Sampling algorithms in numerical linear algebra and their application

Yousef Saad
Department of Computer Science and Engineering

University of Minnesota

Caltech, Nov. 11, 2013

Introduction

- 'Random Sampling' or 'probabilistic methods': use of random data to solve a given problem.
- Eigenvalues, eigenvalue counts, traces, ...
- Many well-known algorithms use a form of random sampling: The Lanczos algorithm
- Recent work : probabilistic methods - See [Halko, Martinsson, Tropp, 2010]
- Huge interest spurred by 'big data'
- In this talk: A few specific applications of random sampling in numerical linear algebra

Introduction: A few examples

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

➤ Arises in cross validation :

$$\frac{\|(I - A(\theta))g\|_2}{\text{Tr}\,(I - A(\theta))} \quad \text{with} \quad A(\theta) \equiv I - D(D^T D + \theta L L^T)^{-1} D^T,$$

D == blurring operator and L is the regularization operator

➤ In [Huntchinson '90] Tr[Inv[A]] is stochastically estimated

➤ 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

[Ref: H. Röder, R. N. Silver, D. A. Drabold, J. J. Dong, Phys. Rev. B. 55, 15382 (1997)]

➤ Will be covered in detail later in this talk.

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

- ➤ Harder than just getting the trace
- ➤ Arises in 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.
- ➤ In uncertainty quantification, the diagonal of the inverse of a covariance matrix is needed [Bekas, Curioni, Fedulova '09]

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

- Arises in any density matrix approach in quantum modeling - for example Density Functional Theory.

- Here, f = Fermi-Dirac operator:

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

Note: when $T \rightarrow 0$ then f becomes a step function.

Note: if f is approximated by a rational function then diag[f(A)] $\approx$ a lin. combination of terms like diag[$(A - \sigma_i I)^{-1}$]

- Linear-Scaling methods based on approximating $f(H)$ and $\mathbf{Diag}(f(H))$ – avoid 'diagonalization' of H

- Rich litterature on ‘linear scaling’ or ’order n’ methods
- The review paper [Benzi, Boito, Razouk, “Decay properties of Specral Projectors with applications to electronic structure”, SIAM review, 2013] provides theoretical foundations
- Several references on approximating $textDiag(f(H))$ for this purpose – See e.g., work by L. Lin, C. Yang, E. E [Code: SelInv]

DIAGONAL OF THE INVERSE

Motivation: Dynamic Mean Field Theory (DMFT)

➤ Quantum mechanical studies of highly correlated particles

➤ Equation to be solved (repeatedly) is Dyson's equation

$$G(\omega) = [(\omega + \mu)I - V - \Sigma(\omega) + T]^{-1}$$

- ω (frequency) and μ (chemical potential) are real
- V = trap potential = real diagonal
- $\Sigma(\omega)$ == local self-energy - a complex diagonal
- T is the hopping matrix (sparse real).

➤ Interested only in diagonal of $G(\omega)$ – in addition, equation must be solved self-consistently and ...

➤ ... must do this for many ω's

Stochastic Estimator

Notation:

- A = original matrix, $B = A^{-1}$.
- $\delta(B) = \text{diag}(B)$ [matlab notation]
- $\mathcal{D}(B)$ = diagonal matrix with diagonal $\delta(B)$
- $\odot$ and $\oslash$: Elementwise multiplication and division of vectors
- $\{v_j\}$: Sequence of s random vectors

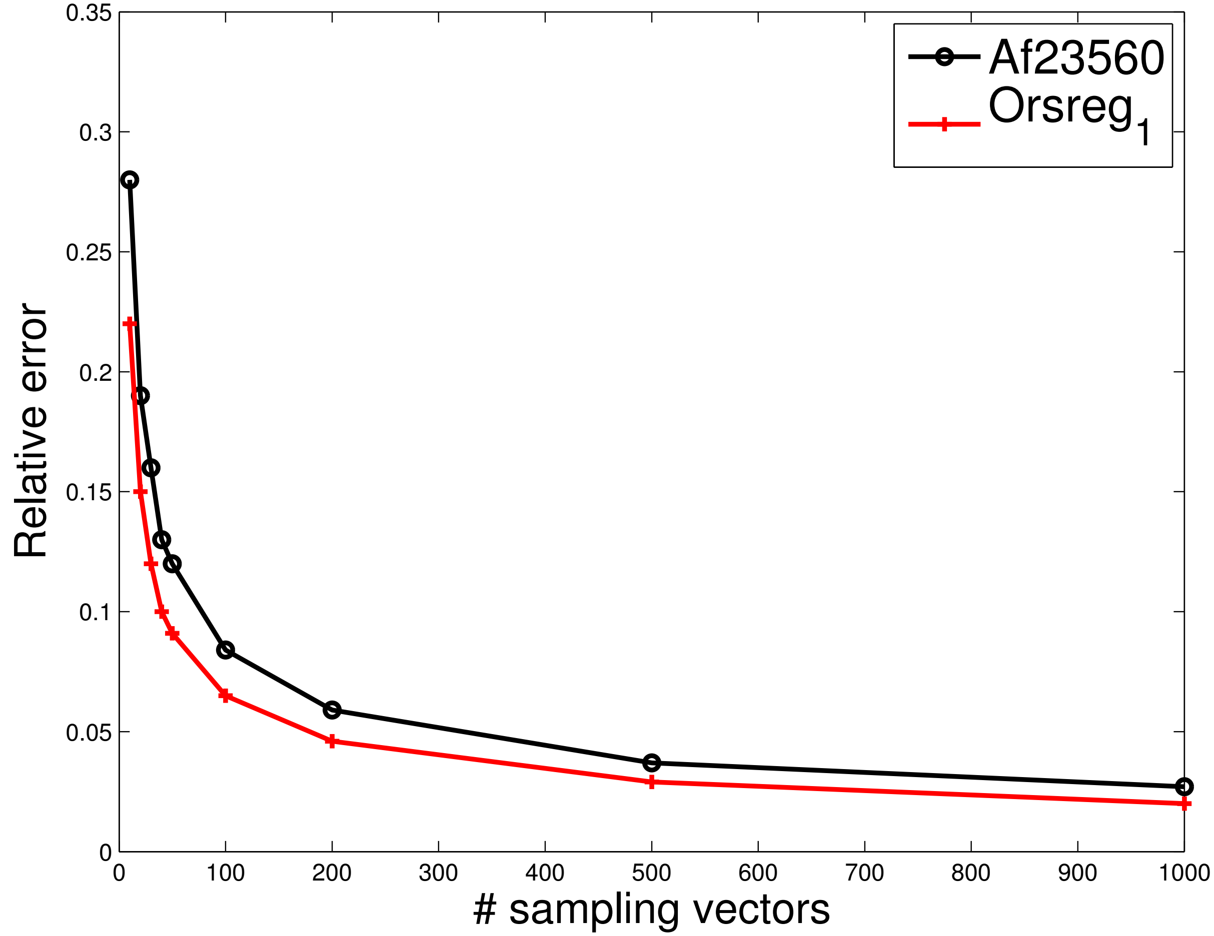
Result:

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

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

Typical convergence curve for stochastic estimator

- Estimating the diagonal of inverse of two sample matrices



➤ Let $V_s = [v_1, v_2, \ldots, v_s]$. Then, alternative expression:

$$\mathcal{D}(B) \approx \mathcal{D}(BV_sV_s^\top)\mathcal{D}^{-1}(V_sV_s^\top)$$

Question: When is this result exact?

Answer:

- Let $V_s \in \mathbb{R}^{n\times s}$ with rows $\{v_{j,:}\}$; and $B \in \mathbb{C}^{n\times n}$ with elements $\{b_{jk}\}$
- Assume that: $\langle v_{j,:}, v_{k,:}\rangle = 0$, $\forall j \neq k$, s.t. $b_{jk} \neq 0$

Then:

$$\mathcal{D}(B) = \mathcal{D}(BV_sV_s^\top)\mathcal{D}^{-1}(V_sV_s^\top)$$

➤ Approximation to b_{ij} exact when rows i and j of V_s are $\perp$

Ideas from information theory: Hadamard matrices

➤ Want the rows of V (with each row scaled by its 2-norm) to be as ‘mutually orthogonal as possible, i.e., want to minimize

$$E_{rms} = \frac{\|I - VV^T\|_F}{\sqrt{n(n-1)}} \quad \text{or} \quad E_{max} = \max_{i \neq j} |VV^T|_{ij}$$

➤ Problems that arise in coding: find code book [rows of V = code words] to minimize ’cross-correlation amplitude’

➤ Welch bounds:

$$E_{rms} \geq \sqrt{\frac{n-s}{(n-1)s}} \qquad E_{max} \geq \sqrt{\frac{n-s}{(n-1)s}}$$

➤ Result: $\exists$ a sequence of s vectors v_k with binary entries which achieve the first Welch bound iff $s = 2$ or $s = 4k$.

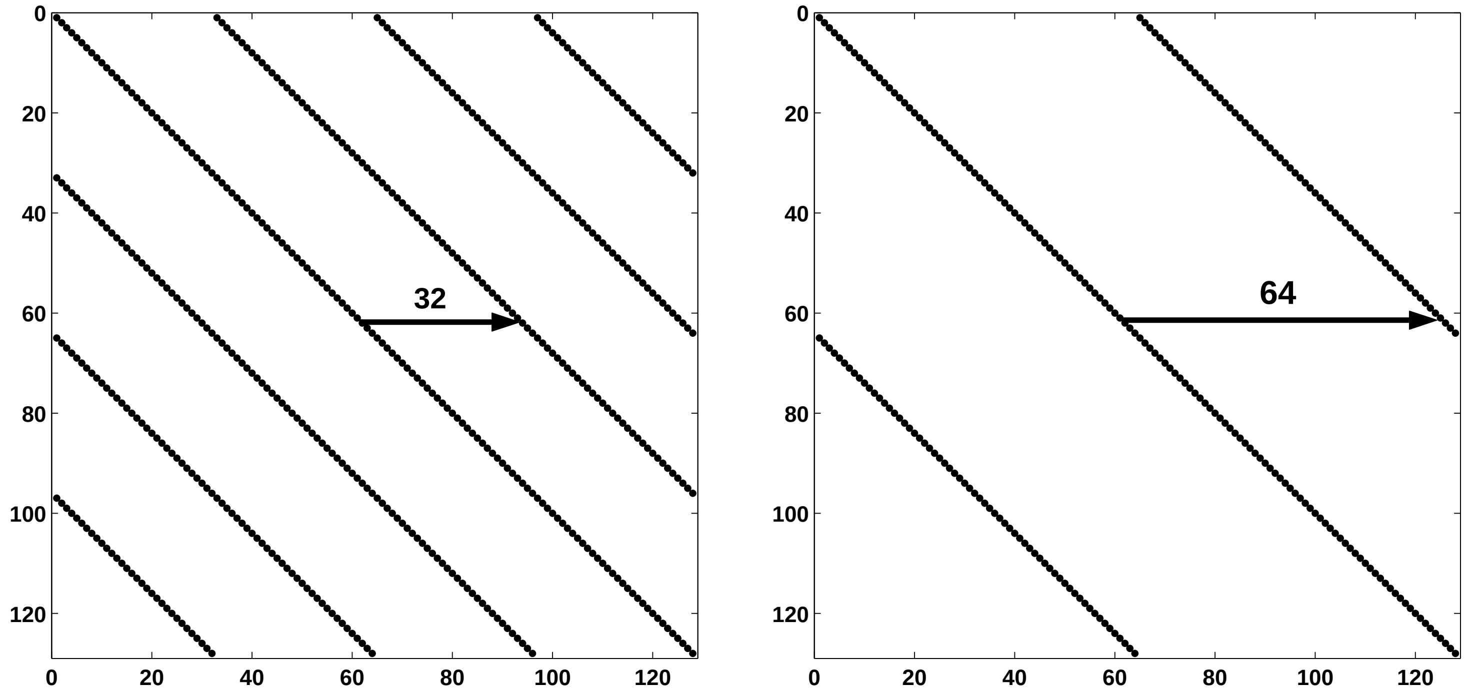
- Hadamard matrices are a special class: $n \times n$ matrices with entries ± 1 and such that $HH^\top = nI$.

