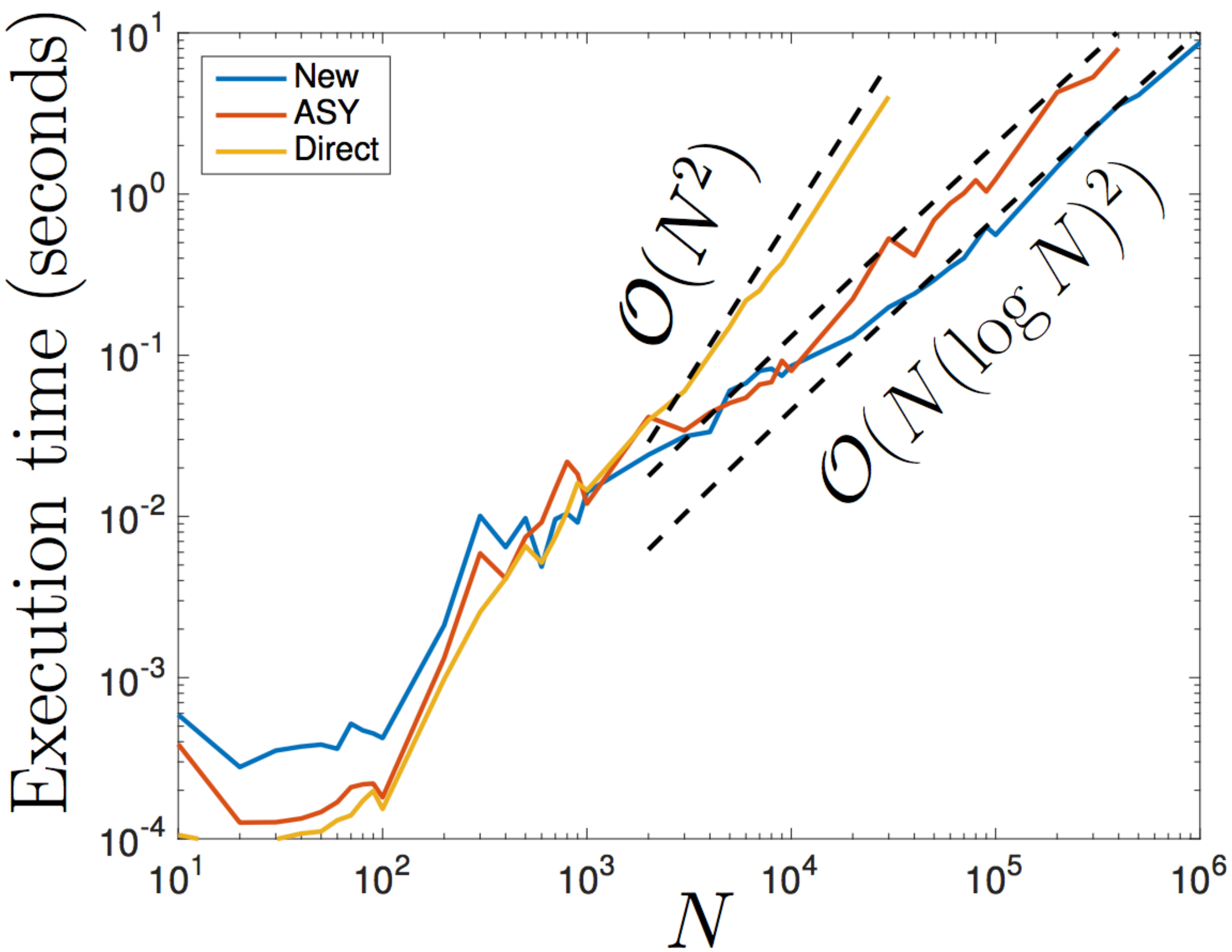
Output: $\underline{a}^{\mathrm{cheb}} = C\underline{a}^{\mathrm{leg}} = D(T \circ H)\underline{a}^{\mathrm{leg}}$

