UNIVERSITY OF MINNESOTA TWIN CITIES

Applications of trace estimation techniques

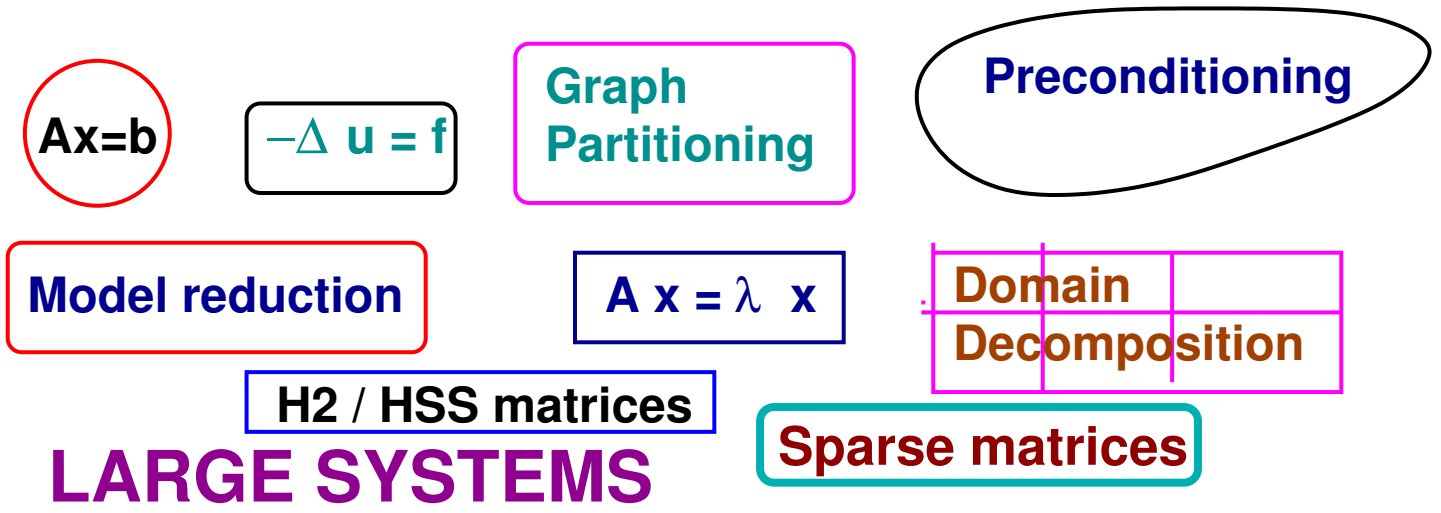
Yousef Saad

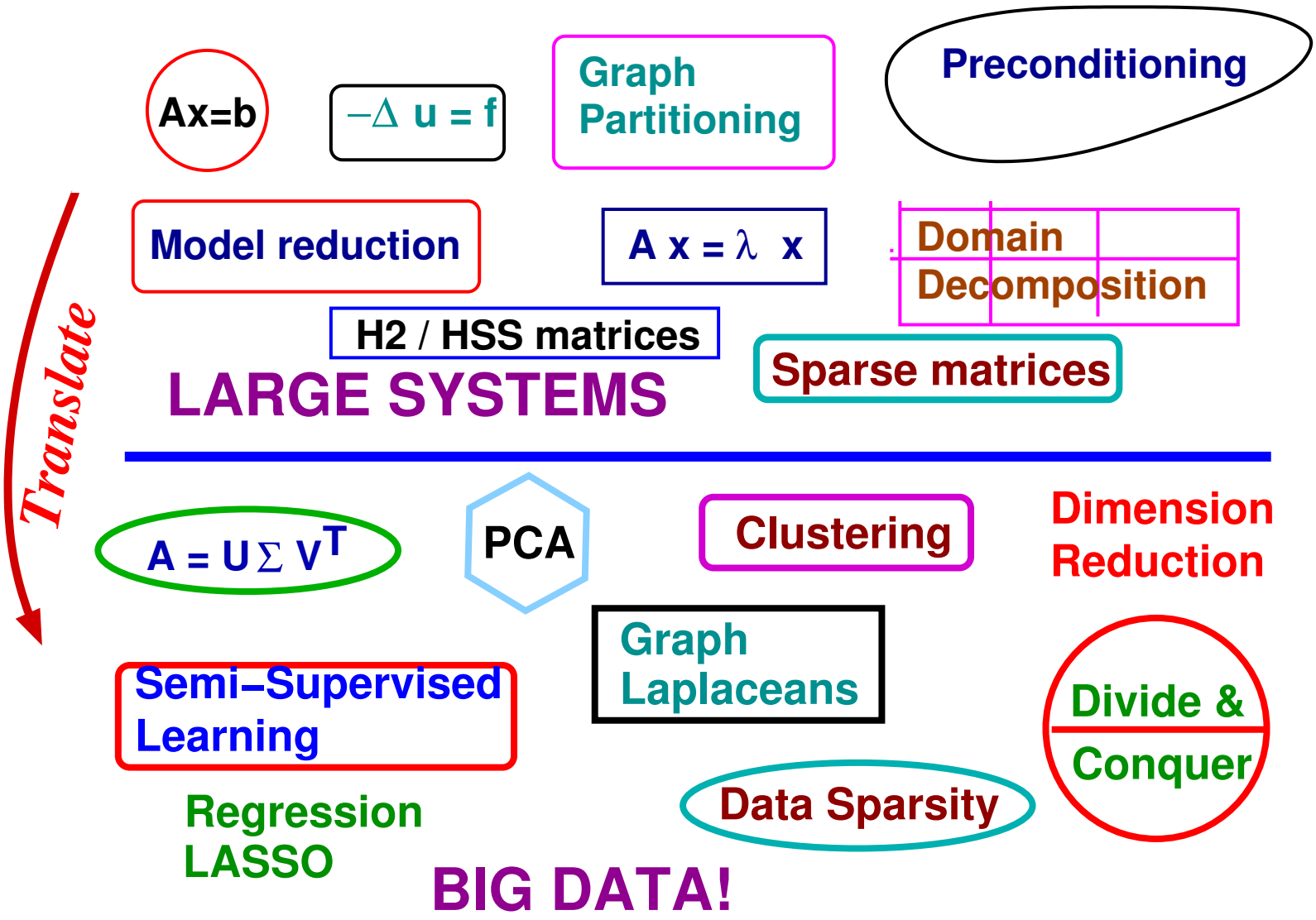
Department of Computer Science and Engineering

University of Minnesota

HPCSE 2017 - Solan, Czech republic

May 22-25, 2017


Ax=b
$-\Delta u = f$
Graph Partitioning
Preconditioning
Model reduction
$A x = \lambda x$
Domain Decomposition
H2 / HSS matrices
Sparse matrices
LARGE SYSTEMS


Ax=b
$-\Delta u = f$
Graph Partitioning
Preconditioning
Translate
Model reduction
$A x = \lambda x$
Domain Decomposition
H2 / HSS matrices
Sparse matrices
LARGE SYSTEMS
$A = U \Sigma V^T$
PCA
Clustering
Dimension Reduction
Graph Laplaceans
Semi–Supervised Learning
Divide & Conquer
Data Sparsity
Regression
LASSO
BIG DATA!

Introduction

- Focus of this talk: Spectral densities
- Spectral density == function that provides a global representation of the spectrum of a Hermitian matrix
- Known in solid state physics as 'Density of States' (DOS)
- Very useful in physics
- Almost unknown (as a tool) in numerical linear algebra

Outline:

1. general introduction, *2.* trace estimation, *3.* the DOS, *4.* how to compute it, *5.* how to use it (applications)

Introduction: A few examples

Problem 1: Compute Tr[inv[A]] the trace of the inverse.

- Arises in cross validation methods [Stats]
- Motivation for the work [Golub & Meurant, "Matrices, Moments, and Quadrature", 1993, Book with same title in 2009]

Problem 2: Compute Tr [f (A)], f a certain function

- Arises in many applications in Physics. Example:
- Stochastic estimations of Tr (f(A)) extensively used by quantum chemists to estimate Density of States, see

[H. Röder, R. N. Silver, D. A. Drabold, J. J. Dong, Phys. Rev. B. 55, 15382 (1997)]. Will be covered in detail later

Problem 3: Compute diag[inv(A)] the diagonal of the inverse

➤ Dynamic Mean Field Theory [DMFT, motivation for our work on this topic]. Related approach: Non Equilibrium Green's Function (NEGF) approach used to model nanoscale transistors.

➤ Uncertainty quantification: diagonal of the inverse of a covariance matrix needed [Bekas, Curioni, Fedulova '09]

Problem 4: Compute diag[f (A)] ; f = a certain function.

➤ Arises in density matrix approaches in quantum modeling

$$f(\epsilon) = \frac{1}{1 + \exp(\frac{\epsilon - \mu}{k_B T})}$$

Here, f = Fermi-Dirac operator
Note: when $T \rightarrow 0$ then $f \rightarrow$ a step function.

➤ Linear-Scaling methods

Problem 5: Estimate the numerical rank.

➤ Amounts to counting the number of singular values above a certain threshold τ == Trace $(\phi_\tau(A^T A))$..

$\phi_\tau(t)$ is a certain step function.