Examples : $\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ and $\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$.

- Achieve both Welch bounds
- Can build larger Hadamard matrices recursively:

Given two Hadamard matrices H_1 and H_2, the Kronecker product $H_1 \otimes H_2$ is a Hadamard matrix.

- Too expensive to use the whole matrix of size n
- Can use V_s = matrix of s first columns of H_n



Pattern of $V_s V_s^\top$, for $s = 32$ and $s = 64$.

Test: Hadamard vectors for AF23560 and ORSREG_1

♯ vectors	`AF23560` RelErr	`ORSREG_1` RelErr
4	0.99	0
8	0.5	0
16	0.0028	0
32	0	0
64	0	0
⋮	⋮	⋮
1024	0	0

➤ Note: half-banwidth of AF23560 is 305. half-banwidth of ORSREG1 is 442.

Other methods for the diagonal of matrix inverse

- Probing techniques [exploit sparsity]
- Direct methods: use LU factorization – exploit paths in graph
- Domain Decomposition type methods [J. Tang and YS'2009]

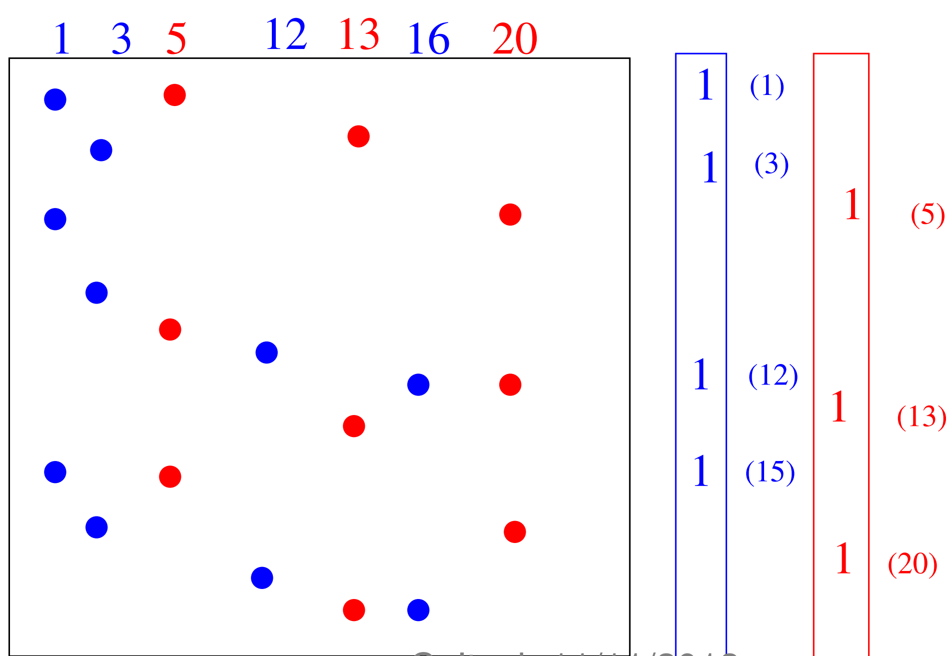
Standard probing (e.g. to compute a Jacobian)

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

Basis of the method: can compute Jacobian if a coloring of the columns is known so that no two columns of the same color overlap.

All entries of same color can be computed with one matvec.

Example: For all blue entries multiply B by the blue vector on right.



What about Diag(inv(A))?

➤ Define v_i - probing vector associated with color i:

$$[v_i]_k = \begin{cases} 1 \text{ if } color(k) == i \\ 0 \text{ otherwise} \end{cases}$$

➤ Standard probing satisfies requirement of Proposition but...

➤ ... this coloring is not what is needed! [It is an overkill]

Alternative:

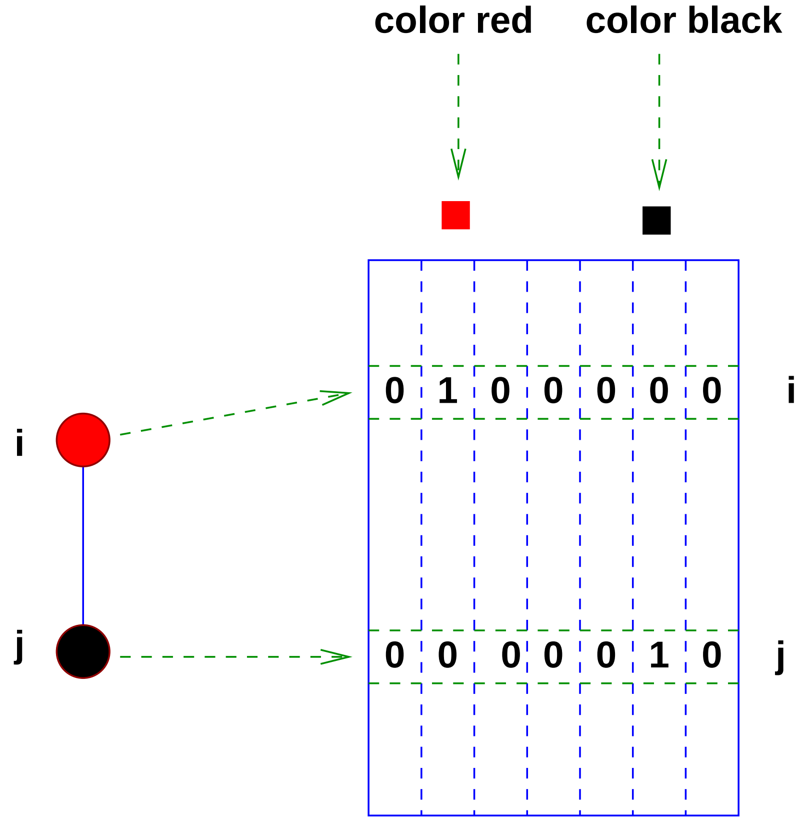
➤ Color the graph of B in the standard graph coloring algorithm [Adjacency graph, not graph of column-overlaps]

Result: Graph coloring yields a valid set of probing vectors for $\mathcal{D}(B)$.

Proof:

➤ Column v_c: one for each node i whose color is c, zero elsewhere.

➤ Row i of V_s: has a '1' in column c, where $c = color(i)$, zero elsewhere.

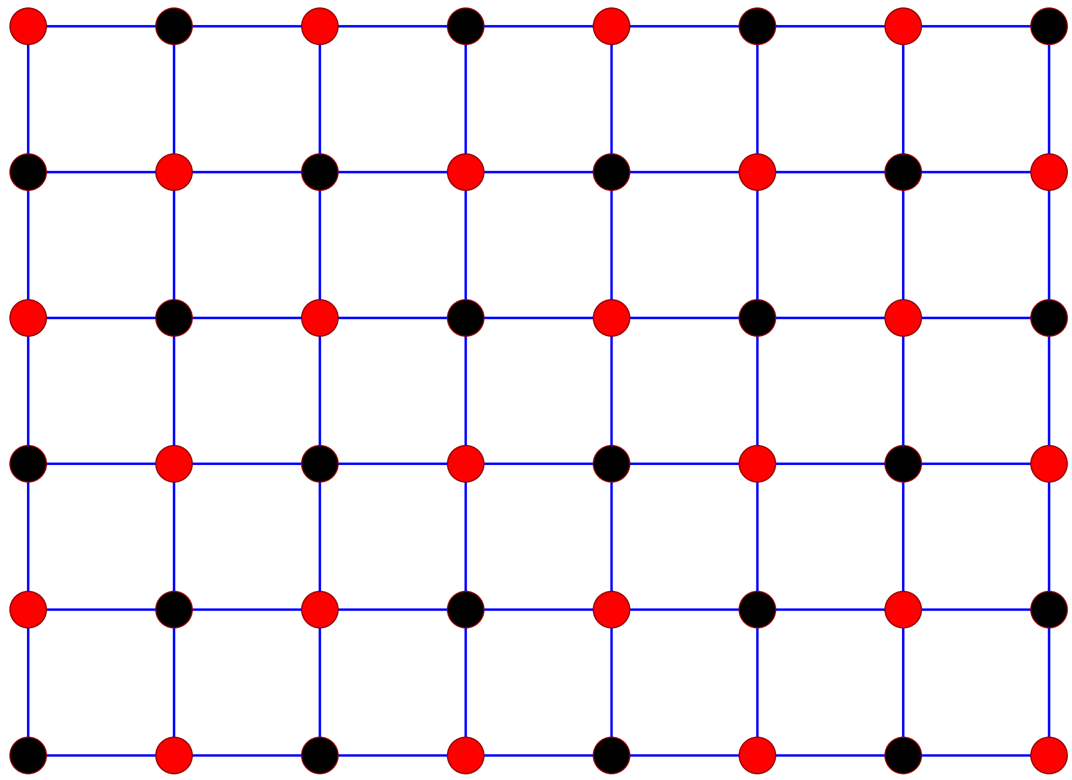


➤ If $b_{ij} \neq 0$ then in matrix V_s:

- i-th row has a '1' in column $color(i)$, '0' elsewhere.
- j-th row has a '1' in column $color(j)$, '0' elsewhere.

➤ The 2 rows are orthogonal.

Example:



- Two colors required for this graph $\rightarrow$ two probing vectors
- Standard method: 6 colors [graph of $B^T B$]

Guessing the pattern of B

- Assume A diagonally dominant
- Write $A = D - E$, with $D = \mathcal{D}(A)$. Then :

$$A^{-1} \approx \underbrace{(I + F + F^2 + \cdots + F^k)D^{-1}}_{B^{(k)}} \text{ with } F \equiv D^{-1}E$$

- When A is D.D. $\|F^k\|$ decreases rapidly.
- Can approximate pattern of B by that of $B^{(k)}$ for some k.
- Distance k graph.