1. Compute the vector $\left(\frac{\Gamma(1/2)}{\Gamma(1)}, \frac{\Gamma(1)}{\Gamma(3/2)}, \frac{\Gamma(3/2)}{\Gamma(2)}, \frac{\Gamma(2)}{\Gamma(5/2)}, \ldots, \frac{\Gamma(N+1/2)}{\Gamma(N+1)}\right)$.
   - Use $\Gamma(z+1) = z\Gamma(z)$ to get $\mathcal{O}(N)$ operations (or asymptotics)
   - This vector implicitly defines H and T.
2. Compute the low-rank approximation $H = \sum_{k=1}^{R} \underline{v}_k \underline{v}_k^{\top}$.
   - Requires $\mathcal{O}(R^2 N)$ operations (see later)
3. Compute the matrix-vector product $\sum_{k=1}^{R} D_{\underline{v}_k} T D_{\underline{v}_k} \underline{a}^{\mathrm{leg}}$
   - Use the FFT to apply T in $\mathcal{O}(N \log N)$ operations.
4. Multiply by $D = \mathrm{diag}(\frac{1}{\pi}, \frac{2}{\pi}, \frac{2}{\pi}, \ldots, \frac{2}{\pi})$ in $\mathcal{O}(N)$ operations.

- Total operations: $\mathcal{O}(R^2 N + RN \log N) = \mathcal{O}(N(\log N)^2)$.

Comparison with state-of-the-art

- Only about **3-5 times faster** than Hale-Townsend 2013 asymptotics method
- We prove and observe **better error growth**
- New algorithm is **simpler** and can do **arbitrary precision** with little modification (BigFloat in Julia)



Low-rank matrix approximations

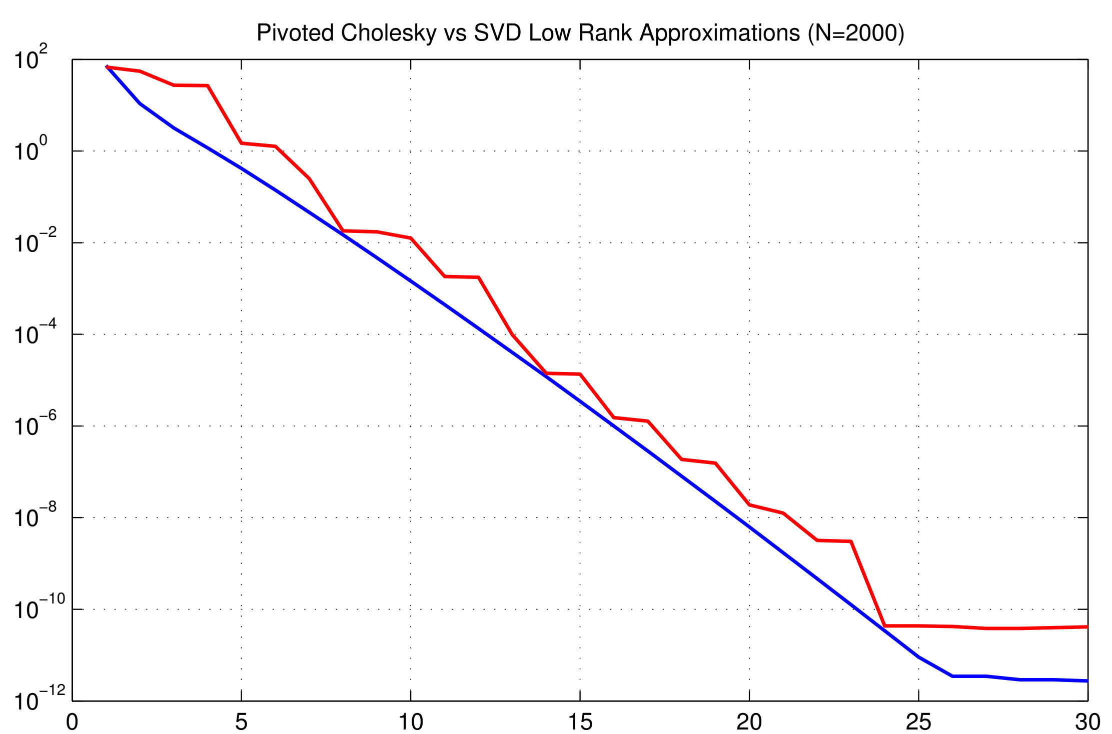
- Singular Value Decomposition (SVD) computes **optimal** low-rank approximations

$$A = \sum_{k=0}^{R} \sigma_k \underline{v}_k \underline{w}_k^T \qquad \sigma_k = \min_{\mathrm{rank}(B)=k} \|A - B\|_2 \qquad \mathcal{O}(N^3)$$

