Lanczos-based typicality methods for Quantum Thermodynamics

Tyler Chen

October 19, 2023

`chen.pw/slides`

This talk

Topic: We'll see some recent progress on the design and analysis of typicality methods for spectral densities.

Throughout: I'll try to provide an accessible introduction to ideas from numerical analysis that might be relevant to computational physicists.

Takeaway: numerical analysis and computational physics can benefit from more collaboration.

What is a matrix function?

A $d \times d$ symmetric matrix $\mathbf{H}$ has real eigenvalues and orthonormal eigenvectors:

$$\mathbf{H} = \sum_{n=1}^{d} \lambda_n |\mathbf{u}_n\rangle\langle\mathbf{u}_n|.$$

The matrix function $f(\mathbf{H})$, induced by $f : \mathbb{R} \to \mathbb{R}$ and $\mathbf{A}$, is defined as

$$f(\mathbf{H}) = \sum_{n=1}^{d} f(\lambda_n) |\mathbf{u}_n\rangle\langle\mathbf{u}_n|$$

In this talk, think of the dimension d as big! E.g. $d = 10^6$ or $d = 10^{10}$, etc.

What is a matrix function?

A $d \times d$ symmetric matrix $\mathbf{H}$ has real eigenvalues and orthonormal eigenvectors:

$$\mathbf{H} = \sum_{n=1}^{d} \lambda_n |\mathbf{u}_n\rangle\langle\mathbf{u}_n|.$$

The matrix function $f(\mathbf{H})$, induced by $f : \mathbb{R} \to \mathbb{R}$ and $\mathbf{A}$, is defined as

$$f(\mathbf{H}) = \sum_{n=1}^{d} f(\lambda_n) |\mathbf{u}_n\rangle\langle\mathbf{u}_n|$$

In this talk, think of the dimension d as big! E.g. $d = 10^6$ or $d = 10^{10}$, etc.

What do we want?

Often, we don't need $f(\mathbf{H})$ itself. In this talk we will discuss:

$$f(\mathbf{H})\mathbf{v}, \qquad \mathbf{v}^\mathsf{T} f(\mathbf{H})\mathbf{v}, \qquad \operatorname{tr}(f(\mathbf{H})) = \sum_{n=1}^{d} f(\lambda_n)$$

Example. If $f(x) = x^{-1}$, then $f(\mathbf{H}) = \mathbf{A}^{-1}$ and $f(\mathbf{H})\mathbf{v} = \mathbf{A}^{-1}\mathbf{v}$ is the solution to the linear system $\mathbf{A}\mathbf{x} = \mathbf{v}$.

- More computationally efficient to compute an approximation to the solution $\mathbf{A}^{-1}\mathbf{v}$ rather than computing $\mathbf{A}^{-1}$ and then multiplying with $\mathbf{v}$.
  - Even if $\mathbf{A}$ is sparse, $f(\mathbf{H})$ is typically dense. Storing a $n \times n$ dense matrix might be intractable.
  - $d = 2^{20} \approx 1\text{M} \implies n \times n$ dense matrix requires 8.8 terrabytes of storage

What do we want?

Often, we don't need $f(\mathbf{H})$ itself. In this talk we will discuss:

$$f(\mathbf{H})\mathbf{v}, \qquad \mathbf{v}^\top f(\mathbf{H})\mathbf{v}, \qquad \operatorname{tr}(f(\mathbf{H})) = \sum_{n=1}^{d} f(\lambda_n)$$

Example. If $f(x) = x^{-1}$, then $f(\mathbf{H}) = \mathbf{A}^{-1}$ and $f(\mathbf{H})\mathbf{v} = \mathbf{A}^{-1}\mathbf{v}$ is the solution to the linear system $\mathbf{A}\mathbf{x} = \mathbf{v}$.

- More computationally efficient to compute an approximation to the solution $\mathbf{A}^{-1}\mathbf{v}$ rather than computing $\mathbf{A}^{-1}$ and then multiplying with $\mathbf{v}$.
  - Even if $\mathbf{A}$ is sparse, $f(\mathbf{H})$ is typically dense. Storing a $n \times n$ dense matrix might be intractable.
  - $d = 2^{20} \approx 1\text{M} \implies n \times n$ dense matrix requires 8.8 terrabytes of storage

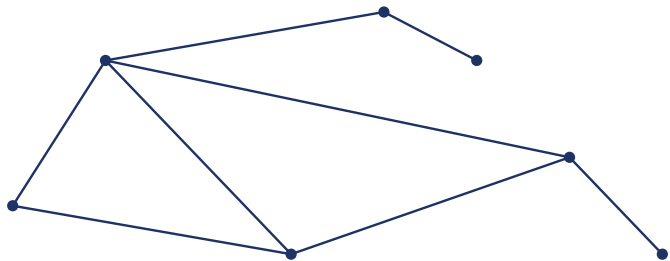
Applications

Applications in many fields: physics, chemistry, biology, statistics, high performance computing, machine learning, etc.

Common functions: inverse, exponential, square root, sign function.

Example application: network science

Let G be a graph (nodes and edges). How many triangles are there?

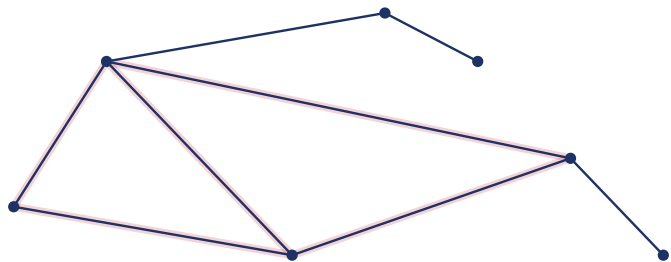


Fact. If $\mathbf{A}$ is the adjacency matrix for G, then

$$\#\text{ of triangles in } G = \frac{\operatorname{tr}(\mathbf{A}^3)}{6}.$$

Example application: network science

Let G be a graph (nodes and edges). How many triangles are there?

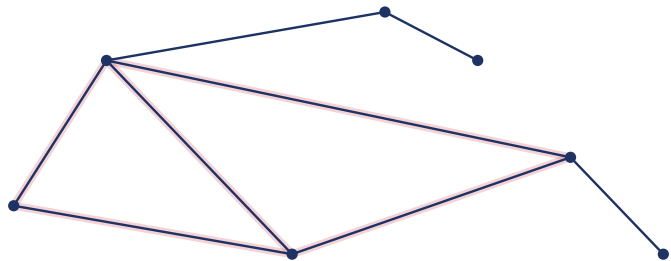


Fact. If $\mathbf{A}$ is the adjacency matrix for G, then

$$\#\text{ of triangles in } G = \frac{\operatorname{tr}(\mathbf{A}^3)}{6}.$$

Example application: network science

Let G be a graph (nodes and edges). How many triangles are there?



Fact. If $\mathbf{A}$ is the adjacency matrix for G, then

$$\#\text{ of triangles in } G = \frac{\operatorname{tr}(\mathbf{A}^3)}{6}.$$

Example application: high performance computing

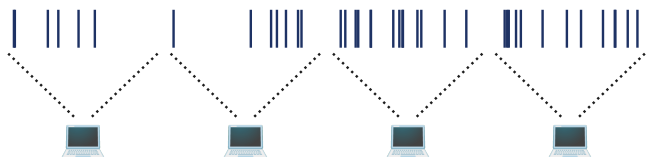
State of the art parallel eigensolvers such as FEAST and EVSL work by splitting the spectrum of $\mathbf{A}$ into pieces, which can each be solved on different machines in parallel.

Let $\mathbb{1}[a \leq x \leq b] = 1$ if $x \in [a, b]$ and 0 otherwise. Then

$$\#\text{ of eigenvalues in } [a, b] = \operatorname{tr}(\mathbb{1}[a \leq \mathbf{A} \leq b]).$$

Example application: high performance computing

State of the art parallel eigensolvers such as FEAST and EVSL work by splitting the spectrum of $\mathbf{A}$ into pieces, which can each be solved on different machines in parallel.

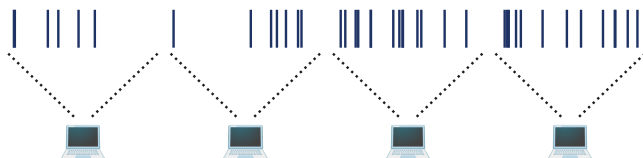


Let $\mathbb{1}[a \leq x \leq b] = 1$ if $x \in [a, b]$ and 0 otherwise. Then

$$\#\text{ of eigenvalues in } [a, b] = \operatorname{tr}(\mathbb{1}[a \leq \mathbf{A} \leq b]).$$

Example application: high performance computing

State of the art parallel eigensolvers such as FEAST and EVSL work by splitting the spectrum of $\mathbf{A}$ into pieces, which can each be solved on different machines in parallel.