Problem 6: Estimate the log-determinant (common in statistics)

$$\log\det(A) = \text{Trace}(\log(A)) = \sum_{i=1}^{n} \log(\lambda_i).$$

.... many others

Important tool: Stochastic Estimator

➤ To estimate diagonal of $B = f(A)$ (e.g., $B = A^{-1}$), let:

- $d(B) = \text{diag}(B)$ [matlab notation]
- $\odot$ and $\oslash$: Elementwise multiplication and division of vectors
- $\{v_j\}$: Sequence of s random vectors

Notation:

Result:

$$d(B) \approx \left[\sum_{j=1}^{s} v_j \odot Bv_j\right] \oslash \left[\sum_{j=1}^{s} v_j \odot v_j\right]$$

C. Bekas , E. Kokiopoulou & YS ('05); C. Bekas, A. Curioni, I. Fedulova '09; ...

Trace of a matrix

- For the trace - take vectors of unit norm and

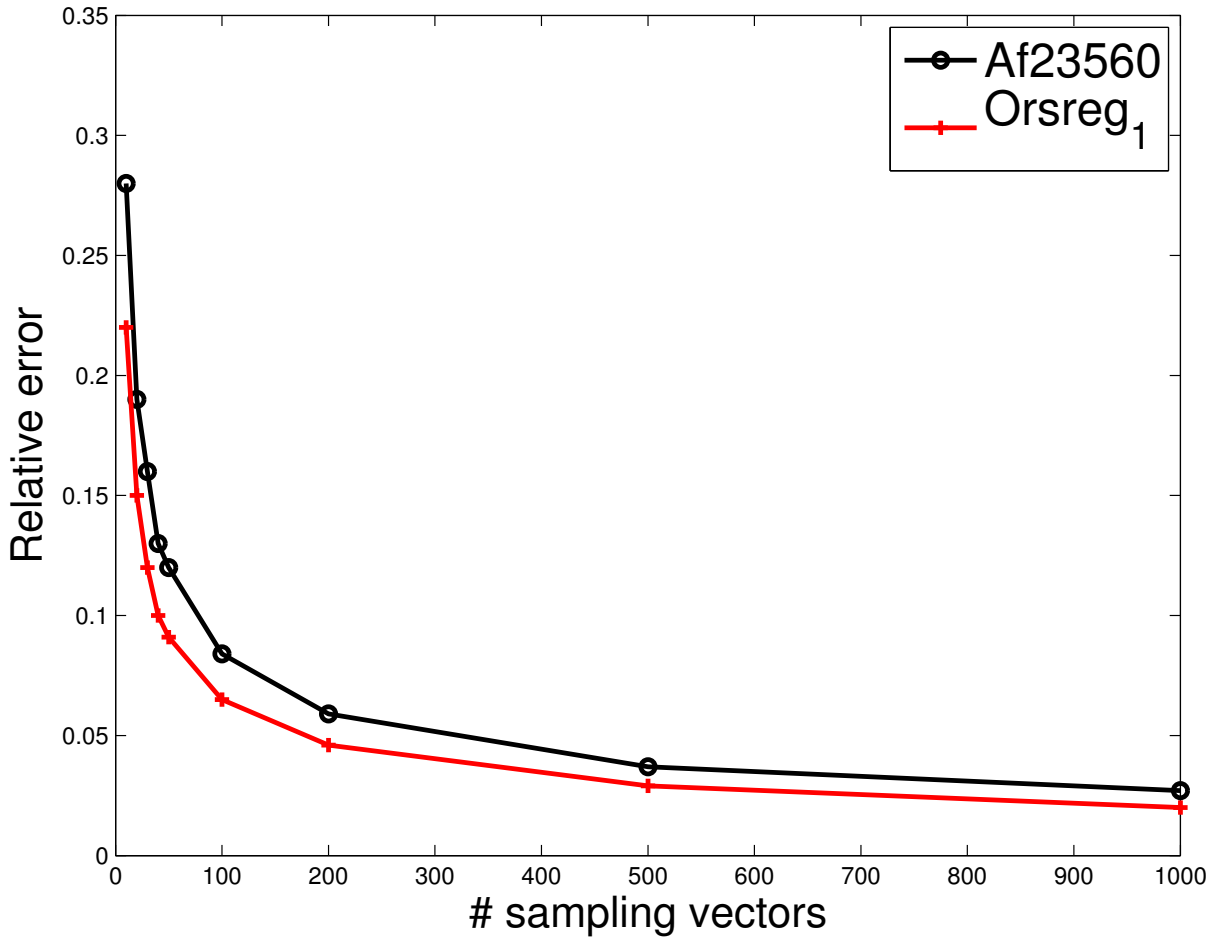
$$\text{Trace}(B) \approx \frac{1}{s} \sum_{j=1}^{s} v_j^T B v_j$$

- Hutchinson's estimator : take random vectors with components of the form $\pm 1/\sqrt{n}$ [Rademacher vectors]

- Extensively studied in literature. See e.g.: Hutchinson '89; H. Avron and S. Toledo '11; G.H. Golub & U. Von Matt '97; Roosta-Khorasani & U. Ascher '15; ...

Typical convergence curve for stochastic estimator

- Estimating the diagonal of inverse of two sample matrices



Alternative: standard probing

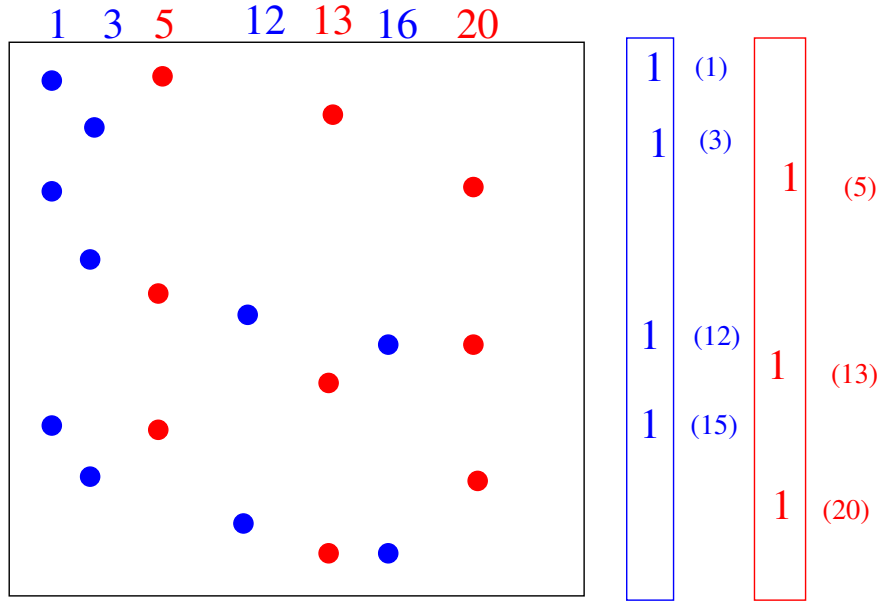
➤ Several names for same method: “probing”; “CPR”, “Sparse Jacobian estimators”,..

Basis of the method: Color columns of matrix so that no two columns of the same color overlap.

Entries of same color can be computed with 1 matvec

➤ Corresponds to coloring graph of $A^T A$.

➤ For problem of diag(A) need only color graph of A



In summary:

- Probing much more powerful when $f(A)$ is known to be nearly sparse (e.g. banded)..
- Approximate pattern (graph) can be obtained inexpensively
- Generally just a handful of probing vectors needed – Can be obtained by coloring graph
- However:
- Not as general: need $f(A)$ to be ' ϵ – sparse '

References:

- J. M. Tang and YS, *A probing method for computing the diagonal of a matrix inverse*, Numer. Lin. Alg. Appl., 19 (2012), pp. 485–501.

See also (improvements)

- Andreas Stathopoulos, Jesse Laeuchli, and Kostas Orginos *Hierarchical Probing for Estimating the Trace of the Matrix Inverse on Toroidal Lattices* SISC, 2012. [somewhat specific to Lattice QCD]

- E. Aune, D. P. Simpson, J. Eidsvik [Statistics and Computing 2012] combine probing with stochastic estimation. Good improvements reported.

DENSITY OF STATES & APPLICATIONS

Density of States

- Formally, the Density Of States (DOS) of a matrix A is

$$\phi(t) = \frac{1}{n} \sum_{j=1}^{n} \delta(t - \lambda_j),$$

where:
- δ is the Dirac δ-function or Dirac distribution
- $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ are the eigenvalues of A

- DOS is also referred to as the spectral density
- Note: number of eigenvalues in an interval $[a, b]$ is

$$\mu_{[a,b]} = \int_a^b \sum_j \delta(t - \lambda_j)\, dt \equiv \int_a^b n\phi(t) dt\,.$$

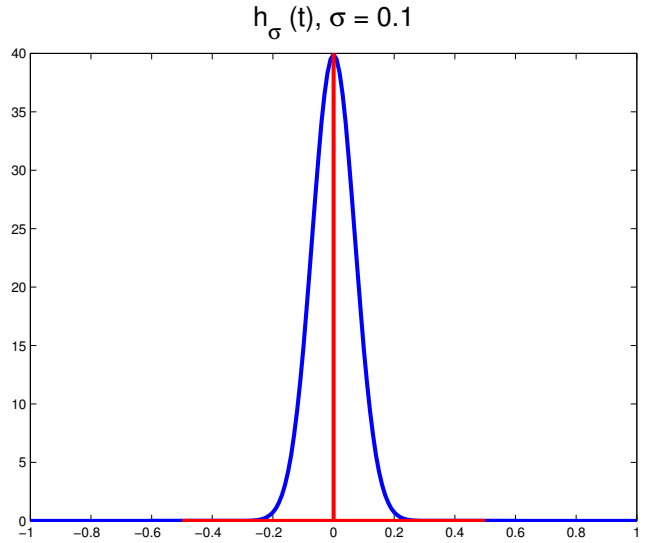
Issue: How to deal with distributions?