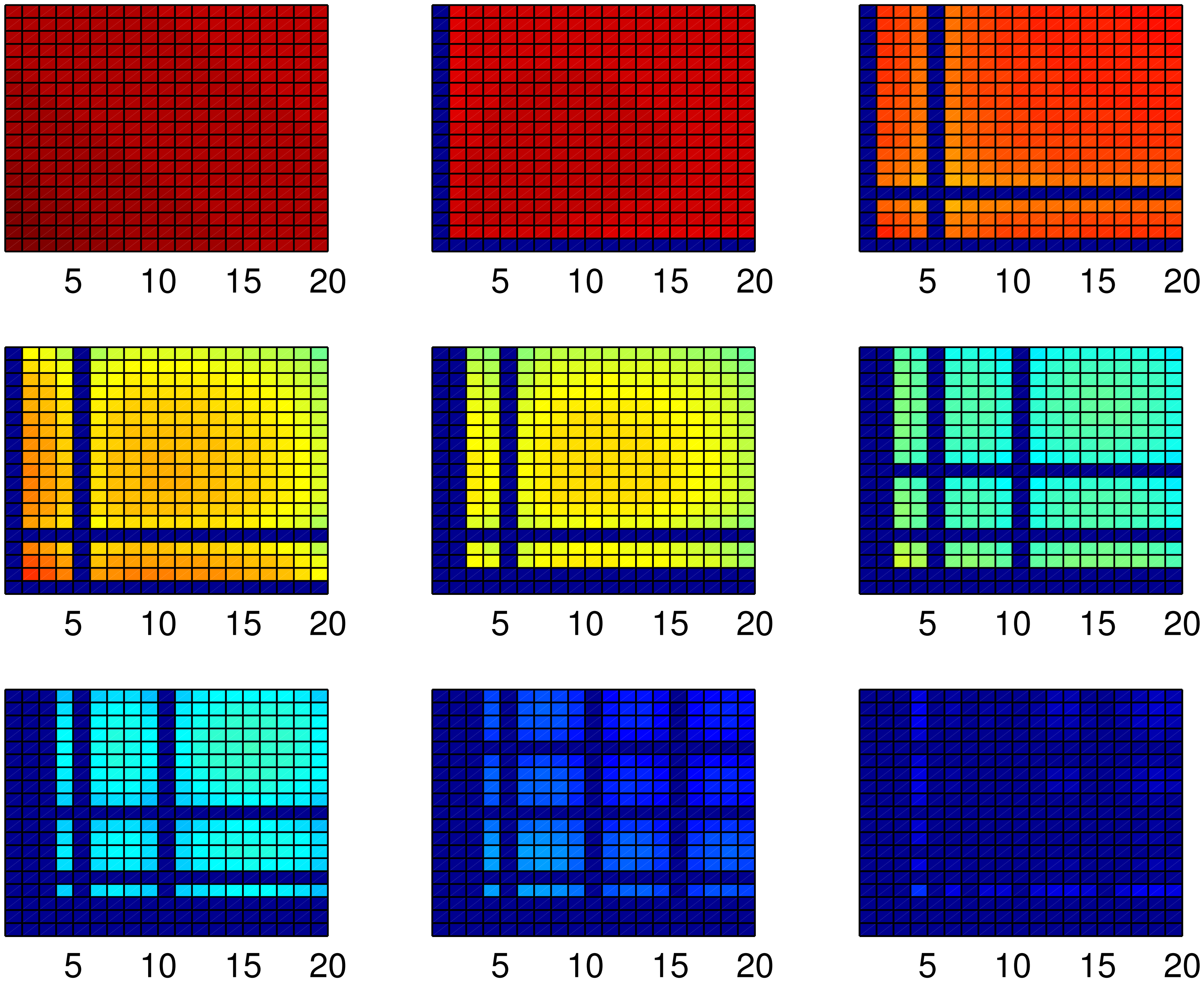
- Randomised SVD (Liberty et al. 2007) : for rank R, "$\mathcal{O}(N\log(N))$" matrices, it requires $\mathcal{O}(RN\log(N))$ operations.

- Cholesky decomposition with partial pivoting. (Harbrecht-Peters-Schilder 2011): for **symmetric positive-semi-definite,** rank R matrices, it requires $\mathcal{O}(R^2N)$ operations.

- For a rank $R = \mathcal{O}(\log(N))$ matrix, both require $\mathcal{O}(N(\log(N))^2)$ operations.



Low rank approximations to our Hankel matrix



Why is the Hankel matrix low rank?