Q: How to select k? Heuristic: Inspect $A^{-1}e_j$ for some j

Improvements

- Recent work by A. Stathopoulos, J. Laeuchli, and K. Orginos, on hierarchical probing. Produce approximate k-distance coloring of the graph to determine the patterns
- Somewhat specific to Lattice QCD
- E. Aune, D. P. Simpson, J. Eidsvik [Statistics and Computing 2012] combine probing with stochastic estimation. Good improvements reported.

EIGENVALUE COUNTS

Eigenvalue counts [with E. Polizzi and E. Di Napoli]

The problem:

➤ Find an estimate of the number of eigenvalues of a matrix in a given interval $[a, \quad b]$.

Main motivation:

➤ Eigensolvers based on splitting the spectrum intervals and extracting eigenpairs from each interval independently.

➤ Contour integration-type methods:
- FEAST approach [Polizzi 2011]
- Sakurai-Sigiura method [2002]

➤ Polynomial filtering:
- Schofield, Chelikowsky, YS'2011.

Eigenvalue counts: Standard approach

- Let spectrum of a Hermitnan matrix $\boldsymbol{A}$ be

$$\boldsymbol{\lambda_1 \le \lambda_2 \le \cdots \le \lambda_n}$$

with eigenvectors $\boldsymbol{u_1, u_2, \cdots, u_n}$

- $\boldsymbol{a, b}$ such that $\boldsymbol{\lambda_1 \le a \le b \le \lambda_n}$.
- Want number $\boldsymbol{\mu_{[a,b]}}$ of $\boldsymbol{\lambda_i}$'s $\in [\boldsymbol{a}, \boldsymbol{b}]$
- Standard method: Use Sylvester inertia theorem
- Requires two $\boldsymbol{LDL^T}$ factorizations $\rightarrow$ can be expensive!

- Alternative: Exploit trace of the eigen-projector:

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

- We know that the trace of P is the wanted number $\mu_{[a,b]}$
- Goal: calculate an approximation to :

$$\mu_{[a,b]} = \text{Tr}\,(P)\ .$$

- P is not available ... but can be approximated by
  - a polynomial in A, or
  - a rational function in A.

Eigenvalue counts: Approximation theory viewpoint

- Interpret P as a step function of A, namely:

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

- Hutchinson's unbiased estimator uses only matrix-vector products to approximate the trace of a generic matrix A.

- Generate random vectors $v_k, k = 1, .., n_v$ with equally probable entries ± 1. Then:

$$\mathtt{tr}(A) \approx \frac{n}{n_v} \sum_{k=1}^{n_v} v_k^\top A v_k.$$

- No need to restrict values to ± 1

Polynomial filtering

- $h(t) \approx \psi(t)$, where ψ is a polynomial of degree k.
- We can estimate the trace of P as:

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

- We use degree p Chebyshev polynomials:

$$h(t) \approx \psi_p(t) = \sum_{j=0}^{p} \gamma_j T_j(t).$$

Computing the polynomials: Jackson-Chebyshev

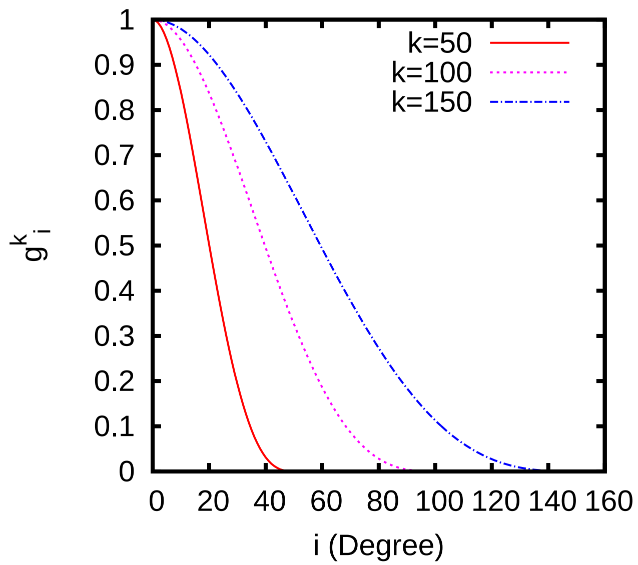
Chebyshev-Jackson approximation of a function f:

$$f(x) \approx \sum_{i=0}^{k} g_i^k \gamma_i T_i(x)$$

$$\gamma_i = \frac{2 - \delta_{i0}}{\pi} \int_{-1}^{1} \frac{1}{\sqrt{1 - x^2}} f(x) dx \quad \delta_{i0} = \text{ Kronecker symbol}$$

The g_i^k's attenuate higher order terms in the sum.

Attenuation coefficient g_i^k for k=50,100,150 $\rightarrow$



Let $\alpha_k = \dfrac{\pi}{k+2}$, then :

$$g_i^k = \frac{\left(1 - \frac{i}{k+2}\right) \sin(\alpha_k) \cos(i\alpha_k) + \frac{1}{k+2} \cos(\alpha_k) \sin(i\alpha_k)}{\sin(\alpha_k)}$$

See

'*Electronic structure calculations in plane-wave codes without diagonalization.*' Laurent O. Jay, Hanchul Kim, YS, and James R. Chelikowsky. *Computer Physics Communications*, 118:21–30, 1999.

The expansion coefficients γ_i

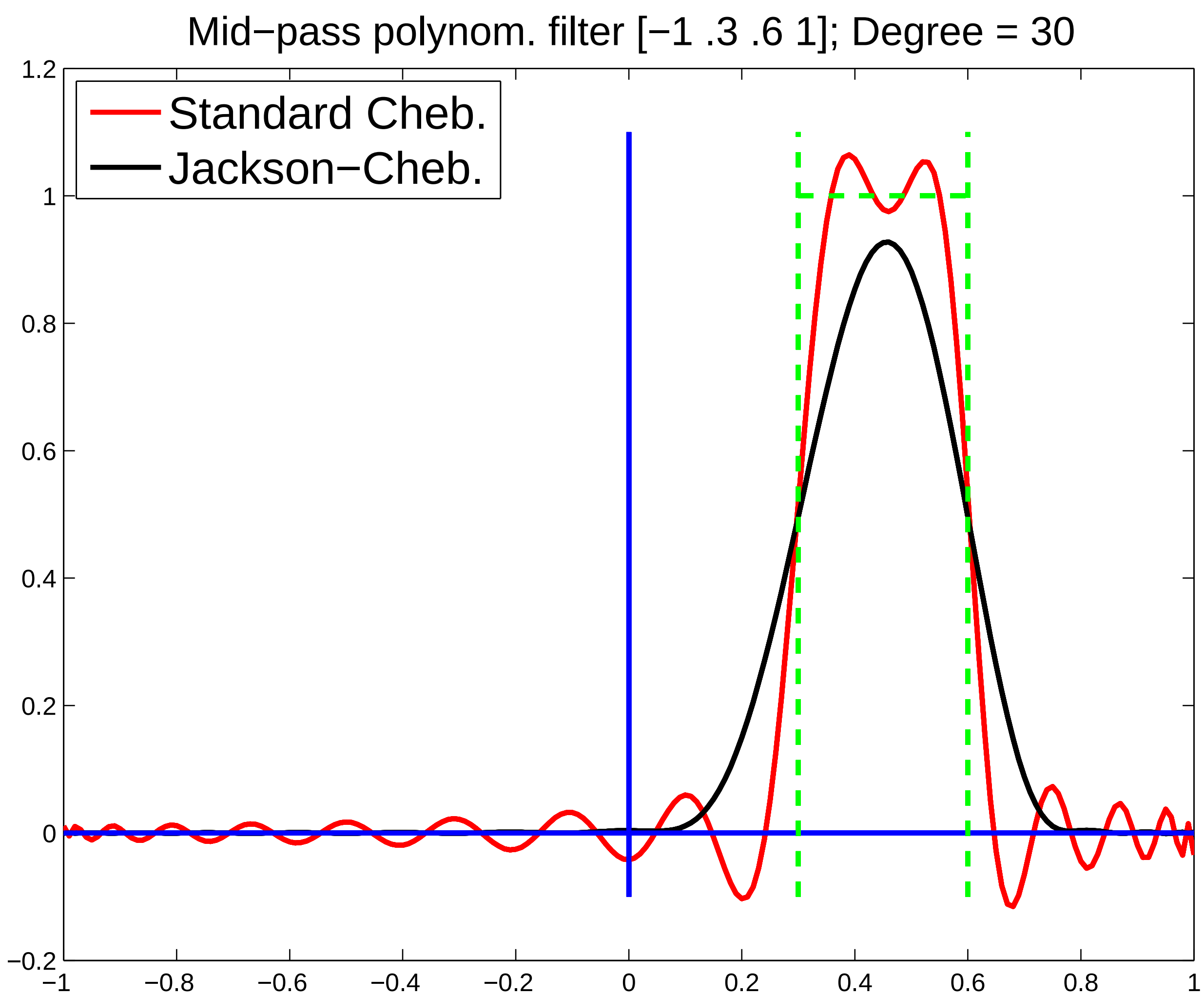
When $f(x)$ is a step function on $[a, b] \subseteq [-1\ 1]$:

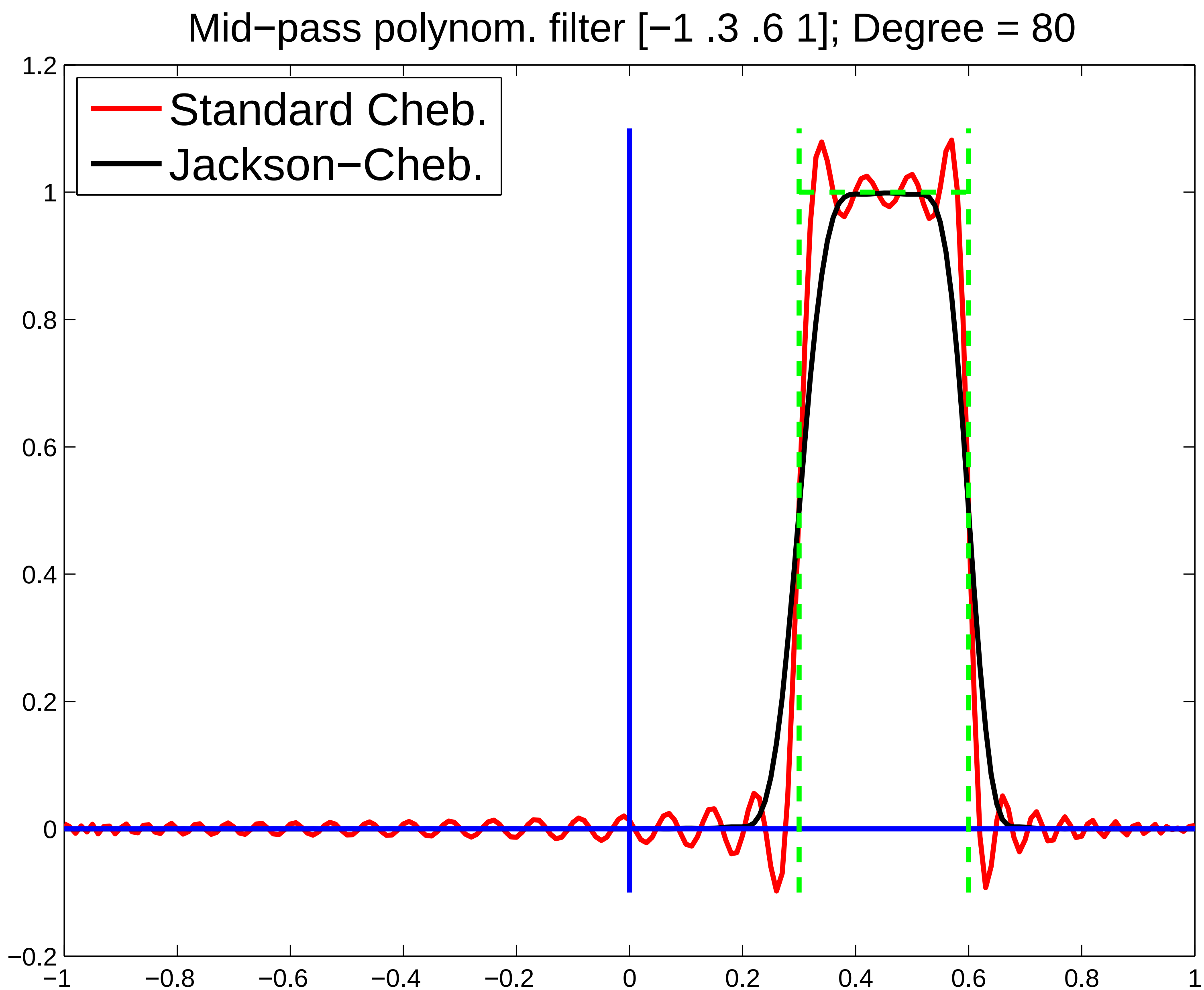
$$\gamma_i = \begin{cases} \frac{1}{\pi}(\arccos(a) - \arccos(b)) & : i = 0 \\ \frac{2}{\pi}\left(\frac{\sin(i\arccos(a)) - \sin(i\arccos(b))}{i}\right) & : i > 0 \end{cases}$$

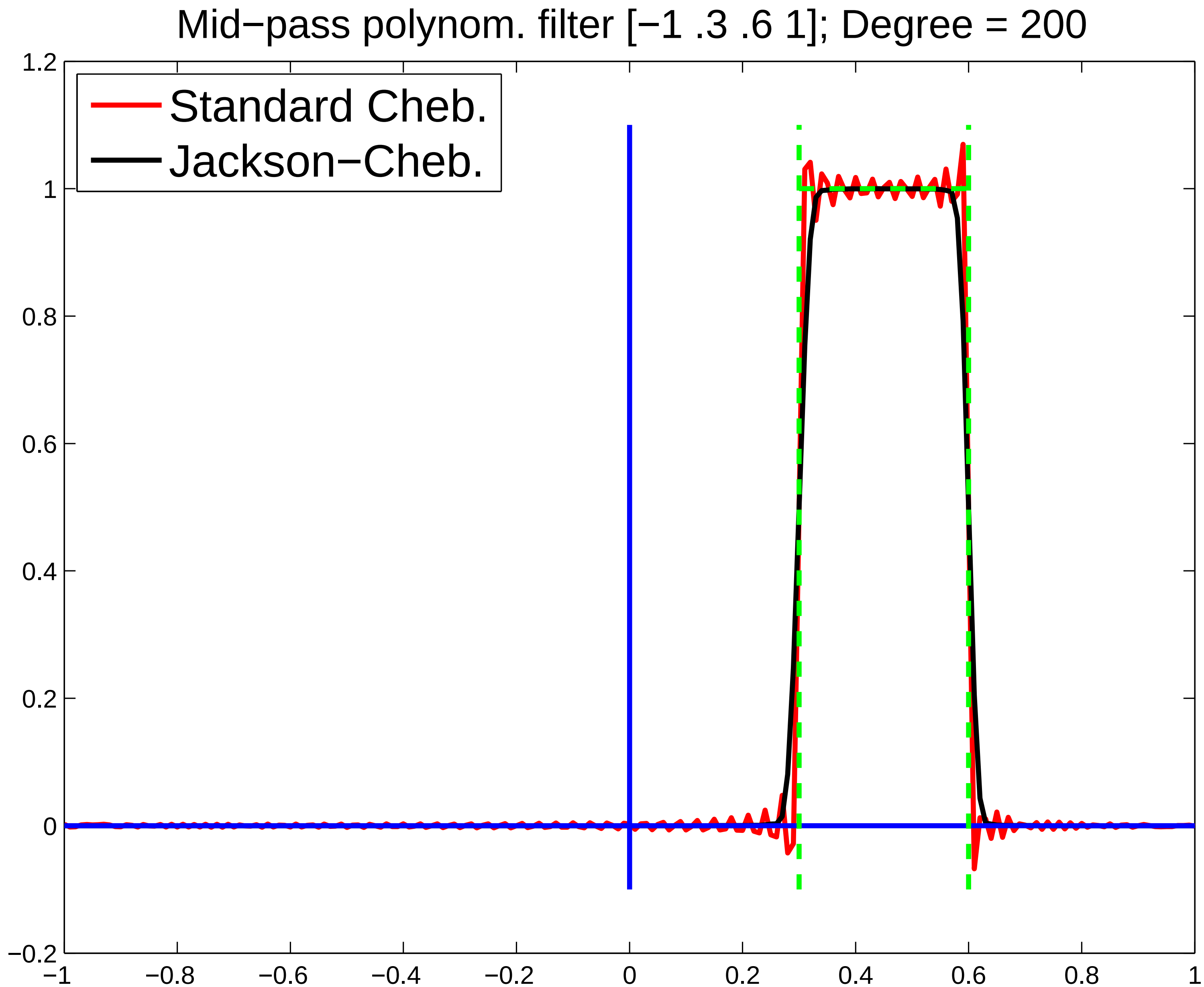
- A few examples follow –

Computing the polynomials: Jackson-Chebyshev

- Polynomials of degree 30 for $[a, \; b] = [.3, \; .6]$




Mid−pass polynom. filter [−1 .3 .6 1]; Degree = 80
Standard Cheb.
Jackson−Cheb.
1.2
1
0.8
0.6
0.4
0.2
0
−0.2
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1


Mid−pass polynom. filter [−1 .3 .6 1]; Degree = 200
Standard Cheb.
Jackson−Cheb.
1.2
1
0.8
0.6
0.4
0.2
0
−0.2
−1
−0.8
−0.6
−0.4
−0.2
0
0.2
0.4
0.6
0.8
1

$$\mu_{[a,b]} = \text{Tr}\,(P) \approx \frac{n}{n_v} \sum_{k=1}^{n_v} \left[\sum_{j=0}^{p} \gamma_j v_k^T T_j(A) v_k \right].$$

Easy to compute $T_j(A)v_k$ with 3-term recurrence of Chebyshev polynomials

$$w_{j+1} = 2Aw_j - w_{j-1}.$$

(A is transformed so its eigenvalues are in $[-1\ 1]$)

Generalized eigenvalue problems

$$Ax = \lambda Bx$$

- Matrices A and B are symmetric and B is positive definite.

The projector P becomes

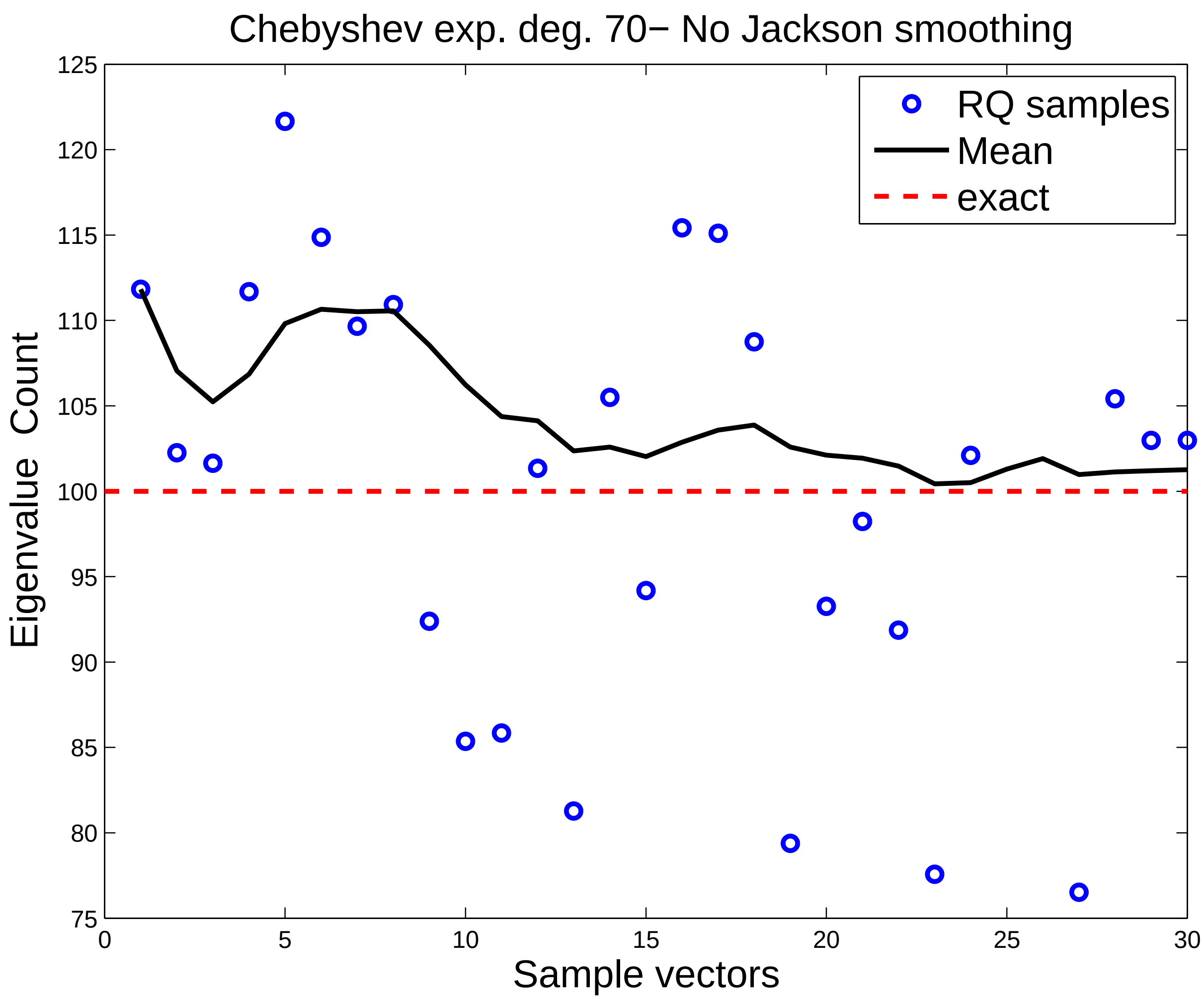
$$P = \sum_{\lambda_i \in [a\ b]} u_i u_i^T B,$$