- Highly 'discontinuous', not easy to handle numerically
- Solution for practical and theoretical purposes: replace ϕ by a regularized ('blurred') version ϕ_σ:

$$\phi_\sigma(t) = \frac{1}{n} \sum_{j=1}^{n} h_\sigma(t - \lambda_j),$$

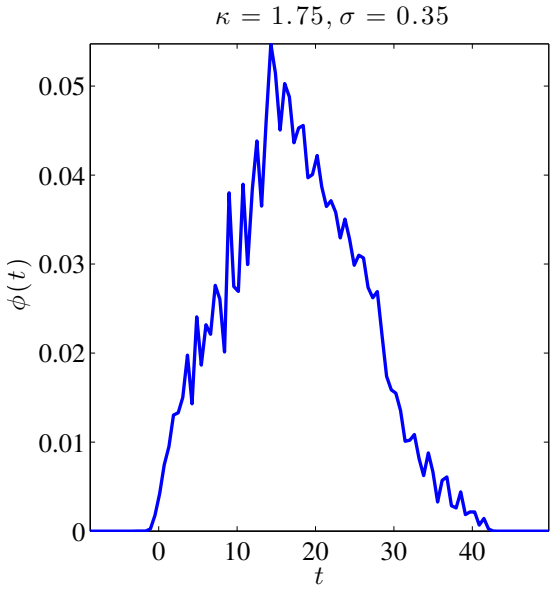
Where, for example:

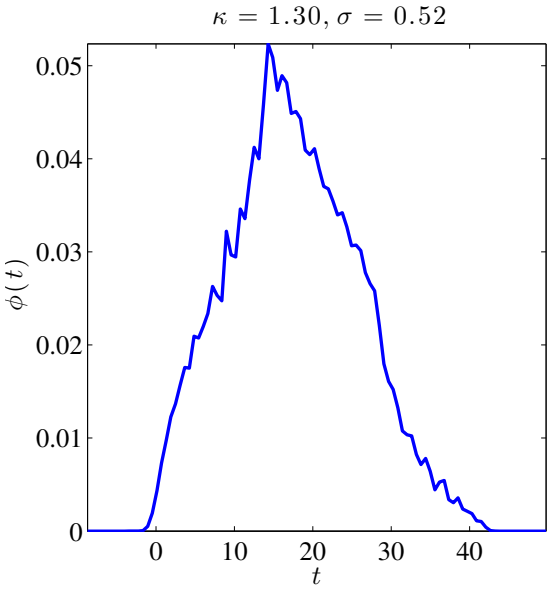
$$h_\sigma(t) = \frac{1}{(2\pi\sigma^2)^{1/2}} e^{-\frac{t^2}{2\sigma^2}}.$$

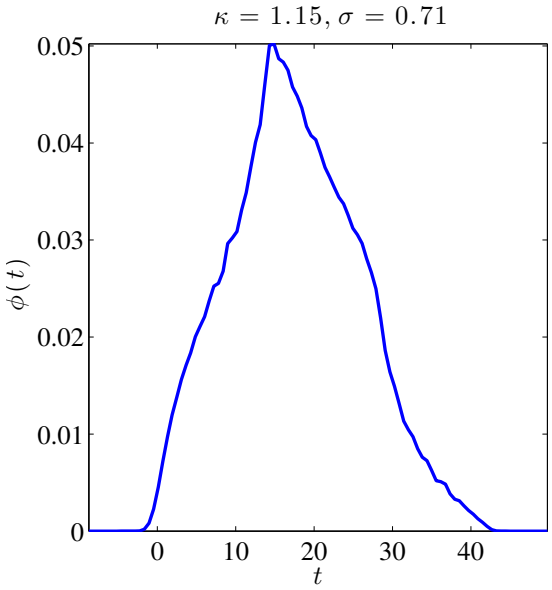


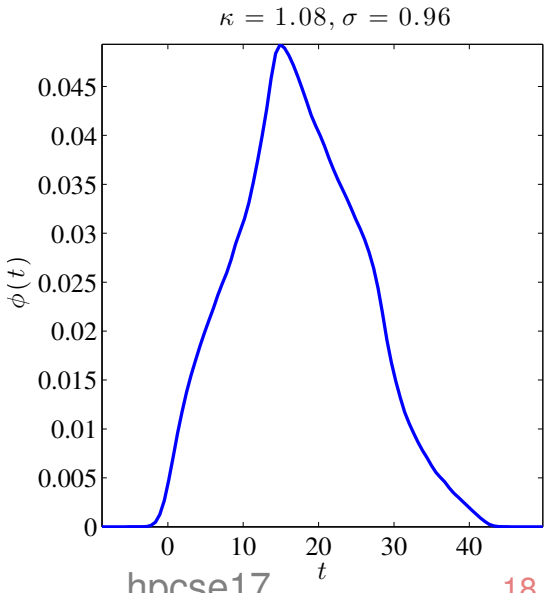
- Smoothed $\phi(t)$ can be viewed as a distribution function == probability of finding eigenvalues of A in a given infinitesimal interval near t.
- In Solid-State physics, λ_i's represent single-particle energy levels.
- So the DOS represents # of levels per unit energy.
- Many uses in physics

- How to select smoothing parameter σ? Example for Si_2









- Higher $\sigma \rightarrow$ smoother curve
- But loss of detail ..
- Compromise: $\sigma = \frac{h}{2\sqrt{2\log(\kappa)}}$,
- $h =$ resolution, $\kappa =$ parameter > 1

Computing the DOS: The Kernel Polynomial Method

- Used by Chemists to calculate the DOS – see Silver and Röder'94 , Wang '94, Drabold-Sankey'93, + others
- Basic idea: expand DOS into Chebyshev polynomials
- Use trace estimator [discovered independently] to get traces needed in calculations
- Assume change of variable done so eigenvalues lie in $[-1,\ 1]$.
- Include the weight function in the expansion so expand:

$$\hat{\phi}(t) = \sqrt{1-t^2}\phi(t) = \sqrt{1-t^2} \times \frac{1}{n}\sum_{j=1}^{n} \delta(t-\lambda_j).$$

Then, (full) expansion is: $\hat{\phi}(t) = \sum_{k=0}^{\infty} \mu_k T_k(t).$

- Expansion coefficients μ_k are formally defined by:

$$\begin{aligned}\mu_k &= \frac{2-\delta_{k0}}{\pi}\int_{-1}^{1}\frac{1}{\sqrt{1-t^2}}T_k(t)\hat{\phi}(t)dt \\ &= \frac{2-\delta_{k0}}{\pi}\int_{-1}^{1}\frac{1}{\sqrt{1-t^2}}T_k(t)\sqrt{1-t^2}\phi(t)dt \\ &= \frac{2-\delta_{k0}}{n\pi}\sum_{j=1}^{n}T_k(\lambda_j).\end{aligned}$$

- Here $2-\delta_{k0}$ == 1 when $k=0$ and == 2 otherwise.
- Note: $\sum T_k(\lambda_i) = \mathit{Trace}[T_k(A)]$
- Estimate this, e.g., via stochastic estimator
- Generate random vectors $v^{(1)}, v^{(2)}, \cdots, v^{(n_{\text{vec}})}$
- Assume normal distribution with zero mean

- Each vector is normalized so that $\|v^{(l)}\| = 1, l = 1, \ldots, n_{\text{vec}}$.
- Estimate the trace of $T_k(A)$ with stochastisc estimator:

$$\text{Trace}(T_k(A)) \approx \frac{1}{n_{\text{vec}}} \sum_{l=1}^{n_{\text{vec}}} \left(v^{(l)}\right)^T T_k(A) v^{(l)}.$$

- Will lead to the desired estimate:

$$\mu_k \approx \frac{2 - \delta_{k0}}{n\pi n_{\text{vec}}} \sum_{l=1}^{n_{\text{vec}}} \left(v^{(l)}\right)^T T_k(A) v^{(l)}.$$