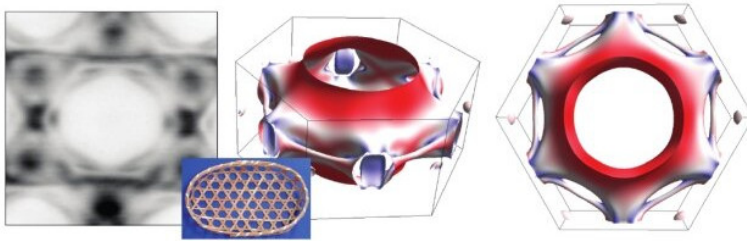


Let $\mathbb{1}[a \leq x \leq b] = 1$ if $x \in [a, b]$ and 0 otherwise. Then

$$\#\text{ of eigenvalues in } [a, b] = \operatorname{tr}(\mathbb{1}[a \leq \mathbf{A} \leq b]).$$

Example application: quantum thermodynamics

Let $\mathbf{A}$ be the Hamiltonian of a quantum system.



If the system is held in thermal equilibrium at inverse temperature $\beta = k_B/T$, then thermodynamic observables such as the specific heat, magnetization, heat capacity, etc. can be obtained from the partition function:

$$Z(\beta) = \mathrm{tr}(\exp(-\beta \mathbf{A})).$$

[0] https://phys.org/news/2023-06-quantum-materials-electron.html

Part I

Algorithms and convergence theory

Spectral densities

Given $\mathbf{H}$ (Hamiltonian), we're interested in the density of states (DOS):

$$\rho(x) = \sum_{n=1}^{d} \frac{1}{d} \delta(x - \lambda_n)$$

We probably can't efficiently (in $\ll d^3$ time) compute $\rho(x)$. Why?

Note that

$$\operatorname{tr}(f(\mathbf{H})) = d \int f(x)\rho(x)\mathrm{d}x.$$

We might be interested in functions like:

$$f(x) = \exp(-\beta E), \qquad f(x) = \beta E \exp(-\beta E), \qquad f(x) = \ln(x).$$

Spectral densities

Given $\mathbf{H}$ (Hamiltonian), we're interested in the density of states (DOS):

$$\rho(x) = \sum_{n=1}^{d} \frac{1}{d} \delta(x - \lambda_n)$$

We probably can't efficiently (in $\ll d^3$ time) compute $\rho(x)$. Why?

Note that

$$\mathrm{tr}(f(\mathbf{H})) = d \int f(x)\rho(x)\mathrm{d}x.$$

We might be interested in functions like:

$$f(x) = \exp(-\beta E), \qquad f(x) = \beta E \exp(-\beta E), \qquad f(x) = \ln(x).$$

Weighted spectral densities

Given a state $|\mathbf{r}\rangle$, we can define the local density of states (LDOS)

$$\hat{\rho}(x) = \sum_{n=1}^{d} |\langle \mathbf{r} | \mathbf{u}_n \rangle|^2 \delta(x - \lambda_n).$$

Note that

$$\langle \mathbf{r} | f(\mathbf{H}) | \mathbf{r} \rangle = \int f(x) \hat{\rho}(x) \mathrm{d}x.$$

We still can't efficiently compute $\hat{\rho}(x)$, but we can efficiently compute moments:

$$\langle \mathbf{r} | \mathbf{H}^k | \mathbf{r} \rangle = \int x^k \hat{\rho}(x) \mathrm{d}x$$

Can compute moments through degree s using $s/2$ matrix-vector products with $\mathbf{H}$.

Weighted spectral densities

Given a state $|\mathbf{r}\rangle$, we can define the local density of states (LDOS)

$$\hat{\rho}(x) = \sum_{n=1}^{d} |\langle \mathbf{r}|\mathbf{u}_n\rangle|^2 \delta(x - \lambda_n).$$

Note that

$$\langle \mathbf{r}|f(\mathbf{H})|\mathbf{r}\rangle = \int f(x)\hat{\rho}(x)\mathrm{d}x.$$

We still can't efficiently compute $\hat{\rho}(x)$, but we can efficiently compute moments:

$$\langle \mathbf{r}|\mathbf{H}^k|\mathbf{r}\rangle = \int x^k \hat{\rho}(x)\mathrm{d}x$$

Can compute moments through degree s using $s/2$ matrix-vector products with $\mathbf{H}$.

Weighted spectral densities

Note that we can compute moments through degree s using $s/2$ matrix-vector products with $\mathbf{H}$:

Iteratively compute

$$|\mathbf{r}\rangle, \quad \mathbf{H}|\mathbf{r}\rangle, \quad \mathbf{H}^2|\mathbf{r}\rangle = \mathbf{H}(\mathbf{H}|\mathbf{r}\rangle), \quad \ldots$$

Then use $\mathbf{H}^i|\mathbf{r}\rangle$ and $\mathbf{H}^j|\mathbf{r}\rangle$ to compute

$$\langle \mathbf{r}|\mathbf{H}^j\mathbf{H}^i|\mathbf{r}\rangle = \langle \mathbf{r}|\mathbf{H}^{i+j}|\mathbf{r}\rangle.$$

Typicality

If $|\mathbf{r}\rangle = \frac{1}{\sqrt{d}}(|\mathbf{u}_1\rangle + \cdots + |\mathbf{u}_d\rangle)$, then $|\langle \mathbf{r}|\mathbf{u}_n\rangle|^2 = d^{-1}$ and LDOS is exactly DOS.

Let $|\mathbf{r}\rangle$ be a (uniform) random state. By symmetry $|\langle \mathbf{r}|\mathbf{u}_n\rangle|^2$ all have the same distribution, so

$$|\langle \mathbf{r}|\mathbf{u}_n\rangle|^2 \approx d^{-1}$$

and hence

$$\hat{\rho}(x) \approx \rho(x).$$

Algorithmically, this lets us approxiamte DOS with LDOS (perhaps averaged over several random states).[1]

[1] can also be use for partial traces Chen and Cheng 2022

Implicit trace estimation

In numerical analysis and theoretical computer science we use this idea for trace estimation. Other distributions for $|\mathbf{r}\rangle$ are common (e.g. ± 1 entries, Gaussian entries).

If $|\mathbf{r}_1\rangle, \ldots, |\mathbf{r}_m\rangle$ are independent copies of $|\mathbf{r}\rangle$, we can get concentration inequalities[2] such as:

$$\mathbb{P}\left[\left|d^{-1}\operatorname{tr}(\mathbf{A}) - \frac{1}{m}\sum_{i=1}^{m}\langle \mathbf{r}_i|\mathbf{A}|\mathbf{r}_i\rangle\right| > \epsilon\right] < 2\exp\left(-C\frac{d\epsilon^2}{\|\mathbf{A}\|_2^2}\right).$$

This roughly says we can approximate $d^{-1}\operatorname{tr}(\mathbf{A})$ to accuracy ϵ using $O(d^{-1}\epsilon^{-2})$ matrix-vector products with $\mathbf{A}$.

[2] Reimann 2007; Popescu, Short, and Winter 2006; Avron and Toledo 2011; Roosta-Khorasani and Ascher 2014; Cortinovis and Kressner 2021.

Implicit trace estimation: beyond Monte Carlo

Recent trace estimation algorithms[3] can improve this to $O(d^{-1}\epsilon^{-1})$. These produce a low-rank approximation $\tilde{\mathbf{A}}$ to $\mathbf{A}$ and make use of the fact that

$$\operatorname{tr}(\mathbf{A}) = \operatorname{tr}(\tilde{\mathbf{A}}) + \operatorname{tr}(\mathbf{A} - \tilde{\mathbf{A}}).$$

This is closely related to deflation.[4]

A number of improvements:

- Practical parameters[5]
- More efficient deflation[6]
- What if $\mathbf{A} = f(\mathbf{H})$?[7]

[3] Meyer, Musco, Musco, and Woodruff 2021.

[4] Girard 1987; Weiße, Wellein, Alvermann, and Fehske 2006; Gambhir, Stathopoulos, and Orginos 2017.

[5] Persson, Cortinovis, and Kressner 2022.

[6] Epperly, Tropp, and Webber 2023.

[7] Persson and Kressner 2023; Chen and Hallman 2023.

Back to spectral densities: approximating a density from its moments

We can't (efficiently) compute LDOS $\hat{\rho}(x)$, but we can compute it's moments. How can we use this to approximate $\hat{\rho}(x)$ and in turn integrals against $\hat{\rho}(x)$?