- Again: Eigenvalue count == Tr (P)
- Exploit relation: inertia$(A - \sigma B)$ = inertia$(B^{-1}A - \sigma I)$
- No need to factor or to solve systems

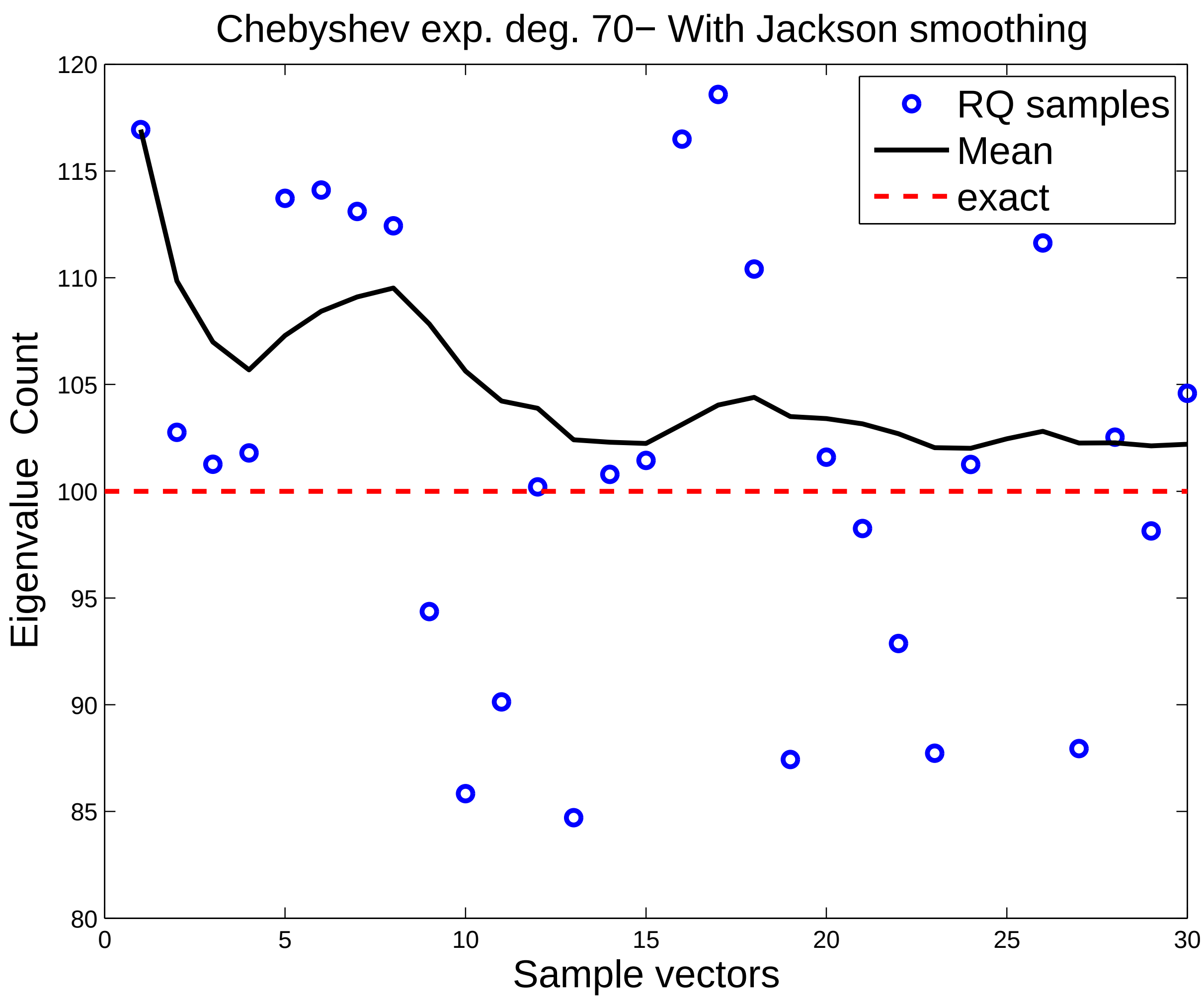
An example

- Matrix ‘Na5’ from PARSEC [see U. Florida collection]
- $n = 5832$, $nnz = 305630$ nonzero entries.
- Obtain the eigenvalue count when $a = (\lambda_{100} + \lambda_{101})/2$ and $b = (\lambda_{200} + \lambda_{201})/2$ so $\mu_{[a,b]} = 100$.
- Use pol. of degree 70.

Without Jackson Damping

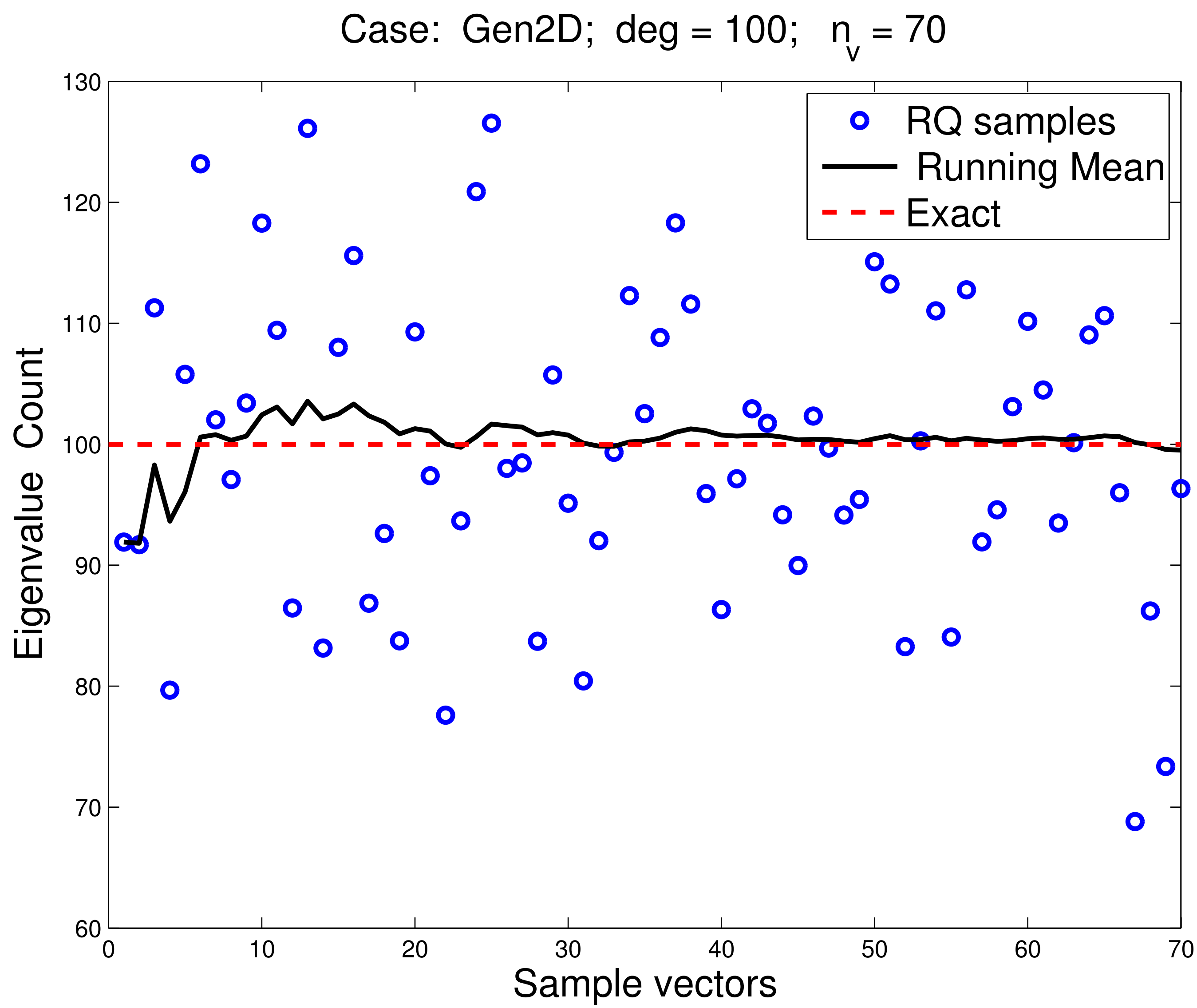


With Jackson Damping



An example from FEAST

- FEAST developed by Eric Polizzi (Amherst)..
- Based on a form of subspace iteration with a rational function of A
- Also works for generalized problems $Au = \lambda B$.
- Example: a small generalized problem ($n = 12,450, nnz = 86,808$).
- Result with standard Chebyshev shown. Deg=100, $nv = 70$.


Case: Gen2D; deg = 100; n_v = 70
RQ samples
Running Mean
Exact
Eigenvalue Count
Sample vectors
130
120
110
100
90
80
70
60
0
10
20
30
40
50
60
70

➤ A few more comments:

- Method also works with rational approximations ...
- .. and it works for nonsymmetric problems (eigenvalues inside a given contour).
- For details see paper:

E. Di Napoli, E, Polizzi, and YS. *Efficient estimation of eigenvalue counts in an interval.* Preprint – see arXiv: http://arxiv.org/abs/13

DENSITY OF STATES

Computing Densities of States [with Lin-Lin and Chao Yang]

➤ 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

➤ Note: number of eigenvalues in an interval $[a, b]$ is

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

- $\phi(t)$ == a probability distribution function == probability of finding eigenvalues of A in a given infinitesimal interval near t.
- DOS is also referred to as the spectral density
- 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

Issue: How to deal with Distributions

- Highly discontinuous nature – not easy to handle

- Solution for practical and theoretical purposes: replace ϕ by a 'blurred" (continuous) version ϕ_σ:

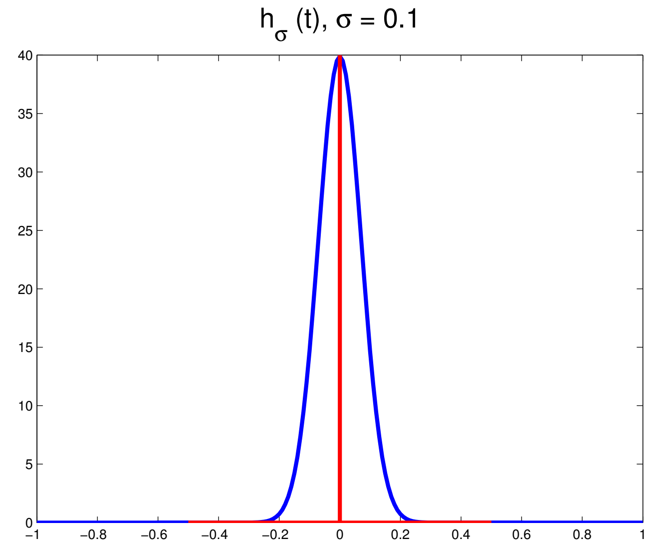
$$\phi_\sigma(t) = \frac{1}{n}\sum_{j=1}^{n} h_\sigma(t-\lambda_j),$$

where $h_\sigma(t)$ = any $\mathcal{C}^\infty$ function s.t.:

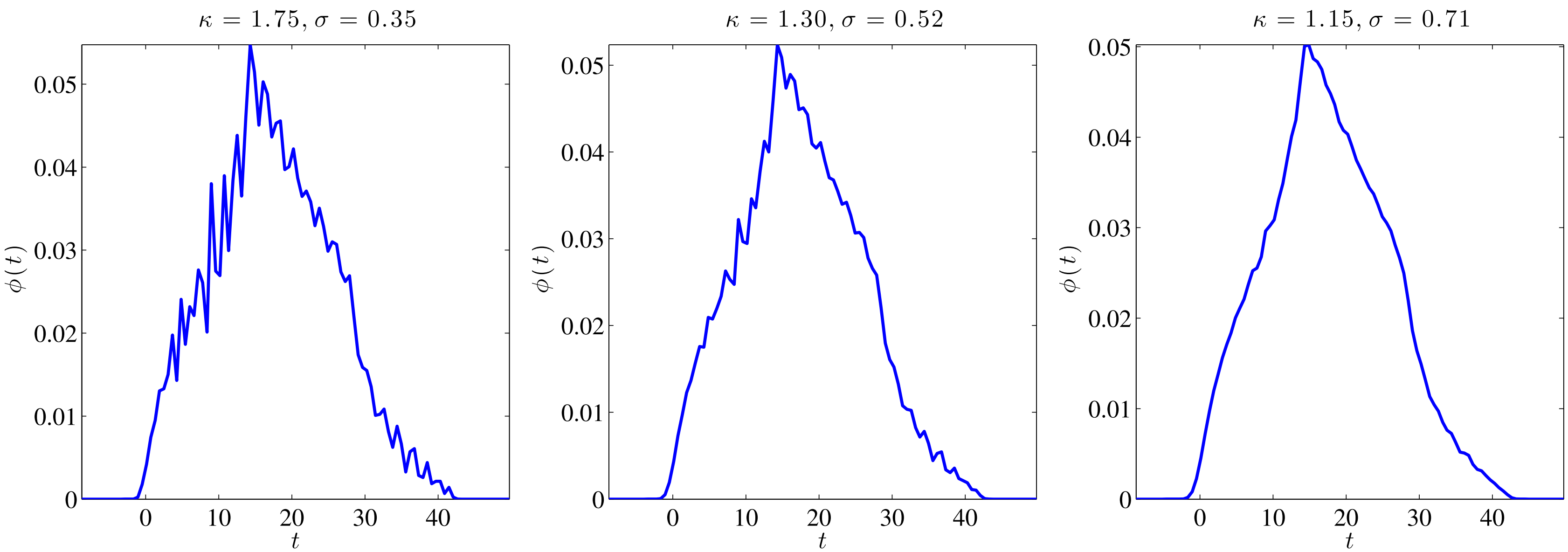
- $\int_{-\infty}^{+\infty} h_\sigma(s)ds = 1$
- h_σ has a peak at zero

- An example is the Gaussian:

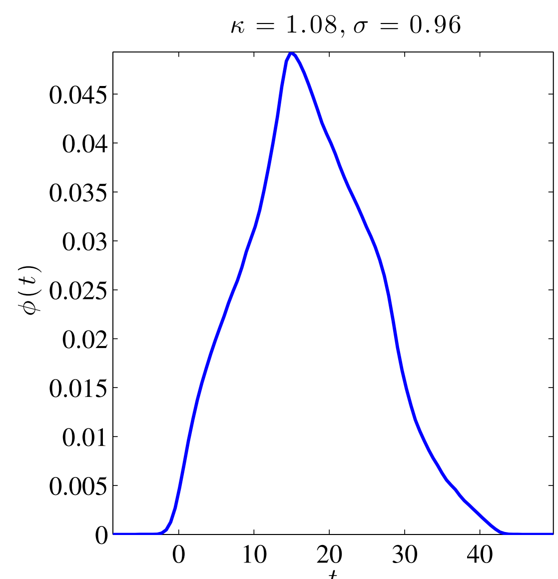
$$h_\sigma(t) = \frac{1}{(2\pi\sigma^2)^{1/2}} e^{-\frac{t^2}{2\sigma^2}}.$$



How to select $\boldsymbol{\sigma}$? Example for $\boldsymbol{Si_2}$



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



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 estimators [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:

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

- Here $2-\delta_{k0}$ == 1 when $k=0$ and == 2 otherwise.
- Note: $\sum T_k(\lambda_i) =$ *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:

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

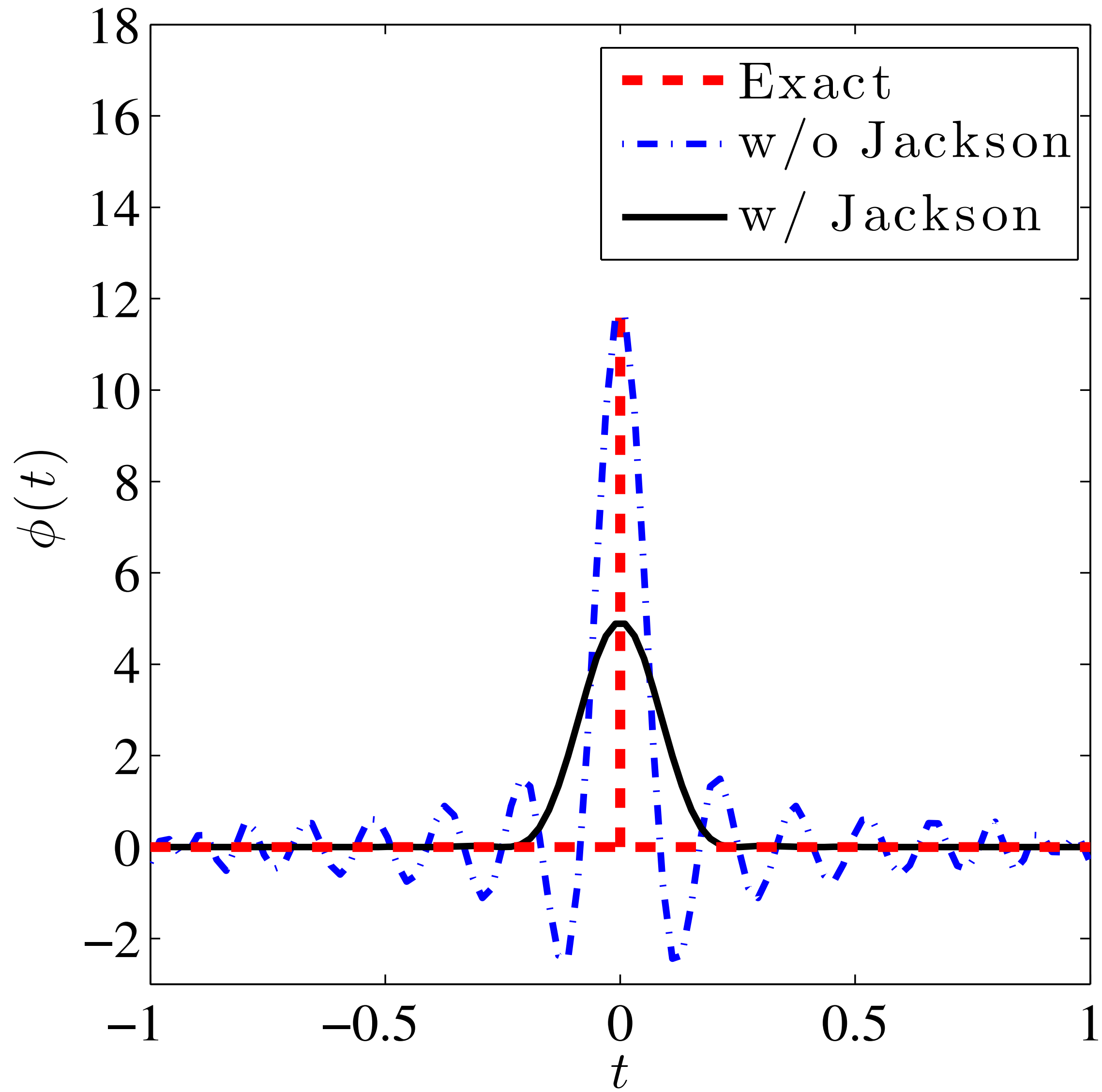
- To compute scalars of the form $v^T T_k(A) v$, exploit 3-term recurrence of the Chebyshev polynomial:

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

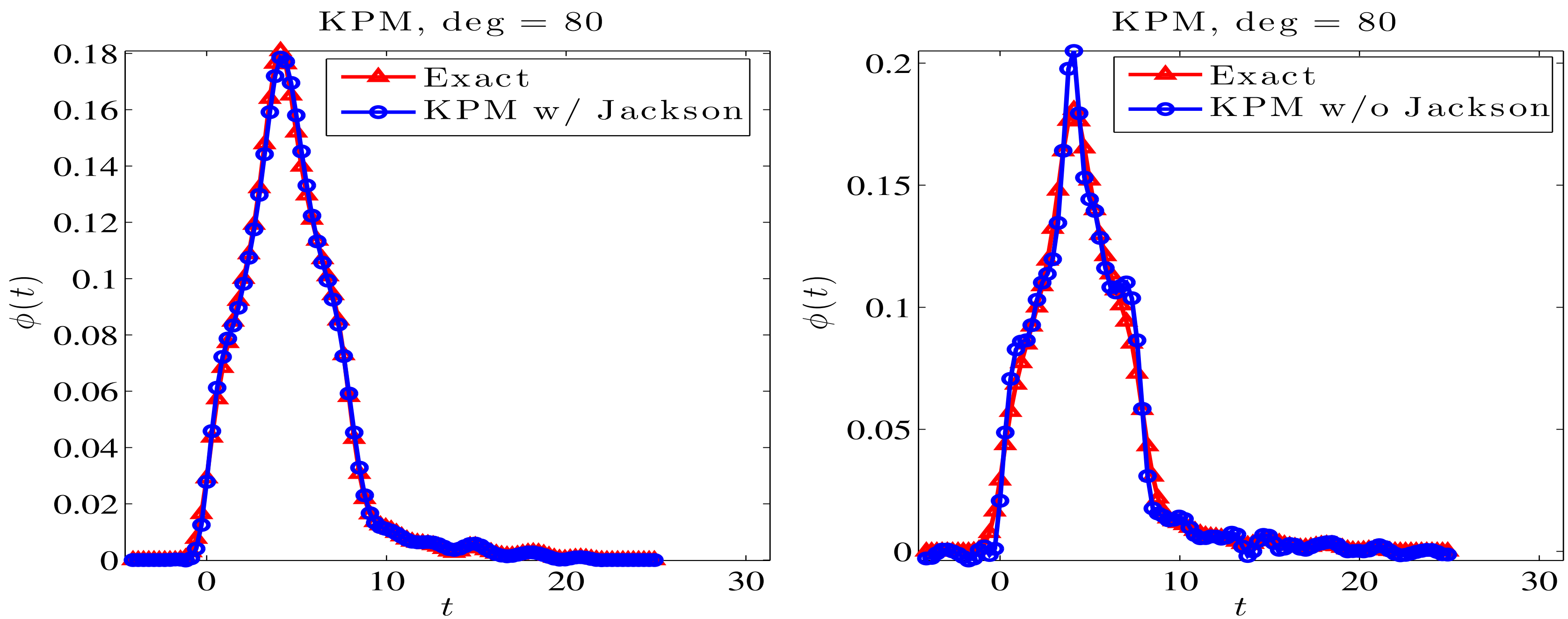
so if we let $v_k \equiv T_k(A)v$, we have