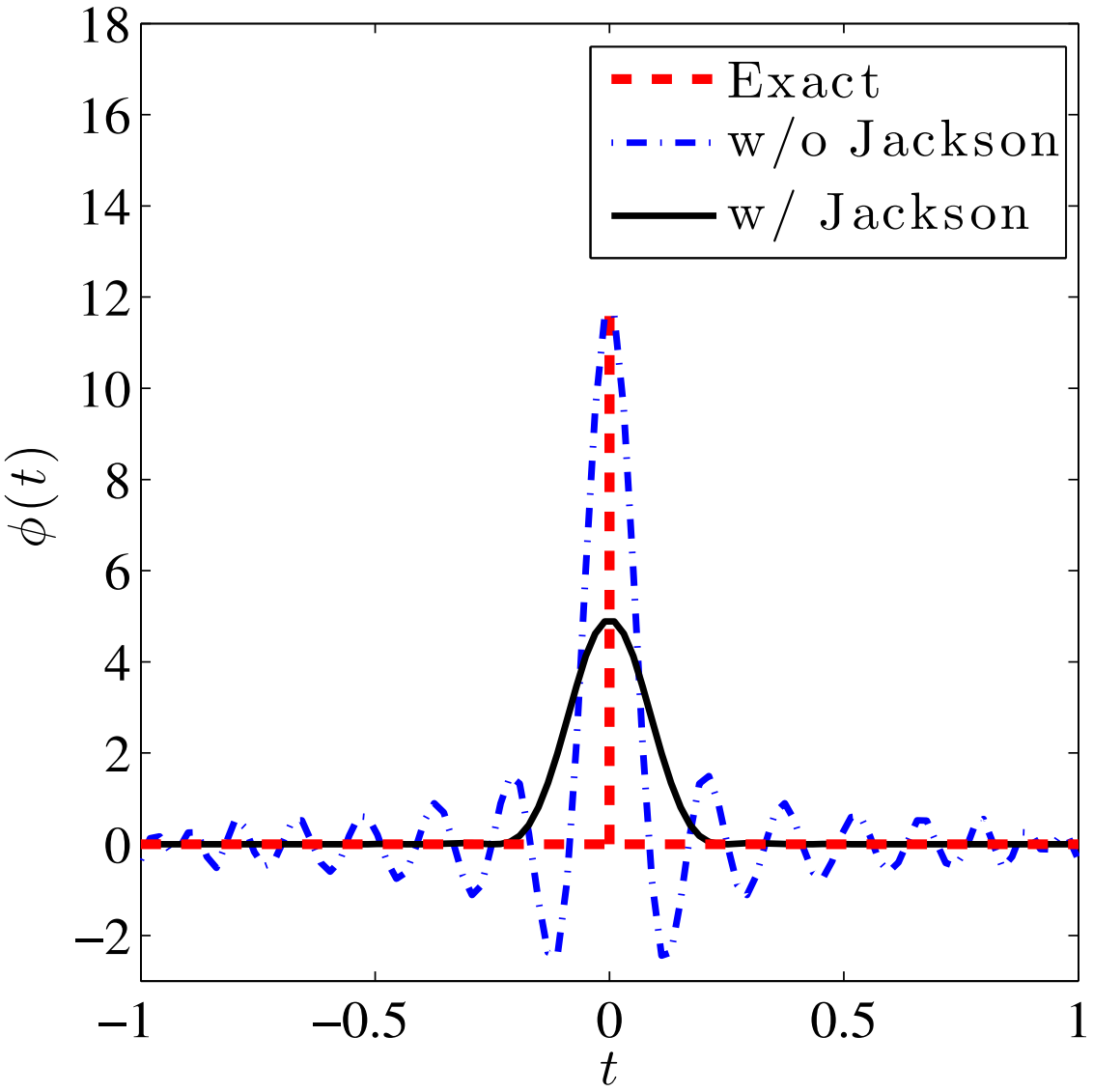
- To compute scalars of the form $v^T T_k(A) v$, exploit 3-term recurrence of the Chebyshev polynomial:

$$T_{k+1}(A)v = 2AT_k(A)v - T_{k-1}(A)v$$

so if we let $v_k \equiv T_k(A)v$, we have

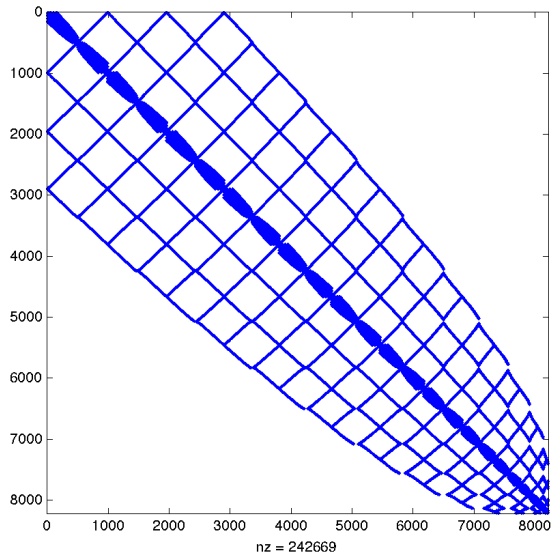
$$v_{k+1} = 2Av_k - v_{k-1}$$

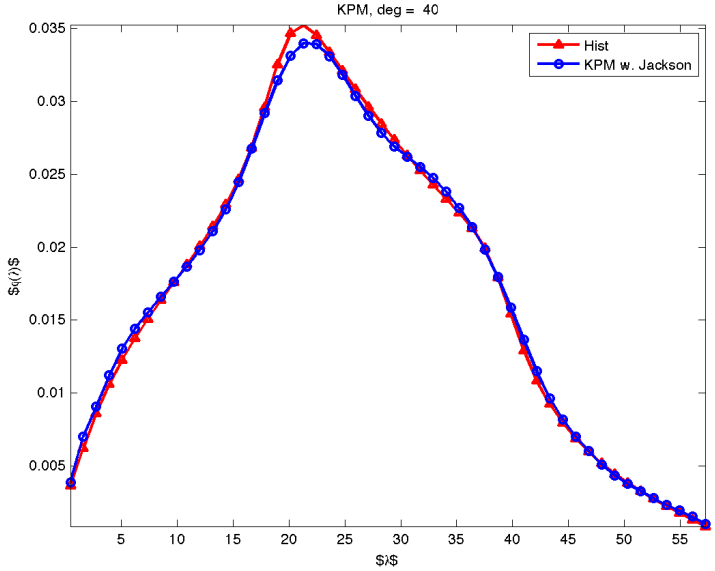
- Jackson smoothing can be used –



An example: The Benzene matrix

```
>> TestKpmDos
 Matrix Benzene n =8219  nnz = 242669
Degree =  40  # sample vectors =  10
Elapsed time is 0.235189 seconds.
```





Use of the Lanczos Algorithm

➤ Background: The Lanczos algorithm generates an orthonormal basis $V_m = [v_1, v_2, \cdots, v_m]$ for the Krylov subspace:

$$\textbf{span}\{v_1, Av_1, \cdots, A^{m-1}v_1\}$$

➤ ... such that:
$V_m^H A V_m = T_m$ - with

$$T_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \beta_3 & \alpha_3 & \beta_4 & \\ & & \cdot & \cdot & \cdot \\ & & & \cdot & \cdot & \cdot \\ & & & & \beta_m & \alpha_m \end{pmatrix}$$

- Lanczos process builds orthogonal polynomials wrt to dot product:

$$\int p(t)q(t)dt \equiv (p(A)v_1, q(A)v_1)$$

- In theory v_i's defined by 3-term recurrence are orthogonal.
- Let $\theta_i,\ i = 1 \cdots, m$ be the eigenvalues of T_m [Ritz values]
- y_i's associated eigenvectors; Ritz vectors: $\{V_m y_i\}_{i=1:m}$
- Ritz values approximate eigenvalues
- Could compute θ_i's then get approximate DOS from these
- Problem: θ_i not good enough approximations – especially inside the spectrum.

➤ Better idea: exploit relation of Lanczos with (discrete) orthogonal polynomials and related Gaussian quadrature:

$$\int p(t)dt \approx \sum_{i=1}^{m} a_i p(\theta_i) \quad a_i = \left[e_1^T y_i\right]^2$$

➤ See, e.g., Golub & Meurant '93, and also Gautschi'81, Golub and Welsch '69.

➤ Formula exact when p is a polynomial of degree $\leq 2m+1$

- Consider now $\int p(t)dt =< p, 1 >=$ (Stieljes) integral $\equiv$

$$(p(A)v, v) = \sum \beta_i^2 p(\lambda_i) \equiv < \phi_v, p >$$

- Then $\langle \phi_v, p \rangle \approx \sum a_i p(\theta_i) = \sum a_i \langle \delta_{\theta_i}, p \rangle \rightarrow$

$$\phi_v \approx \sum a_i \delta_{\theta_i}$$

- To mimick the effect of $\beta_i = 1, \forall i$, use several vectors v and average the result of the above formula over them..

Other methods

➤ The Lanczos spectroscopic approach : A sort of signal processing approach to detect peaks using Fourier analysis

➤ The Delta-Chebyshev approach: Smooth ϕ with Gaussians, then expand Gaussians using Legendre polynomials

➤ Haydock's method: interesting 'classic' approach in physics - uses Lanczos to unravel 'near-poles' of $(A - \epsilon i I)^{-1}$

For details see:

- Approximating spectral densities of large matrices, Lin Lin, YS, and Chao Yang - SIAM Review '16. Also in: [arXiv: http://arxiv.org/abs/1308.5467]

Experiments

- Goal: to compare errors for similar number of matrix-vector products
- Example: Kohn-Sham Hamiltonian associated with a benzene molecule generated from PARSEC. $n = 8,219$
- In all cases, we use 10 sampling vectors
- General observation: DGL, Lanczos, and KPM are best,
- Spectroscopic method does OK
- Haydock's method [another method based on the Lanczos algorithm] not as good

Method	L^1 error	L^2 error	L^∞ error
KPM w/ Jackson, deg=80	2.592e-02	5.032e-03	2.785e-03
KPM w/o Jackson, deg=80	2.634e-02	4.454e-03	2.002e-03
KPM Legendre, deg=80	2.504e-02	3.788e-03	1.174e-03
Spectroscopic, deg=40	5.589e-02	8.652e-03	2.871e-03
Spectroscopic, deg=100	4.624e-02	7.582e-03	2.447e-03
DGL, deg=80	1.998e-02	3.379e-03	1.149e-03
Lanczos, deg=80	2.755e-02	4.178e-03	1.599e-03
Haydock, deg=40	6.951e-01	1.302e-01	6.176e-02
Haydock, deg=100	2.581e-01	4.653e-02	1.420e-02

L^1, L^2, and L^∞ error compared with the normalized "surrogate" DOS for benzene matrix

➤ Many more experiments in survey paper [L. Lin, YS, C. Yang, SIAM Review, 2015].

What about matrix pencils?

➤ DOS for generalized eigenvalue problems

$$Ax = \lambda Bx$$

➤ Assume: A is symmetric and B is SPD.

➤ In principle: can just apply methods to $B^{-1}Ax = \lambda x$, using B - inner products.

➤ Requires factoring B. Too expensive [Think 3D Pbs]

⋆ *Observe:* B is usually very *strongly* diagonally dominant.