Both KPM and SLQ address use the moment data to get approximations:

KPM: Approximate a function with it's Chebyshev approximation of degree s, then integrate this approximation using moment data.

SLQ: Construct a discrete approximation with k Diracs and use moment data to enforce that polynomials up to degree $2k-1$ are integrated exactly.

Back to spectral densities: approximating a density from its moments

We can't (efficiently) compute LDOS $\hat{\rho}(x)$, but we can compute it's moments. How can we use this to approximate $\hat{\rho}(x)$ and in turn integrals against $\hat{\rho}(x)$?

Both KPM and SLQ address use the moment data to get approximations:

KPM: Approximate a function with it's Chebyshev approximation of degree s, then integrate this approximation using moment data.

SLQ: Construct a discrete approximation with k Diracs and use moment data to enforce that polynomials up to degree $2k-1$ are integrated exactly.

The kernel polynomial method

Fix a reference density $\sigma(x)$ and let $\{p_n\}$ be the orthonormal polynomials:

$$\int p_n(x) p_m(x) \sigma(x) \mathrm{d}x = \delta_{mn}.$$

Expand the ratio $\hat{\rho}(x)/\sigma(x)$ in the orthogonal polynomial basis:

$$\frac{\hat{\rho}(x)}{\sigma(x)} = \sum_{n=0}^{\infty} \left(\int \frac{\hat{\rho}(x)}{\sigma(x)} p_n(x) \sigma(x) \mathrm{d}x \right) p_n(x) = \sum_{n=0}^{\infty} \left(\int p_n(x) \hat{\rho}(x) \mathrm{d}x \right) p_n(x).$$

Truncate this series at degree s and multiply by $\sigma(x)$:

$$\rho_{\mathrm{KPM}}(x) := \sigma(x) \sum_{n=0}^{s} \left(\int p_n(x) \hat{\rho}(x) \mathrm{d}x \right) p_n(x) = \sigma(x) \sum_{n=0}^{s} \langle \mathbf{r} | p_n(\mathbf{H}) | \mathbf{r} \rangle p_n(x).$$

Maybe also add damping to ensure approximation is non-negative.

The kernel polynomial method

Fix a reference density $\sigma(x)$ and let $\{p_n\}$ be the orthonormal polynomials:

$$\int p_n(x) p_m(x) \sigma(x) \mathrm{d}x = \delta_{mn}.$$

Expand the ratio $\hat{\rho}(x)/\sigma(x)$ in the orthogonal polynomial basis:

$$\frac{\hat{\rho}(x)}{\sigma(x)} = \sum_{n=0}^{\infty} \left(\int \frac{\hat{\rho}(x)}{\sigma(x)} p_n(x) \sigma(x) \mathrm{d}x \right) p_n(x) = \sum_{n=0}^{\infty} \left(\int p_n(x) \hat{\rho}(x) \mathrm{d}x \right) p_n(x).$$

Truncate this series at degree s and multiply by $\sigma(x)$:

$$\rho_{\mathrm{KPM}}(x) := \sigma(x) \sum_{n=0}^{s} \left(\int p_n(x) \hat{\rho}(x) \mathrm{d}x \right) p_n(x) = \sigma(x) \sum_{n=0}^{s} \langle \mathbf{r} | p_n(\mathbf{H}) | \mathbf{r} \rangle p_n(x).$$

Maybe also add damping to ensure approximation is non-negative.

The kernel polynomial method

Fix a reference density $\sigma(x)$ and let $\{p_n\}$ be the orthonormal polynomials:

$$\int p_n(x) p_m(x) \sigma(x) \mathrm{d}x = \delta_{mn}.$$

Expand the ratio $\hat{\rho}(x)/\sigma(x)$ in the orthogonal polynomial basis:

$$\frac{\hat{\rho}(x)}{\sigma(x)} = \sum_{n=0}^{\infty} \left(\int \frac{\hat{\rho}(x)}{\sigma(x)} p_n(x) \sigma(x) \mathrm{d}x \right) p_n(x) = \sum_{n=0}^{\infty} \left(\int p_n(x) \hat{\rho}(x) \mathrm{d}x \right) p_n(x).$$

Truncate this series at degree s and multiply by $\sigma(x)$:

$$\rho_{\mathrm{KPM}}(x) := \sigma(x) \sum_{n=0}^{s} \left(\int p_n(x) \hat{\rho}(x) \mathrm{d}x \right) p_n(x) = \sigma(x) \sum_{n=0}^{s} \langle \mathbf{r} | p_n(\mathbf{H}) | \mathbf{r} \rangle p_n(x).$$

Maybe also add damping to ensure approximation is non-negative.

How do we compute the moments?

The main computational cost is to compute the moments $\langle \mathbf{r} | p_n(\mathbf{H}) | \mathbf{r} \rangle$.

A common reference density[8] is $\sigma(x) \propto (1+x)^{-1/2}(1-x)^{-1/2}$ in which case the orthongonal polynomials are (up to scaling) the Chebyshev polynomials:

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \qquad T_1(x) = 2x, \quad T_0(x) = 1.$$

One can compute $T_n(\mathbf{H})|\mathbf{r}\rangle$ by

$$T_n(\mathbf{H})|\mathbf{r}\rangle = 2\mathbf{H}T_{n-1}(\mathbf{H})|\mathbf{r}\rangle - T_{n-2}(\mathbf{H})|\mathbf{r}\rangle, \qquad T_1(\mathbf{H})|\mathbf{r}\rangle = 2\mathbf{H}|\mathbf{r}\rangle, \quad T_0(\mathbf{H})|\mathbf{r}\rangle = |\mathbf{r}\rangle.$$

To get additional cost saving, use the identities

$$T_{2n}(x) = 2T_n(x)^2 - 1, \qquad T_{2n+1}(x) = 2T_{n+1}(x)T_n(x) - T_1(x).$$

[8] To use this density, one must scale $\mathbf{H}$ so the spectrum is contained in $[-1, 1]$.

How do we compute the moments?

The main computational cost is to compute the moments $\langle \mathbf{r}|p_n(\mathbf{H})|\mathbf{r}\rangle$.

A common reference density[8] is $\sigma(x) \propto (1+x)^{-1/2}(1-x)^{-1/2}$ in which case the orthongonal polynomials are (up to scaling) the Chebyshev polynomials:

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \qquad T_1(x) = 2x, \quad T_0(x) = 1.$$

One can compute $T_n(\mathbf{H})|\mathbf{r}\rangle$ by

$$T_n(\mathbf{H})|\mathbf{r}\rangle = 2\mathbf{H}T_{n-1}(\mathbf{H})|\mathbf{r}\rangle - T_{n-2}(\mathbf{H})|\mathbf{r}\rangle, \qquad T_1(\mathbf{H})|\mathbf{r}\rangle = 2\mathbf{H}|\mathbf{r}\rangle, \quad T_0(\mathbf{H})|\mathbf{r}\rangle = |\mathbf{r}\rangle.$$

To get additional cost saving, use the identities

$$T_{2n}(x) = 2T_n(x)^2 - 1, \qquad T_{2n+1}(x) = 2T_{n+1}(x)T_n(x) - T_1(x).$$

[8] To use this density, one must scale $\mathbf{H}$ so the spectrum is contained in $[-1, 1]$.

How do we compute the moments?

The main computational cost is to compute the moments $\langle \mathbf{r}|p_n(\mathbf{H})|\mathbf{r}\rangle$.

A common reference density[8] is $\sigma(x) \propto (1+x)^{-1/2}(1-x)^{-1/2}$ in which case the orthongonal polynomials are (up to scaling) the Chebyshev polynomials:

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x), \qquad T_1(x) = 2x, \quad T_0(x) = 1.$$

One can compute $T_n(\mathbf{H})|\mathbf{r}\rangle$ by

$$T_n(\mathbf{H})|\mathbf{r}\rangle = 2\mathbf{H}T_{n-1}(\mathbf{H})|\mathbf{r}\rangle - T_{n-2}(\mathbf{H})|\mathbf{r}\rangle, \qquad T_1(\mathbf{H})|\mathbf{r}\rangle = 2\mathbf{H}|\mathbf{r}\rangle, \quad T_0(\mathbf{H})|\mathbf{r}\rangle = |\mathbf{r}\rangle.$$

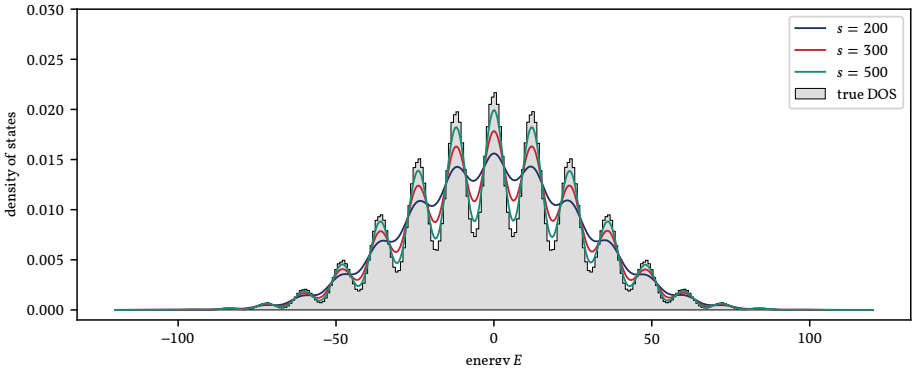
To get additional cost saving, use the identities

$$T_{2n}(x) = 2T_n(x)^2 - 1, \qquad T_{2n+1}(x) = 2T_{n+1}(x)T_n(x) - T_1(x).$$

[8] To use this density, one must scale $\mathbf{H}$ so the spectrum is contained in $[-1, 1]$.