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

➤ Same Jackson smoothing as before can be used



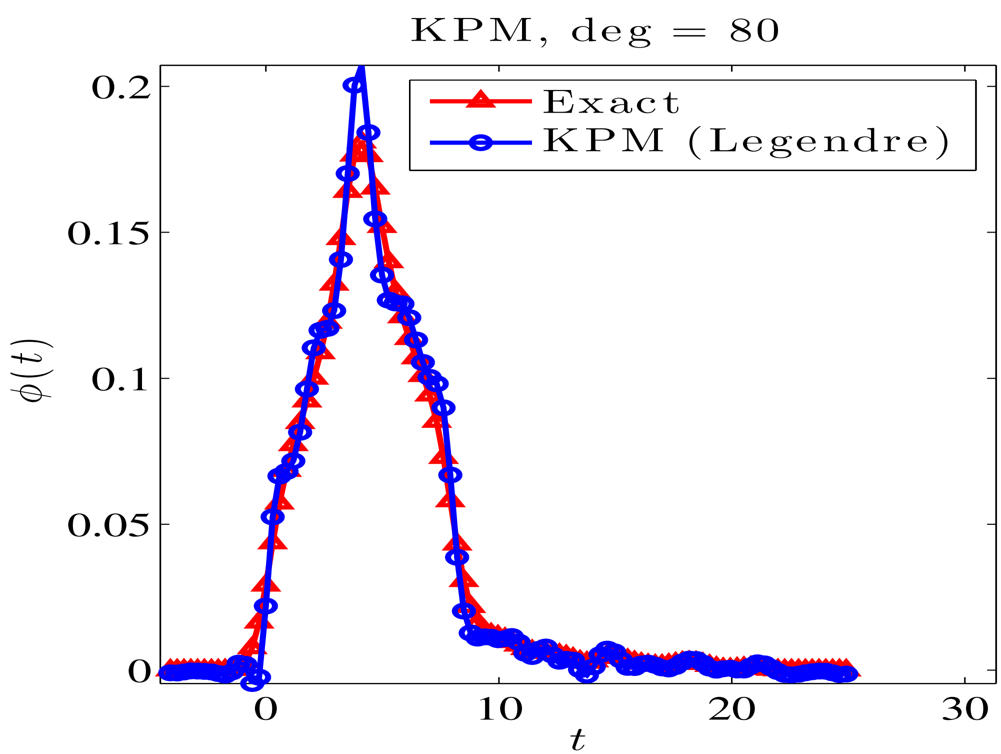
An example with degree 80 polynomials



Left: Jackson damping; right: without Jackson damping.

Why not use Legendre Polynomials?

- They yield very similar results
- Same Example as before – with same degree:



The Lanczos Spectroscopic approach

- Described in Lanczos' book "Applied Analysis, (1956)" as a means to compute eigenvalues.
- Idea: assimilate λ_i;s to frequencies and perform Fourrier analysis to extract them
- Also relies on Chebyshev polynomials
- Though not emphasized in the description, the method uses random sampling
- Let B a symmetric real matrix with eigevalues in [-1,1]
- Let v_0 == an initial vector – expand in eigenbasis as

$$v_0 = \sum_{j=1}^{n} \beta_j u_j, \quad \text{with} \quad \beta_j = u_j^T v_0$$

- Let $v_k = T_k(A)v_0$, for $k = 0, \cdots, M$. Then:

$$v_0^T v_k = \sum_{j=1}^{n} \beta_j^2 T_k(\lambda_j) = \sum_{j=1}^{n} \beta_j^2 \cos(k\theta_j), \text{ with } \lambda_j = \cos\theta_j.$$

View $v_0^T v_k$ as a discretization of the periodic function to the right sampled at $t = 0, 1, \cdots, M$.

$$f(t) = \sum_{j=1}^{n} \beta_j^2 \cos(t\theta_j)$$

- Problem: find values of θ_j, for $j = 1, \cdots, n$

- Compute cosine transform of f; For $p = 0, \cdots, M$:

$$F(p) = \frac{f(0) + (-1)^p f(M)}{2} + \sum_{k=1}^{M-1} f(k) \cos \frac{kp\pi}{M},$$

- If f has an eigenvalue $\lambda = \cos\theta$, then component $\cos(\theta t)$, revealed by a peak at the point

$$p = \frac{l\theta}{\pi}\,.$$

- Peak at p_j corresponds to eigenvalue $\lambda_j = \cos\theta_j$ with $\theta_j = (p_j/M)\pi$, and so,

$$\lambda_j = \cos(\theta_j) = \cos(p_j\pi/M)$$

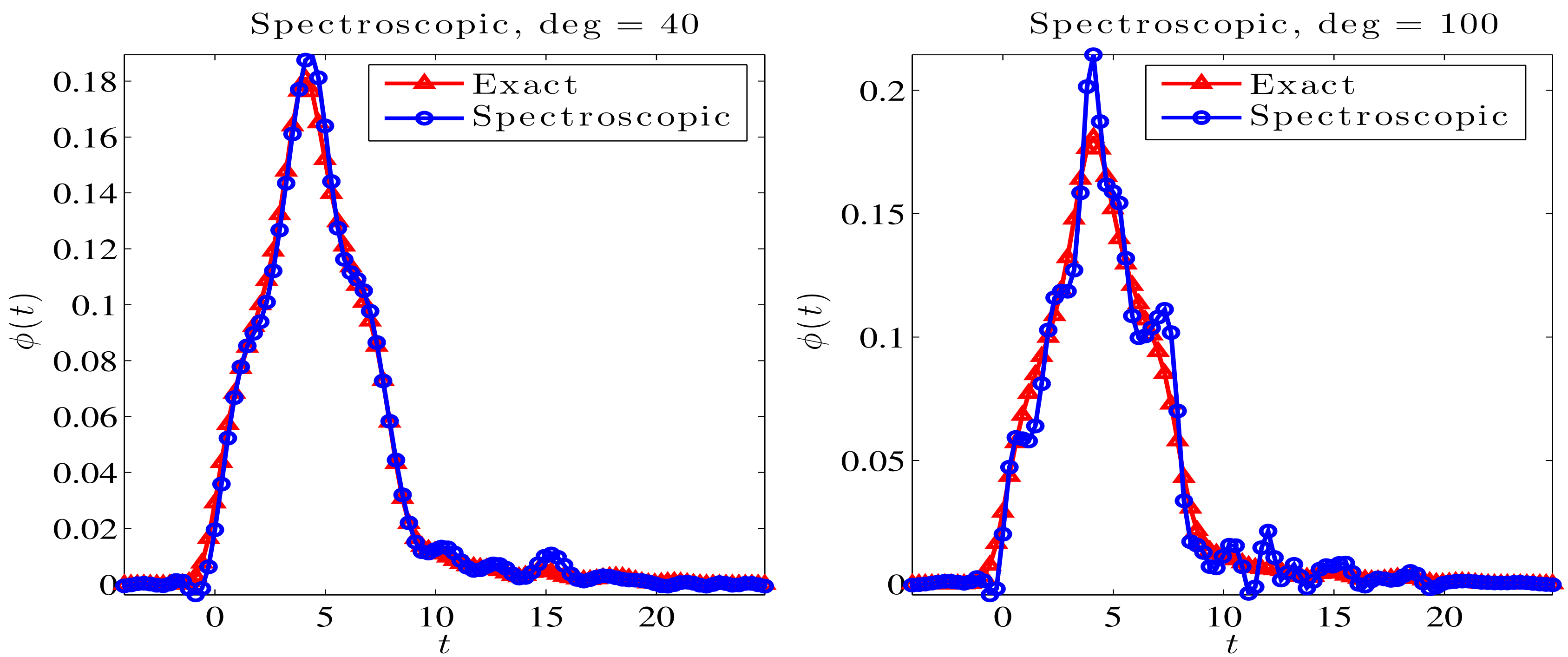
- For a sequence of random vectors compute

$$\hat{F}(\hat{p}) \equiv F\left(\frac{M}{\pi}\arccos\hat{p}\right), \quad \hat{p} = \cos(p\pi/M), p = 0 : M.$$

- Average these values $\rightarrow \phi(t_i) \approx Cst \times \hat{F}(t_i)$

The Lanczos Spectroscopic approach: Example

- Same example as before



Left: Degree 40*;* *Right: degree* 100

Delta Chebyshev

- The Lanczos spectroscopic approach suggests a 'new' idea:
  - Select 'mesh points' t_i on the interval $[-1,\ 1]$ of eigenvalues (still assume $\Lambda(A) \subseteq [-1,\ 1]$).
  - At each point expand the δ function in Chebyshev polynomials.
  - Add the results.
- Each δ-function defined at t_i acts as a 'spectral probe' [Presence of an eigenvalue at t_i can be detected by the value of $\int \delta(t - t_i) dt$ == 1 if $t_i \in \Lambda(A), 0$ otherwise.]
- It turns out that the method just defined is mathematically equivalent to KPM.

Delta-Gauss Legendre

- Idea: Instead of approximating ϕ directly, first select a representative ϕ_σ of ϕ for a given σ and then approximate ϕ_σ.

- ϕ_σ is a 'surrogate' for ϕ. Obtained by replacing δ_λ by :

$$h_\sigma(\lambda - t) = \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left[-\frac{(\lambda - t)^2}{2\sigma^2}\right].$$

- Goal: to expand into Legendre polynomials $L_k(\lambda)$

- With normalization factor expansion is written as:

$$h_\sigma(\lambda - t) = \frac{1}{(2\pi\sigma^2)^{1/2}} \sum_{k=0}^{\infty} \left(k + \frac{1}{2}\right) \gamma_k L_k(\lambda) \, .$$

- To determine the γ_k's we will also need to compute:

$$\psi_k = \int_{-1}^{1} L_k'(s) e^{-\frac{1}{2}((s-t)/\sigma)^2} ds.$$

Set $\zeta_k = e^{-\frac{1}{2}((1-t)/\sigma)^2} - (-1)^k e^{-\frac{1}{2}((1+t)/\sigma)^2}$.