➤ Especially true after Left+Right Diag. scaling :

$$\tilde{B} = S^{-1}BS^{-1} \qquad S = diag(B)^{1/2}$$

➤ General observation for FEM mass matrices. See Theorem 3.2 in L. Kamenski, W. Huang, and H. Xu, Math. Comp.'14:

Theorem Condition number of scaled Galerkin mass matrix with a simplicial mesh has a mesh-independent bound:

$$\kappa(S^{-1}BS^{-1}) \leq d + 2$$

Example: Matrix pair Kuu, Muu from Suite Sparse collection.

➤ Matrices A and B have dimension $n = 7,102$. $\texttt{nnz}(A) = 340,200$ $\texttt{nnz}(B) = 170,134$.

➤ After scaling by diagonals to have diag. entries equal to one, all eigenvalues of B are in interval

$$[0.6254,\ 1.5899]$$

Approximation theory to the rescue.

⋆ *Idea:* Compute the DOS for the standard problem

$$B^{-1/2}AB^{-1/2}u = \lambda u$$

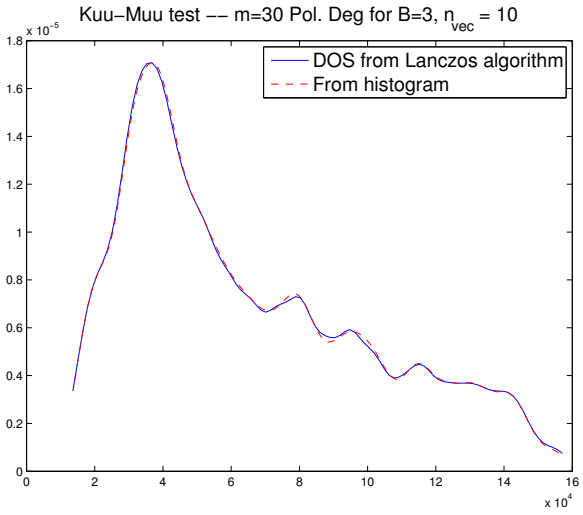
- Use a very low degree polynomial to approximate $B^{-1/2}$.
- We use Chebyshev expansions.
- Degree k determined automatically by enforcing

$$\|t^{-1/2} - p_k(t)\|_\infty < tol$$

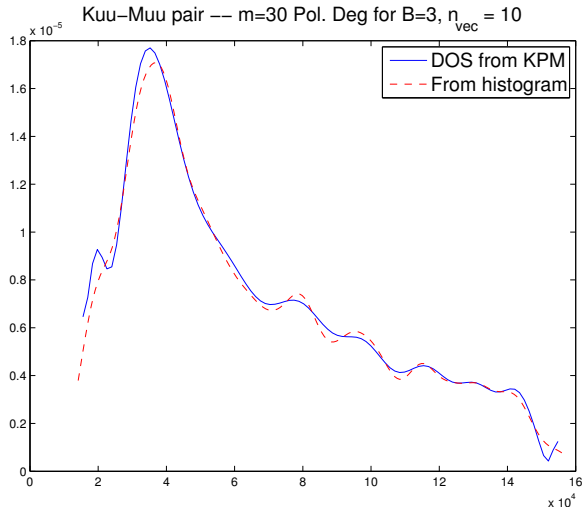
- Theoretical results establish convergence that is exponential with respect to degree.

Example: Results for Kuu-Muu example

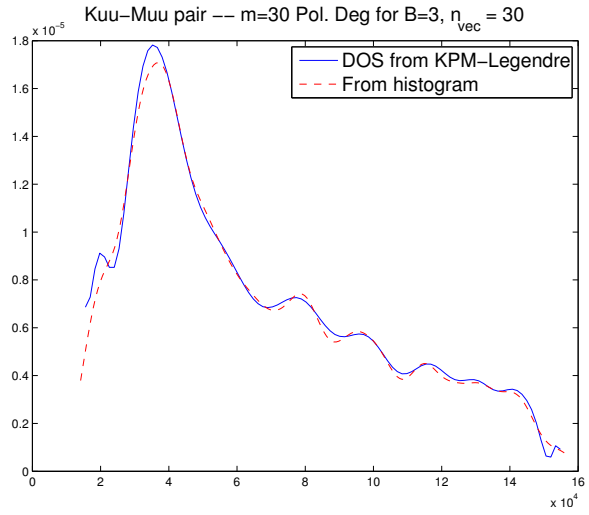
- Using polynomials of degree 3 (!) to approximate $B^{-1/2}$
- Krylov subspace of dim. 30 (== deg. of polynomial in KPM)
- 10 Sample vectors used



Lanczos



KPM-Chebyshev



KPM-Legendre

Application 1: Eigenvalue counts

The problem: Given A (Hermitian) with eigenvalues $\lambda_1 \leq \lambda_2 \cdots \leq \lambda_n$ find an estimate of the number $\mu_{[a,b]}$ of eigenvalues of A in interval $[a, \quad b]$.

Standard method: Sylvester inertia theorem. Requires two LDL^T factorizations $\rightarrow$ expensive!

First alternative: integrate the Spectral Density in $[a, \ b]$.

$$\mu_{[a,b]} \approx n \left(\int_a^b \tilde{\phi}(t) dt \right) = n \sum_{k=0}^{m} \mu_k \left(\int_a^b \frac{T_k(t)}{\sqrt{1-t^2}} dt \right) = \ldots$$

Second method: Estimate trace of the related spectral projector P ($\rightarrow$ u_i's = eigenvectors $\leftrightarrow$ λ_i's)

$$P = \sum_{\lambda_i \in [a \ b]} u_i u_i^T.$$

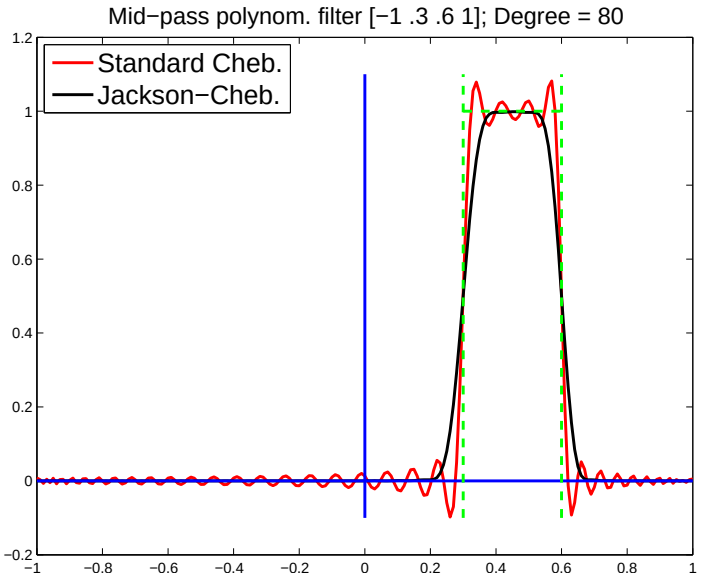
➤ We know: $\mu_{[a,b]} = Tr(P)$. P is not available ... but can be approximated. Note:

$$P = h(A) \quad \text{where} \quad h(t) = \begin{cases} 1 & \text{if } t \in [a\ b] \\ 0 & \text{otherwise} \end{cases}$$

➤ Approximate $h(t)$ by polynom. $\psi(t)$ using Chebyshev expansions

➤ Then $\mu_{[a,b]} \approx \text{Tr}\,(\psi(A))$ approximated by a trace estimator:

$$\mu_{[a,b]} \approx \frac{1}{n_v} \sum_{k=1}^{n_v} v_k^\top \psi(A) v_k$$



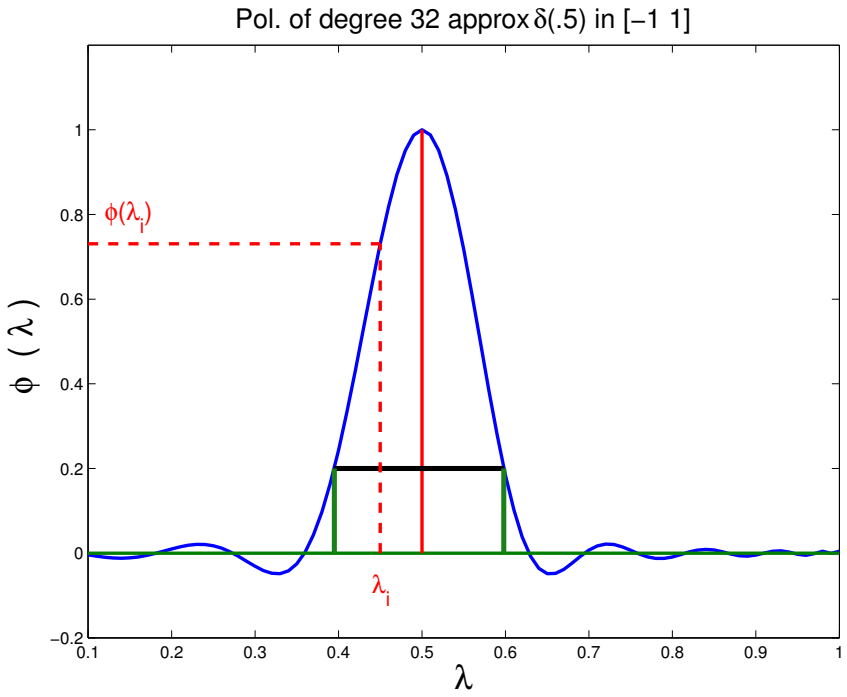
➤ It turns out that the 2 methods are identical.