Numerical Example

The higher the degree s, the better the approximation: resolution$\sim s^{-1}$.



Cost to get moments should be balanced how well LDOS approximates DOS.

Lanczos

The Lanczos algorithm iteratively produces an orthonormal basis $\{|\mathbf{v}_n\rangle\}$ for the Krylov subspace

$$\operatorname{span}\{|\mathbf{r}\rangle, \mathbf{H}|\mathbf{r}\rangle, \dots, \mathbf{H}^k|\mathbf{r}\rangle\} = \{p(\mathbf{H})|\mathbf{r}\rangle : \deg(p) \leq k\}. \tag{1}$$

This is done via a symmetric three-term recurrence

$$|\mathbf{v}_{n+1}\rangle = \frac{1}{\beta_n}\left(\mathbf{H}|\mathbf{v}_n\rangle - \alpha_n|\mathbf{v}_n\rangle - \beta_{n-1}|\mathbf{v}_{n-1}\rangle\right) \tag{2}$$

with initial conditions $|\mathbf{v}_1\rangle = (1/\beta_0)(\mathbf{H}|\mathbf{v}_0\rangle - \alpha_0|\mathbf{v}_0\rangle)$ and $|\mathbf{v}_0\rangle = |\mathbf{r}\rangle$.

At each step α_n is chosen so that $\langle\mathbf{v}_{n+1}|\mathbf{v}_n\rangle = 0$ and then β_n is chosen so that $\langle\mathbf{v}_{n+1}|\mathbf{v}_{n+1}\rangle = 1$.

Lanczos

The Lanczos algorithm iteratively produces an orthonormal basis $\{|\mathbf{v}_n\rangle\}$ for the Krylov subspace

$$\operatorname{span}\{|\mathbf{r}\rangle, \mathbf{H}|\mathbf{r}\rangle, \dots, \mathbf{H}^k|\mathbf{r}\rangle\} = \{p(\mathbf{H})|\mathbf{r}\rangle : \deg(p) \leq k\}. \tag{1}$$

This is done via a symmetric three-term recurrence

$$|\mathbf{v}_{n+1}\rangle = \frac{1}{\beta_n}\left(\mathbf{H}|\mathbf{v}_n\rangle - \alpha_n|\mathbf{v}_n\rangle - \beta_{n-1}|\mathbf{v}_{n-1}\rangle\right) \tag{2}$$

with initial conditions $|\mathbf{v}_1\rangle = (1/\beta_0)(\mathbf{H}|\mathbf{v}_0\rangle - \alpha_0|\mathbf{v}_0\rangle)$ and $|\mathbf{v}_0\rangle = |\mathbf{r}\rangle$.

At each step α_n is chosen so that $\langle\mathbf{v}_{n+1}|\mathbf{v}_n\rangle = 0$ and then β_n is chosen so that $\langle\mathbf{v}_{n+1}|\mathbf{v}_{n+1}\rangle = 1$.

Lanczos

We can write this in matrix form: $\mathbf{HV} = \mathbf{VH}_k + |\mathbf{v}\rangle\langle\mathbf{e}_k|$

$$\mathbf{H}\begin{bmatrix} | & | & & | \\ \mathbf{v}_0 & \mathbf{v}_1 & \cdots & \mathbf{v}_k \\ | & | & & | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \mathbf{v}_0 & \mathbf{v}_1 & \cdots & \mathbf{v}_k \\ | & | & & | \end{bmatrix}\begin{bmatrix} \alpha_0 & \beta_0 & & \\ \beta_0 & \alpha_1 & \ddots & \\ & \ddots & \ddots & \beta_{n-1} \\ & & \beta_{n-1} & \alpha_k \end{bmatrix} + \beta_k|\mathbf{q}_{n+1}\rangle\langle\mathbf{e}_k|.$$

The orthogonality of the $\{|\mathbf{v}_n\rangle\}$ implies:

$$\mathbf{H}_k = \mathbf{V}^\mathsf{T}\mathbf{HV}.$$

A distribution function?

Define

$$\rho_{\mathrm{SLQ}}(x) = \sum_{n=1}^{k} |\langle \mathbf{s}_n | \mathbf{e}_n \rangle|^2 \delta(x - \theta_n),$$

where θ_n are the eigenvalues of $\mathbf{H}_k$ and $\mathbf{s}_n$ are the eigenvectors. Since this is a discrete distribution, it is common to replace $\delta(x - \theta_n)$ with a blurred version (i.e. a Gaussian of a given width).

Note that

$$\int f(x)\rho_{\mathrm{SLQ}}(x)\mathrm{d}x = \langle \mathbf{e}_1 | f(\mathbf{H}_k) | \mathbf{e}_1 \rangle.$$

SLQ moments match LDOS momements

Let p be any polynomial of degree at most $2k-1$. Then

$$\langle \mathbf{r}|p(\mathbf{H})|\mathbf{r}\rangle = \int \hat{\rho}(e)p(x)\mathrm{d}x = \int \rho_{\mathrm{SLQ}}(x)p(E)\mathrm{d}x = \langle \mathbf{e}_1|p(\mathbf{H}_k)|\mathbf{e}_1\rangle.$$

Proof: Suppose $\mathbf{H}^{n-1}|\mathbf{r}\rangle = \mathbf{V}\mathbf{H}_k^{n-1}|\mathbf{e}_1\rangle$. Since $|\mathbf{r}\rangle = \mathbf{V}|\mathbf{e}_1\rangle$, write

$$\mathbf{H}^n|\mathbf{r}\rangle = \mathbf{H}\mathbf{V}\mathbf{H}_k^{n-1}|\mathbf{e}_1\rangle = \mathbf{V}\mathbf{H}_k^n|\mathbf{e}_1\rangle + |\mathbf{v}\rangle\langle \mathbf{e}_k|\mathbf{H}_k^n|\mathbf{e}_1\rangle = \mathbf{V}_k\mathbf{H}_k^n|\mathbf{e}_1\rangle.$$

In last equality: since $\mathbf{H}_k$ is tridiagonal, $\mathbf{H}_k^n$ has bandwidth $2n+1$ and $\langle \mathbf{e}_k|\mathbf{H}_k^n|\mathbf{e}_1\rangle = 0$ provided $n < k$.

Now use $\mathbf{V}^\mathsf{T}\mathbf{V} = \mathbf{I}$ and $\mathbf{V}^\mathsf{T}\mathbf{H}\mathbf{V} = \mathbf{H}_k$ to get $\langle \mathbf{r}|\mathbf{H}^n|\mathbf{r}\rangle$ for $n < 2k$.