- Technically not low rank. The singular values **decay exponentially** (Beckermann-Townsend 2016)

$$\sigma_{2k}(H_N) \leq c\rho^{-k/\log(N)}\|H_N\|_2 \qquad \mathrm{rank}_\varepsilon(H_N) = \mathcal{O}(\log(N)\log(\varepsilon^{-1}))$$

- Proof ideas: Positive semi-definite Hankel matrices can be written as a product of **Krylov matrices**

$$H = K^T K, \qquad K = \left(\underline{w}, A\underline{w}, A^2\underline{w}, \ldots, A^{N-1}\underline{w}\right)$$

- Krylov matrices have with **displacement structure**,

$$AK - KQ = \text{rank } 1 \qquad Q = \begin{pmatrix} 0 & & & -1 \\ 1 & & & \\ & \ddots & & \\ & & 1 & 0 \end{pmatrix}$$

- Ratio of singular values is bounded by a **rational Zolotarev problem**

$$\sigma_{j+k}(K) \leq Z_k(\sigma(A), \sigma(Q))\sigma_j(K)$$

Cheb-to-Leg matrix

$$c_{jk} = \frac{-k\left(j+\frac{1}{2}\right)}{4} \frac{\Gamma\left(\frac{k-j}{2}-\frac{1}{2}\right)}{\Gamma\left(\frac{k-j}{2}+1\right)} \frac{\Gamma\left(\frac{k+j}{2}\right)}{\Gamma\left(\frac{k+j}{2}+\frac{3}{2}\right)}$$

$$0 \le j \le N, \quad 1 \le k \le N, \quad j-k \text{ even}, \quad c_{00} = 1$$

- The situation is almost the same!

$$C = D_1(T \circ H)D_2$$

Ultraspherical-to-ultraspherical matrices

- Orthogonal w.r.t: $w(x) = (1 - x^2)^{\frac{1}{2}+\lambda}$

$$c_{jk} = \begin{cases} \omega_{\lambda_1,\lambda_2}(j+\lambda_2)\,\frac{\Gamma\left(\frac{k-j}{2}+\lambda_1-\lambda_2\right)}{\Gamma\left(\frac{k-j}{2}+1\right)} \cdot \frac{\Gamma\left(\frac{k+j}{2}+\lambda_1\right)}{\Gamma\left(\frac{k+j}{2}+\lambda_2+1\right)}, & 0 \leq j \leq k,\ k-j \text{ even}, \\ 0, & \text{otherwise.} \end{cases}$$

- Same situation. However, if $|\lambda_1 - \lambda_2| > 1$, then the Hankel matrix is not approx. low rank.

- We must perform several integer conversions, which takes $\mathcal{O}(N\lfloor|\lambda_1-\lambda_2|\rfloor)$ and then $|\tilde{\lambda}_1 - \tilde{\lambda}_2| < 1$

Jacobi-to-Jacobi matrix

- Orthogonal w.r.t: $w(x) = (1-x)^\alpha (1+x)^\beta$

- We do **not** have the diagonally scaled Toeplitz-dot-Hankel structure, but if we only convert in **one direction**, then we do:

$$c_{jk}^{(\alpha,\beta)\to(\gamma,\beta)} = \frac{(2+j+\gamma+\beta+1)}{\Gamma(\alpha-\gamma)} \frac{\Gamma(k+\beta+1)}{\Gamma(k+\alpha+\beta+1)} \frac{\Gamma(j+\gamma+\beta+1)}{\Gamma(j+\beta+1)} \cdot \frac{\Gamma(k-j+\alpha-\gamma)}{\Gamma(k-j+1)} \frac{\Gamma(k+j+\alpha+\beta+1)}{\Gamma(k+j+\gamma+\beta+2)}$$

$$c_{jk}^{(\gamma,\beta)\to(\gamma,\delta)} = (-1)^{k-j} c_{jk}^{(\beta,\gamma)\to(\delta,\gamma)}$$

$$C^{(\alpha,\beta)\to(\gamma,\delta)} = C^{(\alpha,\beta)\to(\gamma,\beta)} C^{(\gamma,\beta)\to(\gamma,\delta)}$$

Summary

- In Chebfun technology, it is sometimes necessary to change polynomial basis. E.g. sometimes Legendre better
- Connection coefficient matrix converts coefficients $\underline{a}^{\text{cheb}} = C\underline{a}^{\text{leg}}$
- For classical orthogonal polynomials they can be written as a diagonally scaled Hadamard product:

$C = D_1(T \circ H)D_2$ $\quad$ T Toeplitz $\quad$ H Hankel (approx. low rank)

- Fast-dot-low-rank matrices are also "fast" matrices. E.g. nonuniform FFT. **Any other matrices like this?**

Fast polynomial transforms based on Toeplitz and Hankel Matrices
Townsend A., Webb M., Olver S., to appear in Math. Comp.