- Then, for $k = 0, 1, \cdots$,:

$$\begin{cases} \gamma_{k+1} &= \frac{2k+1}{k+1}\left[\sigma^2(\psi_k - \zeta_k) + t\gamma_k\right] - \frac{k}{k+1}\gamma_{k-1} \\ \psi_{k+1} &= (2k+1)\gamma_k + \psi_{k-1}. \end{cases}$$

Initiialization: set $\gamma_{-1} = \psi_{-1} = 0$ $\psi_1 = \gamma_0$, and $\psi_0 = 0$ and:

$$\gamma_0 = \sigma\sqrt{\frac{\pi}{2}}\left[\text{erf}\left(\frac{1-t}{\sqrt{2}\sigma}\right) + \text{erf}\left(\frac{1+t}{\sqrt{2}\sigma}\right)\right],$$

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:

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

ALGORITHM : 1. *Lanczos*

1. *Choose start vector* v_1 *with* $\|v_1\|_2 = 1$.
2. *For* $j = 1, 2, \ldots, m$ *Do:*
3. $w_j := Av_j - \beta_j v_{j-1}, \qquad (\beta_1 \equiv 0, v_0 \equiv 0)$
4. $\alpha_j := (w_j, v_j)$
5. $w_j := w_j - \alpha_j v_j$
6. $\beta_{j+1} := \|w_j\|_2$. *If* $\beta_{j+1} = 0$ *then Stop*
7. $v_{j+1} := w_j / \beta_{j+1}$
8. *EndDo*

- Basis is such that $V_m^H A V_m = T_m$ - with

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

- Note: three term recurrence

$$\boxed{\beta_{j+1} v_{j+1} = A v_j - \alpha_j v_j - \beta_j v_{j-1}}$$

- Lanczos 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 [from 'outside in']
- 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 $\le 2m+1$

➤ Let, in the sense of distributions:

$$\langle \phi_{v_1}, p \rangle \equiv (p(A)v_1, v_1) = \sum \beta_i^2 p(\lambda_i) = \sum \beta_i^2 \langle \delta_{\lambda_i}, p \rangle$$

Then $\langle \phi_{v_1}, p \rangle \approx \sum a_i p(\theta_i) = \sum a_i \langle \delta_{\theta_i}, p \rangle \rightarrow$

$$\phi_{v_1} \approx \sum a_i \delta_{\theta_i}$$

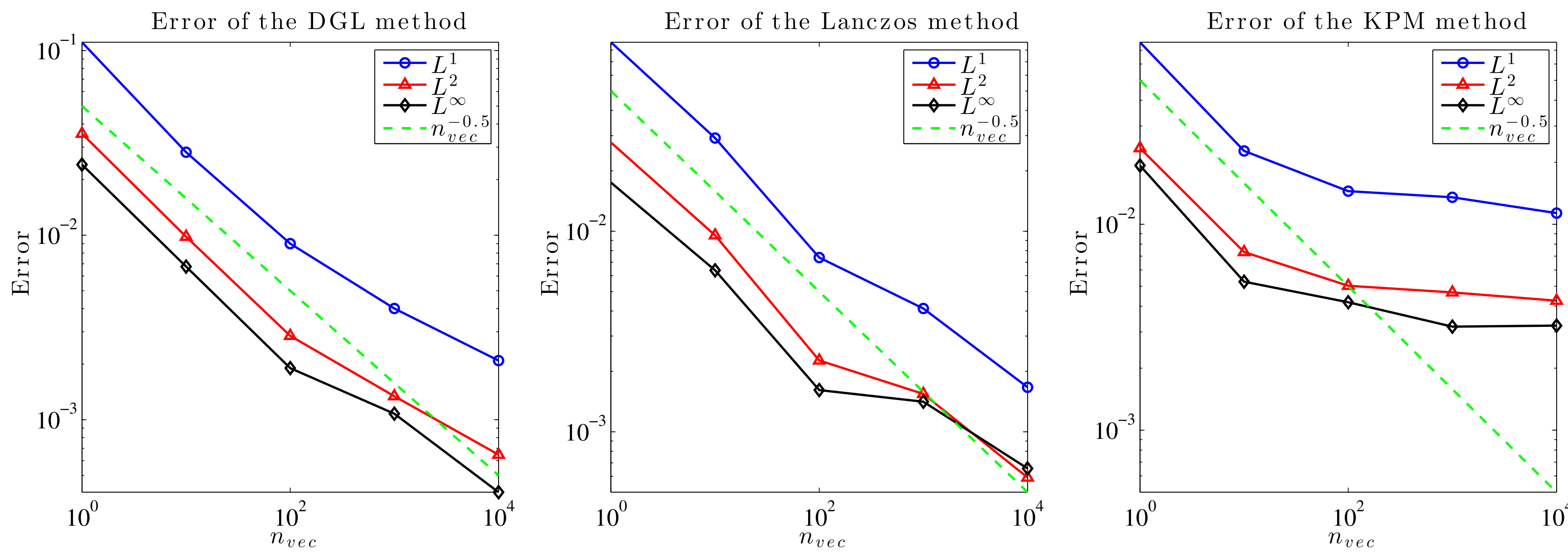
➤ Use several vectors v_1 and average results

Experiments

- Goal: to compare errors for similar number of matrix-vector products
- Example: Kohn-Sham Hamiltonian associated with a benzene molecule generated 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



The L^1, L^2 and L^∞ errors for the DGL , Lanczos, and the KPM methods with varying number of random vectors used (n_{vec}). Same model midified Laplacian. We set $\sigma = 0.56$.

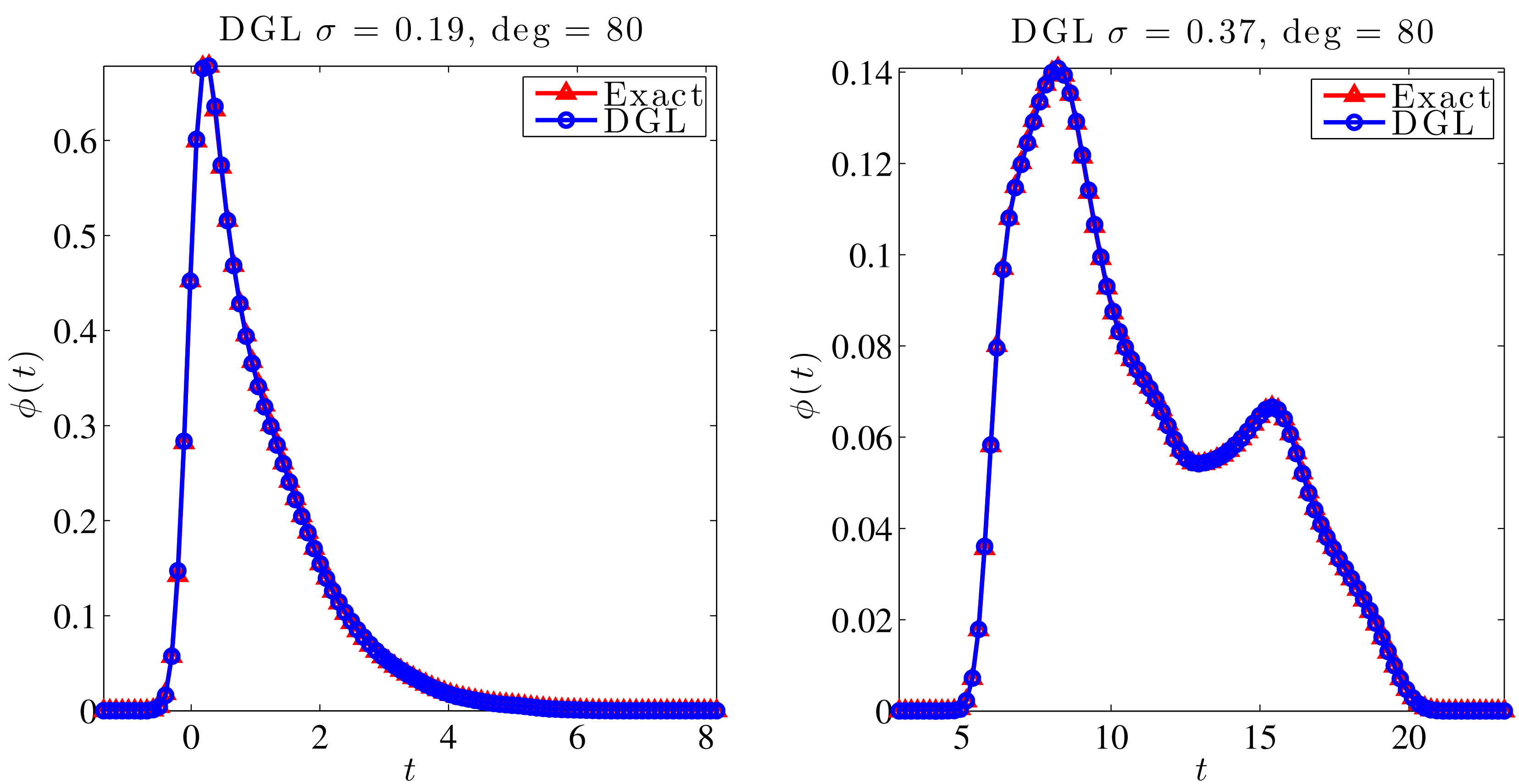
Other matrices

Matrix	n	λ_1	λ_n
$Ga_{10}As_{10}H_{30}$	113,081	-1.2	1.3×10^3
PE3K	9,000	8.1×10^{-6}	1.3×10^2
CFD1	70,656	2.0×10^{-5}	6.8
SHWATER	81,920	5.8	2.0×10^1

Description of the size and the spectrum range of the test matrices.

Matrix	Method	L^1 error	L^2 error	L^∞ error
$Ga_{10}As_{10}H_{30}$	DGL	3.937e-03	3.214e-04	4.301e-05
	Lanczos	4.828e-03	3.940e-04	5.452e-05
PE3K	DGL	4.562e-03	7.368e-04	3.143e-04
	Lanczos	5.459e-03	7.372e-04	3.294e-04
CFD1	DGL	2.276e-03	1.299e-03	1.746e-03
	Lanczos	2.024e-03	1.286e-03	2.478e-03
SHWATER	DGL	3.779e-03	1.282e-03	9.328e-04
	Lanczos	3.047e-03	9.829e-04	6.100e-04

L^1, L^2, and L^∞ error associated with the approximate spectral densities produced by the DGL and Lanczos methods for different test matrices.



Approximate spectral densities of CFD1 and SHWATER matrices obtained by DGL along with exact smoothed ones

Conclusion

- ➤ Probabilistic algorithms provide powerful tools for solving various problems: eigenvalue counts, DOS, Diag $(f(A))$..

- ➤ Most of the algorithms we discussed rely on estimating trace of $f(A)$ or Diag($f(A)$).

- ➤ Analysis left to do: adapt known decay bounds (Benzi al,..) to analyze convergence

- ➤ Also: Can we do better than random sampling [e.g., probing,..]?

- ➤ Physicists are interested in modified forms of the density of states. $\rightarrow$ Explore extentions of what we did.