SLQ moments match LDOS momements

Let p be any polynomial of degree at most $2k-1$. Then

$$\langle \mathbf{r}|p(\mathbf{H})|\mathbf{r}\rangle = \int \hat{\rho}(e)p(x)\mathrm{d}x = \int \rho_{\mathrm{SLQ}}(x)p(E)\mathrm{d}x = \langle \mathbf{e}_1|p(\mathbf{H}_k)|\mathbf{e}_1\rangle.$$

Proof: Suppose $\mathbf{H}^{n-1}|\mathbf{r}\rangle = \mathbf{V}\mathbf{H}_k^{n-1}|\mathbf{e}_1\rangle$. Since $|\mathbf{r}\rangle = \mathbf{V}|\mathbf{e}_1\rangle$, write

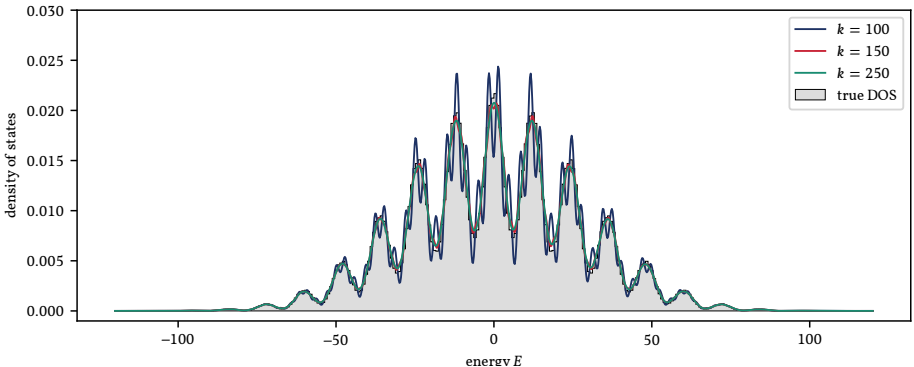
$$\mathbf{H}^n|\mathbf{r}\rangle = \mathbf{H}\mathbf{V}\mathbf{H}_k^{n-1}|\mathbf{e}_1\rangle = \mathbf{V}\mathbf{H}_k^n|\mathbf{e}_1\rangle + |\mathbf{v}\rangle\langle \mathbf{e}_k|\mathbf{H}_k^n|\mathbf{e}_1\rangle = \mathbf{V}_k\mathbf{H}_k^n|\mathbf{e}_1\rangle.$$

In last equality: since $\mathbf{H}_k$ is tridiagonal, $\mathbf{H}_k^n$ has bandwidth $2n+1$ and $\langle \mathbf{e}_k|\mathbf{H}_k^n|\mathbf{e}_1\rangle = 0$ provided $n < k$.

Now use $\mathbf{V}^\mathsf{T}\mathbf{V} = \mathbf{I}$ and $\mathbf{V}^\mathsf{T}\mathbf{H}\mathbf{V} = \mathbf{H}_k$ to get $\langle \mathbf{r}|\mathbf{H}^n|\mathbf{r}\rangle$ for $n < 2k$.

Numerical Example

The higher the degree $s = 2k - 1$, the better the approximation: resolution$\sim s^{-1}$.

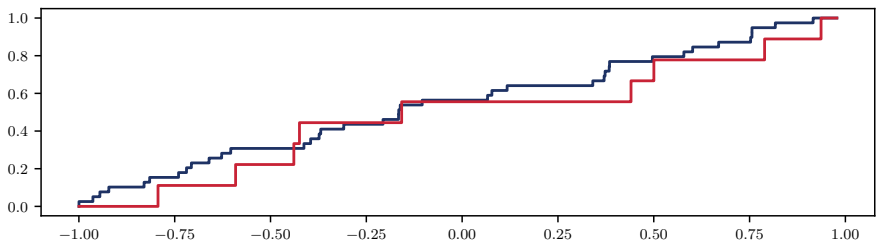


Cost to get moments should be balanced how well LDOS approximates DOS.

Measuring the similarity of distributions

The Wasserstein distance measures the similarity between distributions:

$$d_{\mathrm{W}}(\psi_1, \psi_2) = \int |\Psi_1(x) - \Psi_2(x)| \mathrm{d}x.$$



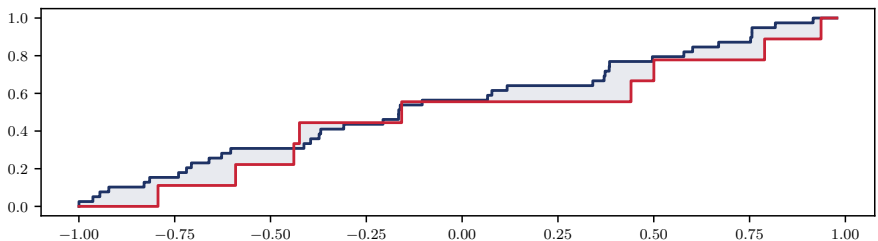
This is equivalent to

$$d_{\mathrm{W}}(\psi_1, \psi_2) = \max \left\{ \left| \int f(x)\psi_1(x)\mathrm{d}x - \int f(x)\psi_2(x)\mathrm{d}x \right| : |f(x) - f(y)| \leq |x - y| \; \forall x, y \right\}$$

Measuring the similarity of distributions

The Wasserstein distance measures the similarity between distributions:

$$d_{\mathrm{W}}(\psi_1, \psi_2) = \int |\Psi_1(x) - \Psi_2(x)| \mathrm{d}x.$$



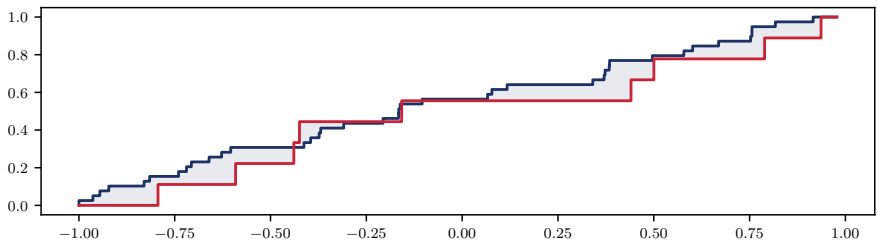
This is equivalent to

$$d_{\mathrm{W}}(\psi_1, \psi_2) = \max \left\{ \left| \int f(x)\psi_1(x)\mathrm{d}x - \int f(x)\psi_2(x)\mathrm{d}x \right| : |f(x) - f(y)| \leq |x - y| \; \forall x, y \right\}$$

Measuring the similarity of distributions

The Wasserstein distance measures the similarity between distributions:

$$d_{\mathrm{W}}(\psi_1, \psi_2) = \int |\Psi_1(x) - \Psi_2(x)| \mathrm{d}x.$$



This is equivalent to

$$d_{\mathrm{W}}(\psi_1, \psi_2) = \max \left\{ \left| \int f(x)\psi_1(x)\mathrm{d}x - \int f(x)\psi_2(x)\mathrm{d}x \right| : |f(x) - f(y)| \leq |x - y| \; \forall x, y \right\}$$

Theoretical analysis (high level)[9]

Fact: 1-Lipshitz functions can be approximated to accuracy ϵ with a degree $s = O(\epsilon^{-1})$ polynomial. This polynomial has decaying Chebyshev coefficients.

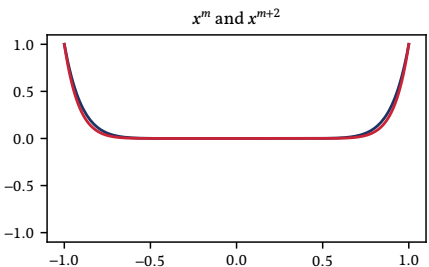
Fact: if two distributions have exactly the same moments through degree k, the the Wasserstein distance is $O(k^{-1})$.

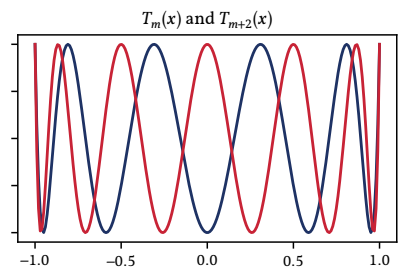
[9] Braverman, Krishnan, and Musco 2022; Chen, Trogdon, and Ubaru 2022.

Chebyshev moments vs monomial moments

While two distribution functions with exactly the same first k moments have Wasserstein distance $O(k^{-1})$, if the monomial moments are even a little different, the Wasserstein distance can be big.

Instead, one should look at Chebyshev moments, since Wasserstein distance is stable with respect to perturbations in these moments.

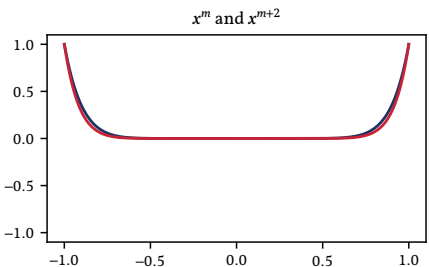


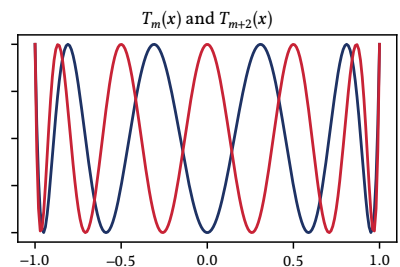


Chebyshev moments vs monomial moments

While two distribution functions with exactly the same first k moments have Wasserstein distance $O(k^{-1})$, if the monomial moments are even a little different, the Wasserstein distance can be big.

Instead, one should look at Chebyshev moments, since Wasserstein distance is stable with respect to perturbations in these moments.





