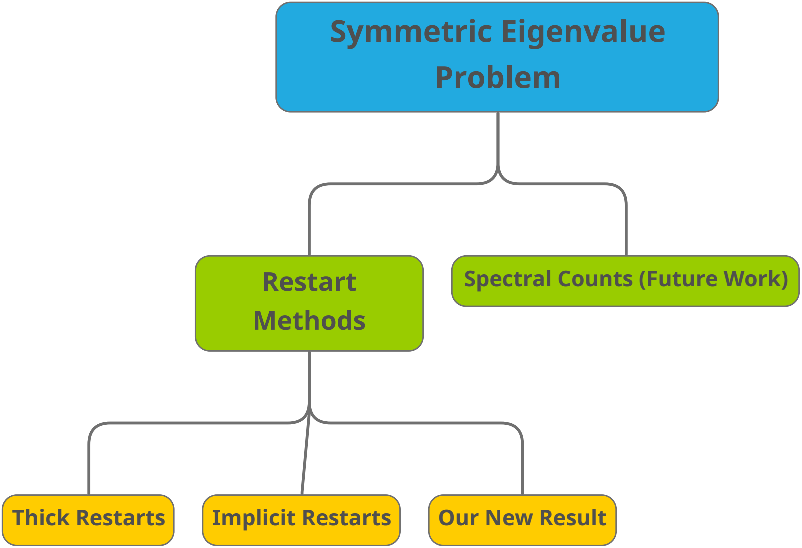
Symmetric Eigenvalue Problems

Kyle Monette

Department of Mathematics and Applied Mathematical Sciences
University of Rhode Island

October 22, 2025


Symmetric Eigenvalue Problem
Restart Methods
Spectral Counts (Future Work)
Thick Restarts
Implicit Restarts
Our New Result

Symmetric Eigenvalue Problem

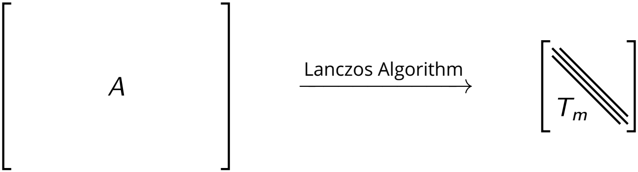
Given: Large, sparse, symmetric $A \in \mathbb{R}^{n\times n}$.
Compute some approximate eigenpairs (λ, x). That is, $Ax = \lambda x$.

If n is small, then A can be "tridiagonalized" to T. An $\mathcal{O}(n^3)$ process.

$$A = \begin{bmatrix} \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \\ \times & \times & \times & \times \end{bmatrix} \longrightarrow T = Q^T A Q = \begin{bmatrix} * & * & & \\ * & * & * & \\ & * & * & * \\ & & * & * \end{bmatrix} \longrightarrow \text{Eigenpairs of } T$$

We assume n is too large for this to be done.

Instead, we'll "reduce" A to a **smaller** tridiagonal matrix T_m.



Lanczos Algorithm

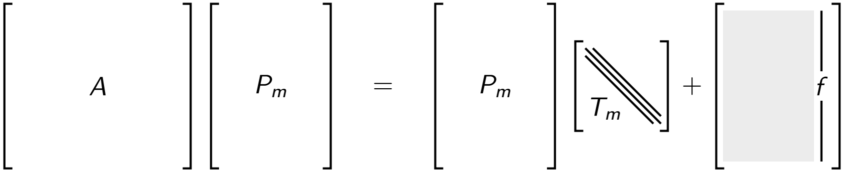
Given symmetric $A \in \mathbb{R}^{n\times n}$, $m \ll n$, and unit vector $p_1 \in \mathbb{R}^n$, generate

$$AP_m = P_m T_m + f e_m^T.$$

$[P, T, f] = \texttt{Lanczos}(A, p_1, m)$

for $j = 1, 2, \ldots, m$ **do**
    $f = Ap_j$
    if $j > 1$ **then** $f = f - p_{j-1}\beta_{j-1}$
    $\alpha_j = f^T p_j$
    $f = f - p_j \alpha_j$ $f = f - P_j(P_j^T f)$
    if $j < m$ **then** $\beta_j = \|f\|$ $p_{j+1} = f/\beta_j$
end for

$$T_m = \begin{bmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \ddots & \ddots & \\ & & \ddots & \alpha_{m-1} & \beta_{m-1} \\ & & & \beta_{m-1} & \alpha_m \end{bmatrix}$$



Ritz Pairs of A

$$\begin{bmatrix} \quad A \quad \end{bmatrix}\begin{bmatrix} P_m \end{bmatrix} = \begin{bmatrix} P_m \end{bmatrix}\begin{bmatrix} T_m \end{bmatrix} + \begin{bmatrix} \quad & f \end{bmatrix}$$

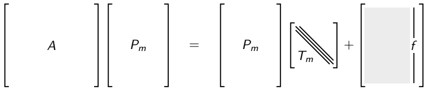
If (θ_j, y_j) is an eigenpair of T_m $\quad (T_m y_j = \theta_j y_j)$ $\quad$ then Ritz pairs of A are

$$(\theta_j, x_j), \quad \text{where} \quad x_j = P_m y_j.$$

The residual is

$$\|Ax_j - \theta_j x_j\| \quad = \quad \|AP_m y_j - \theta_j P_m y_j\| \quad = \quad \left\|f e_m^T y_j\right\| \quad = \quad \|f\| \cdot \left|e_m^T y_j\right|$$

What if residual is large? *Restart* Lanczos, “keeping” desired Ritz vectors.