Application 2: "Spectrum Slicing"

- Situation: very large number of eigenvalues to be computed
- Goal: compute spectrum by slices by applying filtering
- Apply Lanczos or Sub-space iteration to problem:

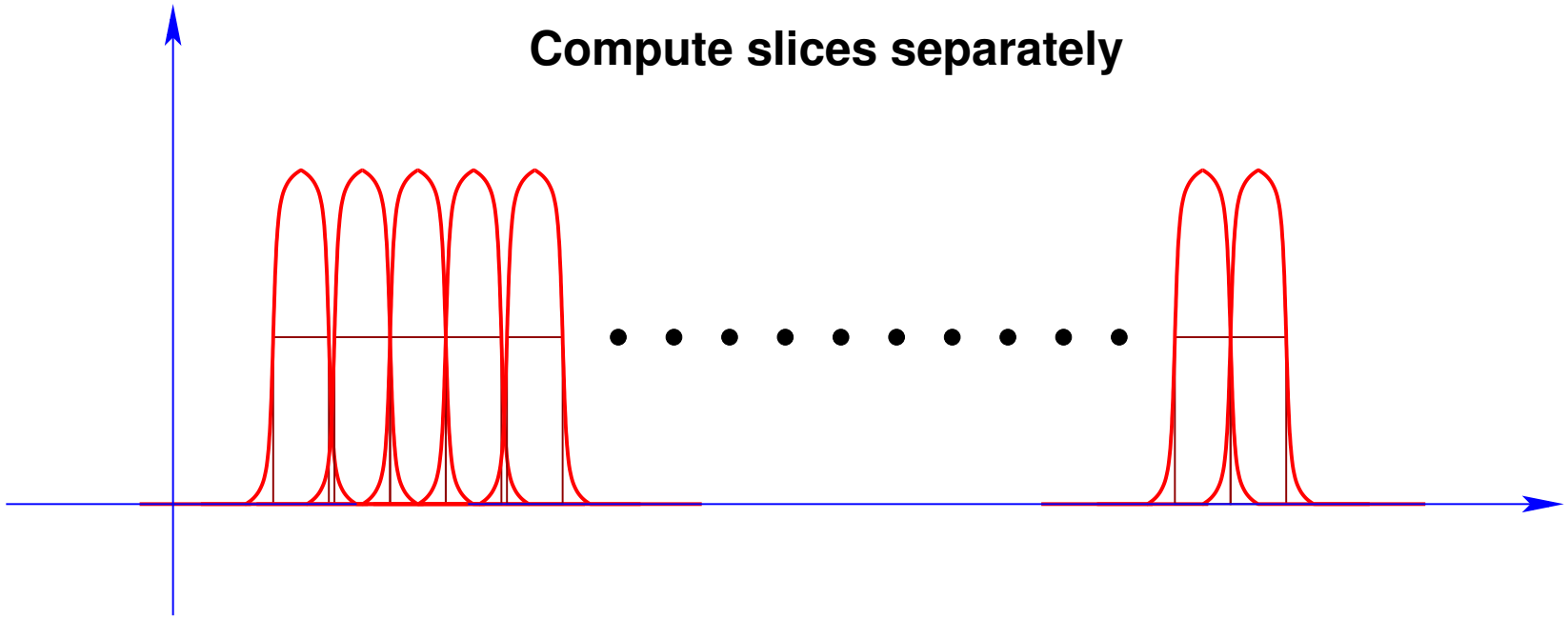
$$\phi(A)u = \mu u$$

$\phi(t)$ $\equiv$ a polynomial or rational function that enhances wanted eigenvalues



Rationale. Eigenvectors on both ends of wanted spectrum need not be orthogonalized against each other :

- Idea: Get the spectrum by ‘slices’ or ’windows’ [e.g., a few hundreds or thousands of pairs at a time]
- Can use polynomial or rational filters



- Deceivingly simple looking idea.
- Issues:

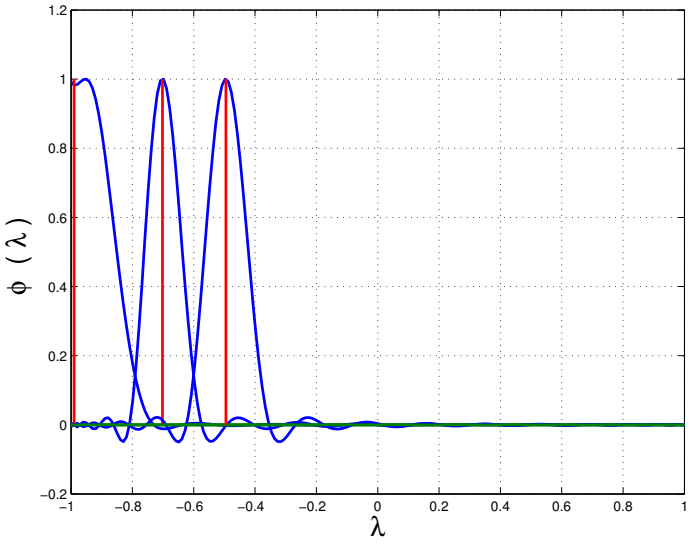
- Deal with interfaces : duplicate/missing eigenvalues
- Window size [need estimate of eigenvalues]
- How to compute each slice? [polynomial / rational filters?, ..]

A digression: the EVSL project

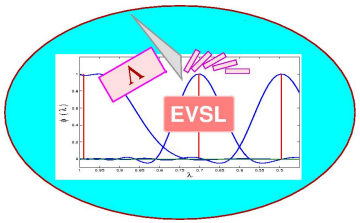
- Newly released EVSL uses polynomial and rational filters
- Each can be appealing in different situations.

Spectrum slicing: cut the overall interval containing the spectrum into small sub-intervals and compute eigenpairs in each sub-interval independently.

For each subinterval: select a filter polynomial of a certain degree so its high part captures the wanted eigenvalues. In illustration, the polynomials are of degree 20 (left), 30 (middle), and 32 (right).



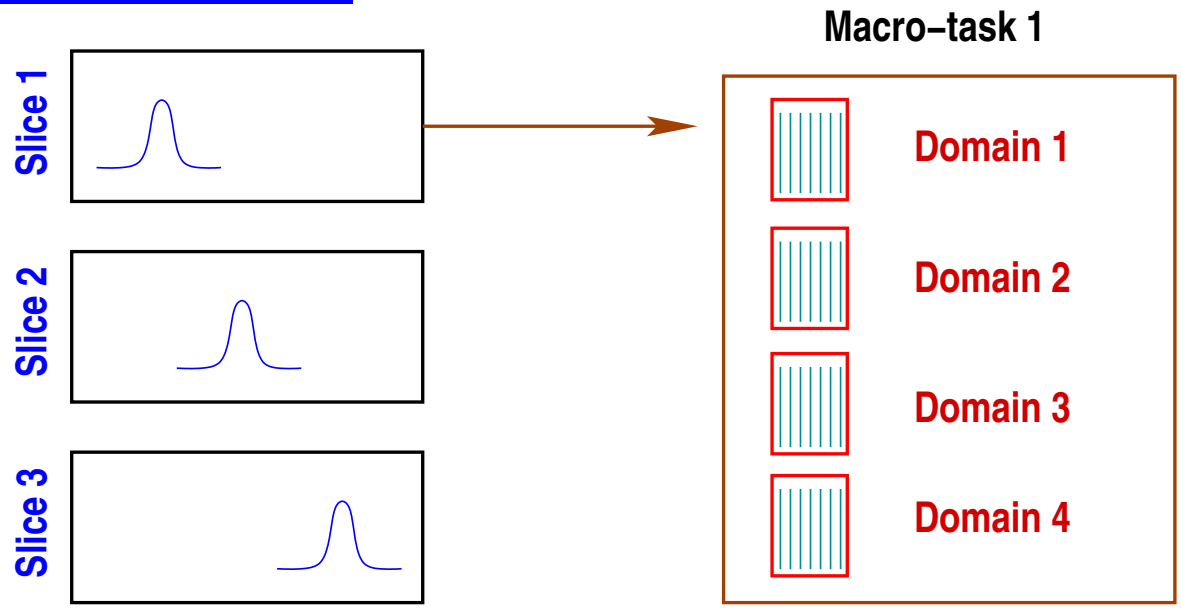
S O F T W A R E



EVSL NEW! a library of (sequential) eigensolvers based on spectrum slicing. **Version 1.0** released on [09/11/2016]
EVSL provides routines for computing eigenvalues located in a given interval, and their associated eigenvectors, of real symmetric matrices. It also provides tools for spectrum slicing, i.e., the technique of subdividing a given interval into p smaller subintervals and computing the eigenvalues in each subinterval independently. EVSL implements a polynomial filtered Lanczos algorithm (thick restart, no restart) a rational filtered Lanczos algorithm (thick restart, no restart), and a polynomial filtered subspace iteration.

ITSOL a library of (sequential) iterative solvers. **Version 2** released. [11/16/2010]
ITSOL can be viewed as an extension of the ITSOL module in the SPARSKIT package. It is written in C and aims at providing additional preconditioners for solving general sparse linear systems of equations. Preconditioners so far in this package include (1) ILUK (ILU preconditioner with level of fill) (2) ILUT (ILU preconditioner with threshold) (3) ILUC (Crout version of ILUT) (4) VBILUK (variable block preconditioner with level of fill - with automatic block detection) (5) VBILUT (variable block preconditioner with threshold - with automatic block detection) (6) ARMS (Algebraic Recursive Multilevel Solvers -- includes actually several methods - In particular the standard ARMS and the ddPQ version which uses nonsymmetric permutations).
ZITSOL a complex version of some of the methods in ITSOL is also available.