Theoretical analysis (high level)[10]

Approach:

- Show KPM/SLQ approximation has almost the same Chebyshev moments as DOS (i.e. that Chebyshev polynomials are integrated almost exactly) through some degree.
- Use the fact above to show this implies all Lipshitz functions are integrated nearly correctly

For a single fixed Lipshitz function, there are easier approaches, but to get a Wasserstein bound, we need something that holds for all Lipshitz functions simultaneously.

[10] Braverman, Krishnan, and Musco 2022; Chen, Trogdon, and Ubaru 2022.

Theoretical analysis (high level)[10]

Approach:

- Show KPM/SLQ approximation has almost the same Chebyshev moments as DOS (i.e. that Chebyshev polynomials are integrated almost exactly) through some degree.
- Use the fact above to show this implies all Lipshitz functions are integrated nearly correctly

For a single fixed Lipshitz function, there are easier approaches, but to get a Wasserstein bound, we need something that holds for all Lipshitz functions simultaneously.

[10] Braverman, Krishnan, and Musco 2022; Chen, Trogdon, and Ubaru 2022.

Theoretical analysis (details sketch)

Claim. Suppose that for all $n = 0, 1, \dots, s$:

$$\left| \int T_n(x)(\psi_1(x) - \psi_2(x))\mathrm{d}x \right| \leq \eta.$$

Then, for any degree s polynomial $p_s(x) = c_0 + c_1 T_1(x) + \cdots + c_s T_s(x)$,

$$\left| \int f(x)(\psi_1(x) - \psi_2(x))\mathrm{d}x \right| \leq 2\|f(x) - p_s(x)\|_{[-1,1]} + 2\eta \sum_{n=1}^{s} |c_n|.$$

Proof. Triangle inequality:

$$\left| \int f(x)(\psi_1(x) - \psi_2(x))\mathrm{d}x \right| \leq \left| \int (f(x) - p_s(x))(\psi_1(x) - \psi_2(x))\mathrm{d}x \right| + \left| \int p_s(x)(\psi_1(x) - \psi_2(x))\mathrm{d}x \right|.$$

Theoretical analysis (details sketch)

Claim. Suppose that for all $n = 0, 1, \dots, s$:

$$\left| \int T_n(x)(\psi_1(x) - \psi_2(x))\mathrm{d}x \right| \leq \eta.$$

Then, for any degree s polynomial $p_s(x) = c_0 + c_1 T_1(x) + \cdots + c_s T_s(x)$,

$$\left| \int f(x)(\psi_1(x) - \psi_2(x))\mathrm{d}x \right| \leq 2\|f(x) - p_s(x)\|_{[-1,1]} + 2\eta \sum_{n=1}^{s} |c_n|.$$

Proof. Triangle inequality:

$$\left| \int f(x)(\psi_1(x) - \psi_2(x))\mathrm{d}x \right| \leq \left| \int (f(x) - p_s(x))(\psi_1(x) - \psi_2(x))\mathrm{d}x \right| + \left| \int p_s(x)(\psi_1(x) - \psi_2(x))\mathrm{d}x \right|.$$

Theoretical analysis (details sketch)

Fact.[11] Suppose $f(x)$ is 1-Lipshitz ($|f(x) - f(y)| \leq |x - y|$) and set $p_s(x)$ as the degree s Jackson's damped Chebyshev approximation to $f(x)$. Then,

$$\|f(x) - p_s(x)\|_{[-1,1]} \leq \frac{6}{s}, \qquad \left| \int p_s(x) T_n(x) \mu_T(x) \mathrm{d}x \right| \leq \frac{4}{\pi n}.$$

Thus, since $1 + 1/2 + 1/3 + \cdots 1/s \leq 1 + \ln(s)$,

$$\left| \int f(x)(\psi_1(x) - \psi_2(x)) \mathrm{d}x \right| \leq \frac{12}{s} + \frac{8 \ln(s) \eta}{\pi}.$$

Maximizing over f, we then get

$$s = O(\epsilon^{-1}),\ \eta = O(\ln(s)^{-1} \epsilon) \quad \Longrightarrow \quad d_{\mathrm{W}}(\psi_1, \psi_2) \leq \epsilon.$$

This gives us gurantees for SLQ (slight modification for damped KPM).

[11] Rivlin 1981; Trefethen 2019.

Theoretical analysis (details sketch)

Fact.[11] Suppose $f(x)$ is 1-Lipshitz ($|f(x) - f(y)| \leq |x - y|$) and set $p_s(x)$ as the degree s Jackson's damped Chebyshev approximation to $f(x)$. Then,

$$\|f(x) - p_s(x)\|_{[-1,1]} \leq \frac{6}{s}, \qquad \left| \int p_s(x) T_n(x) \mu_T(x) \mathrm{d}x \right| \leq \frac{4}{\pi n}.$$

Thus, since $1 + 1/2 + 1/3 + \cdots 1/s \leq 1 + \ln(s)$,

$$\left| \int f(x)(\psi_1(x) - \psi_2(x)) \mathrm{d}x \right| \leq \frac{12}{s} + \frac{8 \ln(s) \eta}{\pi}.$$

Maximizing over f, we then get

$$s = O(\epsilon^{-1}),\ \eta = O(\ln(s)^{-1}\epsilon) \quad \Longrightarrow \quad d_{\mathrm{W}}(\psi_1, \psi_2) \leq \epsilon.$$

This gives us gurantees for SLQ (slight modification for damped KPM).

[11] Rivlin 1981; Trefethen 2019.

Part II

Implementation and finite precision arithmetic

A spectrum adaptive KPM[12]

In the KPM, the only expensive computation was computing moments: $\langle \mathbf{r}|p_n(\mathbf{H})|\mathbf{r}\rangle$.

If we've compute $\mathbf{H}_k$ using Lanczos, then we know for polyniamsl $p(x)$ of degree $< 2k$:

$$\langle \mathbf{r}|p(\mathbf{H})|\mathbf{r}\rangle = \langle \mathbf{e}_1|p(\mathbf{H}_k)|\mathbf{e}_1\rangle.$$

So, we can use Lanczos to implement KPM!

This means we can test out lots of different reference densities $\sigma(x)$ for essentially free (i.e. without accessing $\mathbf{H}$ again).

[12] Chen 2023.

Demo

Some basic functionality is implemented in the `spectral_density` package.[13]

```
pip install spectral_density
```

The design paradigm for `spectral_density` is that computation and approximation should be decoupled. In particular, approximations are obtained in two steps:

- **computation**: repeatedly run the Lanczos algorithm on the matrix of interest with random starting vectors
- **approximation**: use the output of the previous step to obtain spectral density approximations

This package focuses only on the second step; users are free to use any Lanczos implementation for the first step.

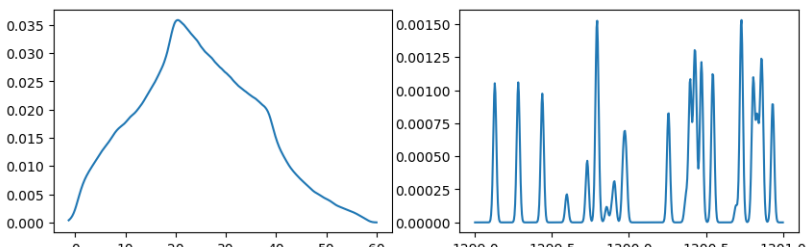
[13] `https://github.com/tchen-research/spectral_density`

Demo: setup

```
import spectral_density as spec

# import Hamiltonian
H = sp.io.mmread('./Ga41As41H72.mtx')
H.tocsr()
d = H.shape[0]

# run Lanczos several times
m = 3
αβ_list = []
for _ in range(m):
    v = np.random.randn(d)
    v /= np.linalg.norm(v)

    k = 150
    αβ_list.append(spec.lanczos(H,v,k,reorth=False))
```

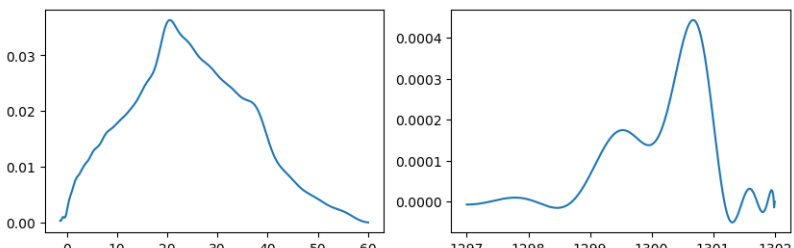
Demo: SLQ

```
ρ_SLQ = spec.SLQ(αβ_list)                # build SLQ instance

axs[0].plot(x,ρ_SLQ(x,width=.6))         # plot (specifying width)
axs[1].plot(x,ρ_SLQ(x,width=.01))
```



Demo: KPM

```
σ = spec.get_arcsin_density(-2,1302)    # specify reference density
ρ_KPM = spec.KPM(αβ_list,σ)             # build KPM instance

axs[0].plot(x,ρ_KPM(x))                 # plot
axs[1].plot(x,ρ_KPM(x))
```

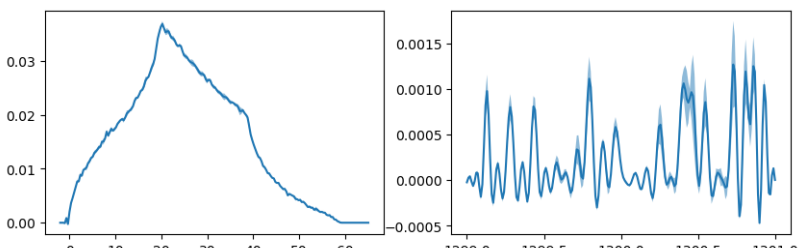


Demo: KPM

```
# use Lanczos output to determine two intervals containing spectrum
a_L = np.min(ρ_SLQ.θ)-4e-1
b_L = np.max(ρ_SLQ.θ[ρ_SLQ.θ<200])+4e-1
a_R = np.min(ρ_SLQ.θ[ρ_SLQ.θ>1200])-4e-1
b_R = np.max(ρ_SLQ.θ)+4e-1

# build a density on each interval
σ_L = spec.get_uniform_density(a_L,b_L)
σ_R = spec.get_semicircle_density(a_R,b_R)

# combine densities to specify reference density
σ = .95*σ_L + .05*σ_R
```

Demo: KPM

```
ρ_KPM = spec.KPM(αβ_list,σ)                    # build KPM instance

axs[0].plot(x,ρ_KPM(x))                        # plot
axs[1].plot(x,ρ_KPM(x))
```



Wait, isn't Lanczos unstable?

In the previous demo, we used the output of Lanczos without reorthogonalization!

There is a general fear of using Lanczos-based methods without expensive reorthogonalization schemes[14]

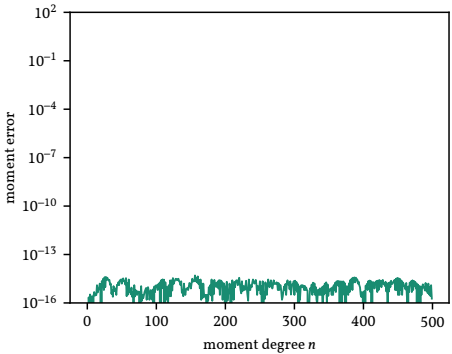
But… there is plenty of evidince that SLQ and related algorithms work fine without reorthogonalization:Long, Prelovšek, Shawish, Karadamoglou, and Zotos 2003; Schnack, Richter, and Steinigeweg 2020, etc.

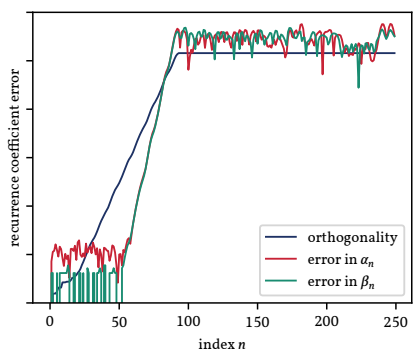
In fact, there is even theory.

[14] Jaklič and Prelovšek 1994; Aichhorn, Daghofer, Evertz, and Linden 2003; Weiße, Wellein, Alvermann, and Fehske 2006; Ubaru, Chen, and Saad 2017; Granziol, Wan, and Garipov 2019.

Numerical Example

People worry about a loss of orthogonality, and appearence of “ghost eigenvalues”. But do these impact the moments used for KPM?





Finite precision theory

In finite precision artihmetic, while $\mathbf{V}$ may no longer be orthogonal, we still have[15]

$$\mathbf{H}\mathbf{V} = \mathbf{V}\mathbf{H}_k + |\mathbf{v}\rangle\langle\mathbf{e}_k| + \mathbf{F}, \qquad \|\mathbf{F}\| = O\big(\epsilon_{\text{mach}}\,\text{poly}(k)\big).$$

From this, one can derive[16]

$$\big\|\tilde{T}_n(\mathbf{H})|\mathbf{r}\rangle - \mathbf{V}\tilde{T}_n(\mathbf{H}_k)|\mathbf{e}_1\rangle\big\| = O\big(\epsilon_{\text{mach}}\,\text{poly}(k)\big).$$

This can then be upgraded to[17]

$$\big|\langle\mathbf{r}|\tilde{T}_n(\mathbf{H})|\mathbf{r}\rangle - \langle\mathbf{e}_1|\tilde{T}_n(\mathbf{H}_k)|\mathbf{e}_1\rangle\big| = O\big(\epsilon_{\text{mach}}\,\text{poly}(k)\big).$$

In other words, SLQ's Chebyshev moments are still almost exact.

[15] Paige 1971; Paige 1976; Paige 1980.
[16] Druskin and Knizhnerman 1992; Musco, Musco, and Sidford 2018.
[17] Knizhnerman 1996.

Finite precision theory

In finite precision artihmetic, while $\mathbf{V}$ may no longer be orthogonal, we still have[15]

$$\mathbf{H}\mathbf{V} = \mathbf{V}\mathbf{H}_k + |\mathbf{v}\rangle\langle\mathbf{e}_k| + \mathbf{F}, \qquad \|\mathbf{F}\| = O\big(\epsilon_{\text{mach}}\,\text{poly}(k)\big).$$

From this, one can derive[16]

$$\big\|\tilde{T}_n(\mathbf{H})|\mathbf{r}\rangle - \mathbf{V}\tilde{T}_n(\mathbf{H}_k)|\mathbf{e}_1\rangle\big\| = O\big(\epsilon_{\text{mach}}\,\text{poly}(k)\big).$$

This can then be upgraded to[17]

$$\big|\langle\mathbf{r}|\tilde{T}_n(\mathbf{H})|\mathbf{r}\rangle - \langle\mathbf{e}_1|\tilde{T}_n(\mathbf{H}_k)|\mathbf{e}_1\rangle\big| = O\big(\epsilon_{\text{mach}}\,\text{poly}(k)\big).$$

In other words, SLQ's Chebyshev moments are still almost exact.

[15] Paige 1971; Paige 1976; Paige 1980.
[16] Druskin and Knizhnerman 1992; Musco, Musco, and Sidford 2018.
[17] Knizhnerman 1996.

A taste of how these anlyses work

Recall we have a perturbed recurrence: $\mathbf{HV} = \mathbf{VH}_k + |\mathbf{v}\rangle\langle\mathbf{e}_k| + \mathbf{F}$.

Define: $|\mathbf{t}_n\rangle = T_n(\mathbf{H})|\mathbf{r}\rangle, \quad |\bar{\mathbf{t}}_n\rangle = T_n(\mathbf{H}_k)|\mathbf{e}_1\rangle, \quad |\mathbf{d}_n\rangle = |\mathbf{t}_n\rangle - \mathbf{V}|\bar{\mathbf{t}}_n\rangle$.

Then, using that $\langle\mathbf{e}_k|\bar{\mathbf{t}}_{n-1}\rangle = 0$ (bc $\mathbf{H}_k$ is tridiagonal):

$$\begin{aligned}
|\mathbf{d}_n\rangle &= (2\mathbf{H}|\mathbf{t}_{n-1}\rangle - |\mathbf{t}_{n-2}\rangle) - (2\mathbf{V}\mathbf{H}_k|\bar{\mathbf{t}}_{n-1}\rangle - \mathbf{V}|\bar{\mathbf{t}}_{n-2}\rangle) \\
&= 2(\mathbf{H}|\mathbf{t}_{n-1}\rangle - (\mathbf{HV} - |\mathbf{v}\rangle\langle\mathbf{e}_k| - \mathbf{F})|\bar{\mathbf{t}}_{n-1}\rangle) - (|\mathbf{t}_{n-2}\rangle - \mathbf{V}|\bar{\mathbf{t}}_{n-2}\rangle) \\
&= 2\mathbf{H}|\mathbf{d}_{n-1}\rangle - |\mathbf{d}_{n-2}\rangle - \mathbf{F}|\bar{\mathbf{t}}_{n-1}\rangle
\end{aligned}$$

This is a perturbed Chebyshev recurrence. One can show:

$$|\mathbf{d}_n\rangle = U_{n-1}(\mathbf{H})\mathbf{F}|\bar{\mathbf{t}}_0\rangle + 2\sum_{i=2}^{n} U_{n-i}(\mathbf{H})\mathbf{F}|\bar{\mathbf{t}}_{i-1}\rangle.$$

Note that Cheyshev-U polynomials don't grow quickly, so this implies $|\mathbf{d}_n\rangle$ is small!