Levels of parallelism



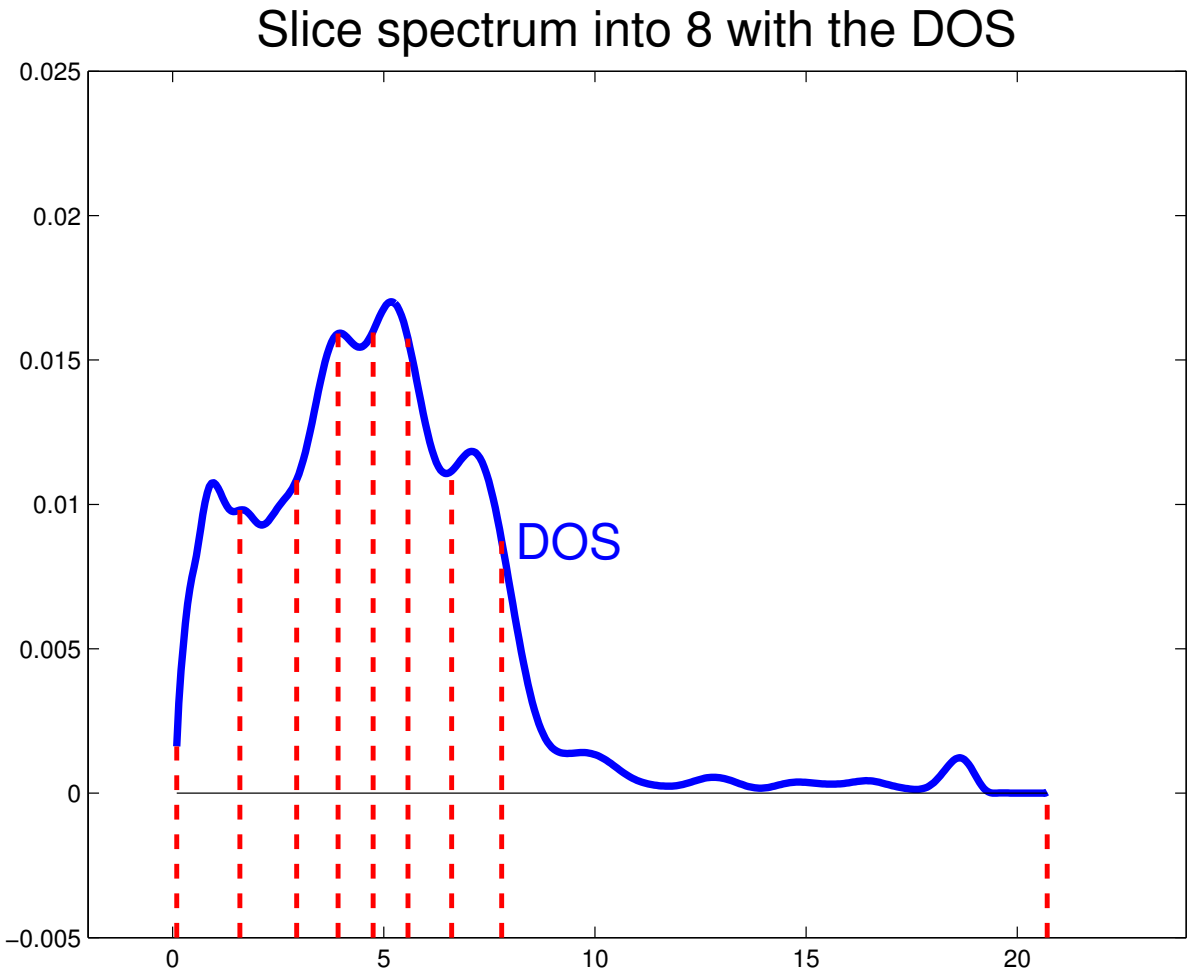
The two main levels of parallelism in EVSL

How do I slice a spectrum?

Analogue question:

How would I slice an onion if I want each slice to have about the same mass?

Answer: Use the DOS.



- We must have:

$$\int_{t_i}^{t_{i+1}} \phi(t)dt = \frac{1}{n_{slices}} \int_a^b \phi(t)dt$$

Application 3: Estimating the rank

- Joint work with S. Ubaru

➤ Very important problem in signal processing applications, machine learning, etc.

➤ Often: a certain rank is selected ad-hoc. Dimension reduction is application with this "guessed" rank.

➤ Can be viewed as a particular case of the eigenvalue count problem - but need a cutoff value..

Approximate rank, Numerical rank

- Notion defined in various ways. A common one:

$$r_\epsilon = \min\{rank(B) : B \in \mathbb{R}^{m\times n}, \|A - B\|_2 \leq \epsilon\},$$

$$r_\epsilon = \text{ Number of sing. values } \geq \epsilon$$

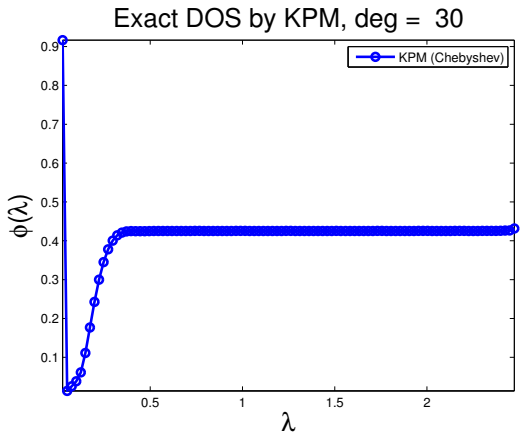
- Two distinct problems:

1. Get a good ϵ 2. Estimate number of sing. values $\geq \epsilon$

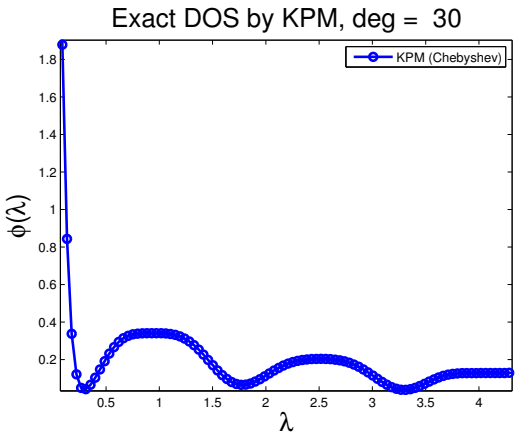
- We will need a cut-off value ('threshold') ϵ.
- Could use 'noise level' for ϵ, but not always available

Threshold selection

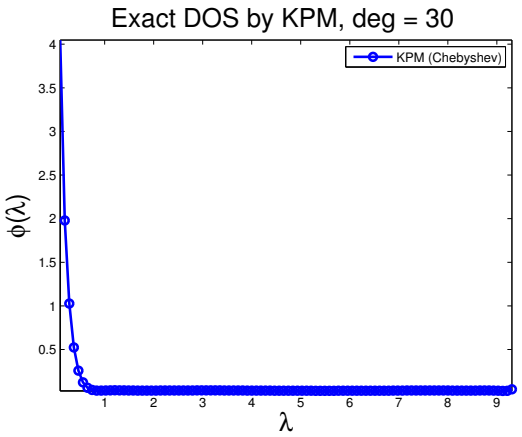
- How to select a good threshold?
- Answer: Obtain it from the DOS function



(A)



(B)



(C)

Exact DOS plots for three different types of matrices.

➤ To find: point immediatly following the initial sharp drop observed.

➤ Simple idea: use derivative of DOS function ϕ

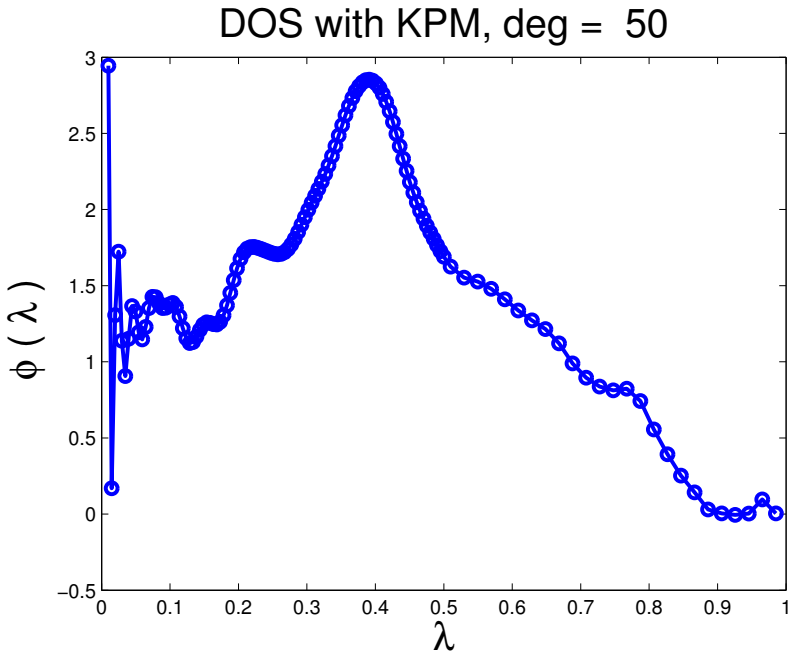
➤ For an $n \times n$ matrix with eigenvalues $\lambda_n \leq \lambda_{n-1} \leq \cdots \leq \lambda_1$:

$$\epsilon = \min\{t : \lambda_n \leq t \leq \lambda_1, \phi'(t) = 0\}.$$

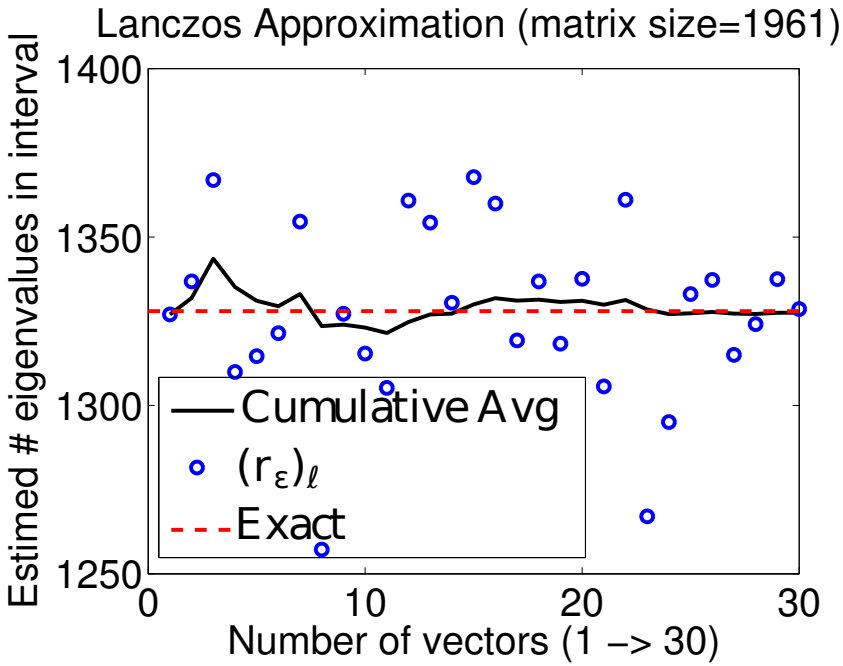
➤ In practice replace by

$$\epsilon = \min\{t : \lambda_n \leq t \leq \lambda_1, |\phi'(t)| \geq \text{tol}\}$$

Experiments



(A)



(B)

(A) The DOS found by KPM.

(B) Approximate rank estimation by The Lanczos method for the example `netz4504`.

Tests with Matérn covariance matrices for grids

- Important in statistical applications

Approximate Rank Estimation of Matérn covariance matrices

Type of Grid (dimension)	Matrix Size	# λ_i's $\geq \epsilon$	r_ϵ	
			KPM	Lanczos
1D regular Grid (2048×1)	2048	16	16.75	15.80
1D no structure Grid (2048×1)	2048	20	20.10	20.46
2D regular Grid (64×64)	4096	72	72.71	72.90
2D no structure Grid (64×64)	4096	70	69.20	71.23
2D deformed Grid (64×64)	4096	69	68.11	69.45

- For all test $M(deg) = 50, n_v$=30

Application 4: The LogDeterminant

Evaluate the Log-determinant of A:

$$\log\det(A) = \text{Trace}(\log(A)) = \sum_{i=1}^{n} \log(\lambda_i).$$

A is SPD.

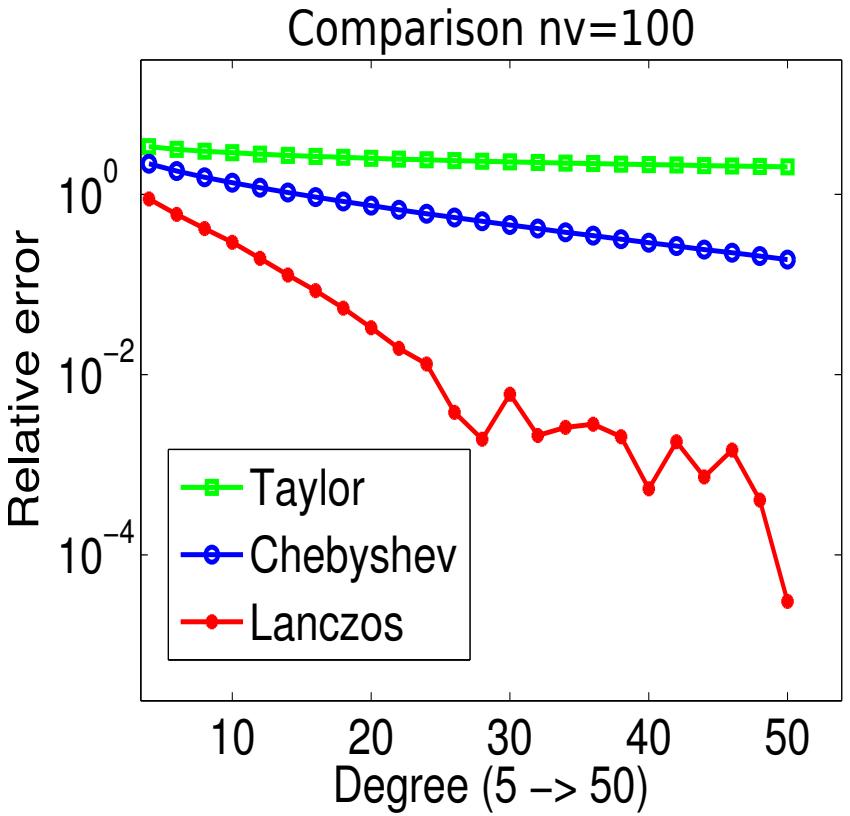
➤ Estimating the log-determinant of a matrix equivalent to estimating the trace of the matrix function $f(A) = \log(A)$.

➤ Can invoke Stochastic Lanczos Quadrature (SLQ) to estimate this trace.

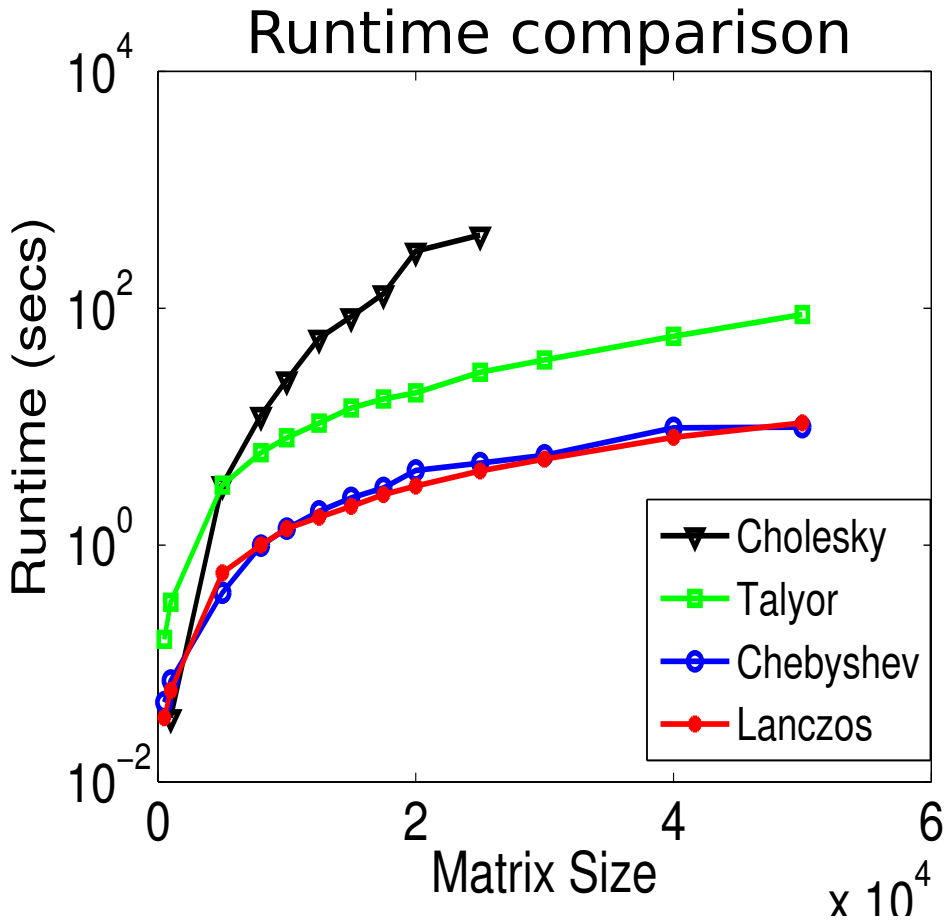
Numerical example: A graph Laplacian `california` of size 9664×9664, $nz \approx 10^5$ from the Univ. of Florida collection.

Rel. error vs degree

- 3 methods: Taylor Series, Chebyshev expansion, SLQ
- # starting vectors $nv = 100$ in all three cases.



Runtime comparisons



Application 6: Log-likelihood.

Comes from parameter estimation for Gaussian processes

➤ Objective is to maximize the log-likelihood function with respect to a 'hyperparameter' vector ξ

$$\log p(z \mid \xi) = -\tfrac{1}{2}\left[z^\top S(\xi)^{-1} z + log \det S(\xi) + \text{cst}\right]$$

where z = data vector and $S(\xi)$ == covariance matrix parameterized by ξ

➤ Can use the same Lanczos runs to estimate $z^\top S(\xi)^{-1} z$ and logDet term simultaneously.

Application 7: calculating nuclear norm

- $\|X\|_* = \sum \sigma_i(X) = \sum \sqrt{\lambda_i(X^T X)}$
- Generalization: Schatten p-norms

$$\|X\|_{*,p} = [\sum \sigma_i(X)^p]^{1/p}$$

- See:

J. Chen, S. Ubaru, YS, "Fast estimation of log-determinant and Schatten norms via stochastic Lanczos quadrature", (Submitted).

Conclusion

- Estimating traces is a key ingredient in many algorithms
- Physics, machine learning, matrix algorithms, ..
- .. many new problems related to ‘data analysis’ and ’statistics’, and in signal processing,

Q: Can we do better than standard random sampling?