A taste of how these anlyses work

Recall we have a perturbed recurrence: $\mathbf{HV} = \mathbf{VH}_k + |\mathbf{v}\rangle\langle\mathbf{e}_k| + \mathbf{F}$.

Define: $|\mathbf{t}_n\rangle = T_n(\mathbf{H})|\mathbf{r}\rangle, \quad |\bar{\mathbf{t}}_n\rangle = T_n(\mathbf{H}_k)|\mathbf{e}_1\rangle, \quad |\mathbf{d}_n\rangle = |\mathbf{t}_n\rangle - \mathbf{V}|\bar{\mathbf{t}}_n\rangle$.

Then, using that $\langle\mathbf{e}_k|\bar{\mathbf{t}}_{n-1}\rangle = 0$ (bc $\mathbf{H}_k$ is tridiagonal):

$$\begin{aligned}
|\mathbf{d}_n\rangle &= (2\mathbf{H}|\mathbf{t}_{n-1}\rangle - |\mathbf{t}_{n-2}\rangle) - (2\mathbf{V}\mathbf{H}_k|\bar{\mathbf{t}}_{n-1}\rangle - \mathbf{V}|\bar{\mathbf{t}}_{n-2}\rangle) \\
&= 2(\mathbf{H}|\mathbf{t}_{n-1}\rangle - (\mathbf{HV} - |\mathbf{v}\rangle\langle\mathbf{e}_k| - \mathbf{F})|\bar{\mathbf{t}}_{n-1}\rangle) - (|\mathbf{t}_{n-2}\rangle - \mathbf{V}|\bar{\mathbf{t}}_{n-2}\rangle) \\
&= 2\mathbf{H}|\mathbf{d}_{n-1}\rangle - |\mathbf{d}_{n-2}\rangle - \mathbf{F}|\bar{\mathbf{t}}_{n-1}\rangle
\end{aligned}$$

This is a perturbed Chebyshev recurrence. One can show:

$$|\mathbf{d}_n\rangle = U_{n-1}(\mathbf{H})\mathbf{F}|\bar{\mathbf{t}}_0\rangle + 2\sum_{i=2}^{n} U_{n-i}(\mathbf{H})\mathbf{F}|\bar{\mathbf{t}}_{i-1}\rangle.$$

Note that Cheyshev-U polynomials don't grow quickly, so this implies $|\mathbf{d}_n\rangle$ is small!

Outlook

- While Lanczos is unstable, the instability has structure
- partial traces Chen and Cheng 2022; Chen, Chen, Li, Nzeuton, Pan, and Wang 2023

References I

Aichhorn, Markus et al. (Apr. 2003). "Low-temperature Lanczos method for strongly correlated systems". In: *Physical Review B* 67.16.

Avron, Haim and Sivan Toledo (Apr. 2011). "Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix". In: *Journal of the ACM* 58.2, pp. 1–34.

Braverman, Vladimir, Aditya Krishnan, and Christopher Musco (June 2022). *Sublinear time spectral density estimation*.

Chen, Tyler (2023). *A spectrum adaptive Kernel Polynomial Method*.

Chen, Tyler and Yu-Chen Cheng (2022). *Numerical computation of the equilibrium-reduced density matrix for strongly coupled open quantum systems*.

Chen, Tyler and Eric Hallman (Aug. 2023). "Krylov-Aware Stochastic Trace Estimation". In: *SIAM Journal on Matrix Analysis and Applications* 44.3, pp. 1218–1244.

Chen, Tyler, Thomas Trogdon, and Shashanka Ubaru (2022). *Randomized matrix-free quadrature for spectrum and spectral sum approximation*.

Chen, Tyler et al. (2023). *Faster randomized partial trace estimation*.

Cortinovis, Alice and Daniel Kressner (July 2021). "On Randomized Trace Estimates for Indefinite Matrices with an Application to Determinants". In: *Foundations of Computational Mathematics*.

Druskin, Vladimir and Leonid Knizhnerman (July 1992). "Error Bounds in the Simple Lanczos Procedure for Computing Functions of Symmetric Matrices and Eigenvalues". In: *Comput. Math. Math. Phys.* 31.7, pp. 20–30.

Epperly, Ethan N., Joel A. Tropp, and Robert J. Webber (2023). *XTrace: Making the most of every sample in stochastic trace estimation*.

References II

Gambhir, Arjun Singh, Andreas Stathopoulos, and Kostas Orginos (Jan. 2017). "Deflation as a Method of Variance Reduction for Estimating the Trace of a Matrix Inverse". In: *SIAM Journal on Scientific Computing* 39.2, A532–A558.

Girard, Didier (1987). *Un algorithme simple et rapide pour la validation croisée généralisée sur des problèmes de grande taille*.

Granziol, Diego, Xingchen Wan, and Timur Garipov (2019). *Deep Curvature Suite*.

Jaklič, J. and P. Prelovšek (Feb. 1994). "Lanczos method for the calculation of finite-temperature quantities in correlated systems". In: *Physical Review B* 49.7, pp. 5065–5068.

Knizhnerman, L. A. (Jan. 1996). "The Simple Lanczos Procedure: Estimates of the Error of the Gauss Quadrature Formula and Their Applications". In: *Comput. Math. Math. Phys.* 36.11, pp. 1481–1492.

Long, M. W. et al. (Dec. 2003). "Finite-temperature dynamical correlations using the microcanonical ensemble and the Lanczos algorithm". In: *Physical Review B* 68.23.

Meyer, Raphael A. et al. (Jan. 2021). "Hutch++: Optimal Stochastic Trace Estimation". In: *Symposium on Simplicity in Algorithms (SOSA)*. Society for Industrial and Applied Mathematics, pp. 142–155.

Musco, Cameron, Christopher Musco, and Aaron Sidford (2018). "Stability of the Lanczos Method for Matrix Function Approximation". In: *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms*. SODA '18. New Orleans, Louisiana: Society for Industrial and Applied Mathematics, pp. 1605–1624.

Paige, Christopher Conway (1971). "The computation of eigenvalues and eigenvectors of very large sparse matrices.". PhD thesis. University of London.

— (Dec. 1976). "Error Analysis of the Lanczos Algorithm for Tridiagonalizing a Symmetric Matrix". In: *IMA Journal of Applied Mathematics* 18.3, pp. 341–349.

References III

Paige, Christopher Conway (1980). "Accuracy and effectiveness of the Lanczos algorithm for the symmetric eigenproblem". In: *Linear Algebra and its Applications* 34, pp. 235–258.

Persson, David, Alice Cortinovis, and Daniel Kressner (July 2022). "Improved Variants of the Hutch++ Algorithm for Trace Estimation". In: *SIAM Journal on Matrix Analysis and Applications* 43.3, pp. 1162–1185.

Persson, David and Daniel Kressner (June 2023). "Randomized Low-Rank Approximation of Monotone Matrix Functions". In: *SIAM Journal on Matrix Analysis and Applications* 44.2, pp. 894–918.

Popescu, Sandu, Anthony J. Short, and Andreas Winter (Oct. 2006). "Entanglement and the foundations of statistical mechanics". In: *Nature Physics* 2.11, pp. 754–758.

Reimann, Peter (Oct. 2007). "Typicality for Generalized Microcanonical Ensembles". In: *Physical Review Letters* 99.16.

Rivlin, Theodore J. (1981). *An introduction to the approximation of functions*. Unabridged and corr. republication of the 1969 ed. Dover books on advanced mathematics. Dover.

Roosta-Khorasani, Farbod and Uri Ascher (Sept. 2014). "Improved Bounds on Sample Size for Implicit Matrix Trace Estimators". In: *Foundations of Computational Mathematics* 15.5, pp. 1187–1212.

Schnack, Jürgen, Johannes Richter, and Robin Steinigeweg (Feb. 2020). "Accuracy of the finite-temperature Lanczos method compared to simple typicality-based estimates". In: *Physical Review Research* 2.1.

Trefethen, Lloyd N. (2019). *Approximation Theory and Approximation Practice, Extended Edition*. SIAM.

Ubaru, Shashanka, Jie Chen, and Yousef Saad (2017). "Fast Estimation of $tr(f(A))$ via Stochastic Lanczos Quadrature". In: *SIAM Journal on Matrix Analysis and Applications* 38.4, pp. 1075–1099.

Weiße, Alexander et al. (Mar. 2006). "The kernel polynomial method". In: *Reviews of Modern Physics* 78.1, pp. 275–306.