If (θ_j, y_j) is an eigenpair of T_m $\quad (T_m y_j = \theta_j y_j)$ $\quad$ then Ritz pairs of A are

$$(\theta_j, x_j), \quad \text{where} \quad x_j = P_m y_j.$$

The residual is

$$\|Ax_j - \theta_j x_j\| \quad = \quad \|AP_m y_j - \theta_j P_m y_j\| \quad = \quad \left\|f e_m^T y_j\right\| \quad = \quad \|f\| \cdot \left|e_m^T y_j\right|$$

What if residual is large? *Restart* Lanczos, “keeping” desired Ritz vectors.

Thick Restarts

- MATLAB post-2016
- Krylov-Schur [Ste02], [Wu&Sim00]

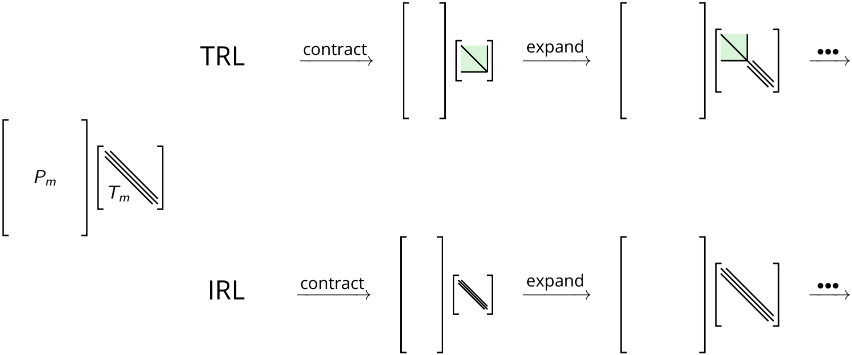
Implicit Restarts

- MATLAB pre-2016
  (Octave, ARPACK)
- IRA / IRL [Sor92] / [Cal&Sor94]

TRL and IRL

Until convergence . . .

1. Generate m-Lanczos factorization: $AP_m = P_m T_m + f e_m^T$.
2. Compute Ritz pairs: (θ_j, x_j) $\quad x_j = P_m y_j$ and $T_m y_j = \theta_j y_j$.
3. Choose $k < m$ "desired" Ritz pairs.
4. "Restart" with these pairs; build a new m-Lanczos factorization.



Thick Restarting Lanczos

$$Ax_j - \theta_j x_j \;=\; f e_m^T y_j \quad \Rightarrow \quad Ax_j \;=\; \underbrace{\beta_m p_{m+1}}_{f} e_m^T y_j + \theta_j x_j \;=\; \underbrace{\overline{\beta}_j}_{\beta_m e_m^T y_j} p_{m+1} + \theta_j x_j$$

Let $\overline{P}_k = \begin{bmatrix} x_1 & \ldots & x_k \end{bmatrix}$ and $\overline{P}_{k+1} = \begin{bmatrix} x_1 & \ldots & x_k & p_{m+1} \end{bmatrix}$.

One can show that

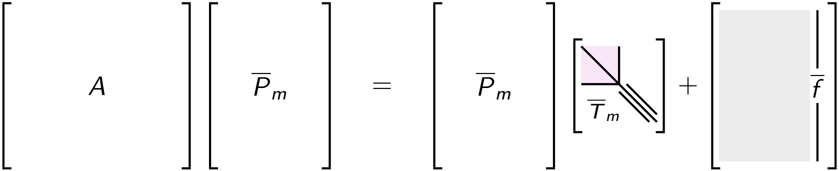
$$\overline{P}_{k+1}^T A \overline{P}_{k+1} \;:=\; \overline{T}_{k+1} \;=\; \begin{bmatrix} \theta_1 & & & & \overline{\beta}_1 \\ & \theta_2 & & & \overline{\beta}_2 \\ & & \ddots & & \vdots \\ & & & \theta_k & \overline{\beta}_k \\ \overline{\beta}_1 & \overline{\beta}_2 & \ldots & \overline{\beta}_k & \alpha_{k+1} \end{bmatrix}$$

where $\alpha_{k+1} = p_{m+1}^T A p_{m+1}$.

Now the “restart” makes sense:

Start Lanczos with vector p_{m+1}, and do $m-(k+1)$ steps.

The first $k+1$ chunk is $A\overline{P}_{k+1} = \overline{P}_{k+1}\,\overline{T}_{k+1} + \beta_{k+1} p_{k+2} e_{k+1}^T$.

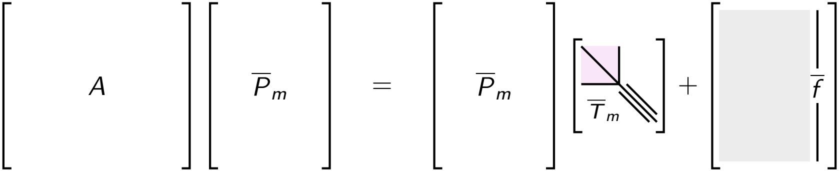


Thick Restarting Lanczos

Now the "restart" makes sense:

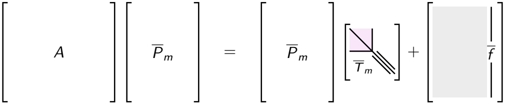
Start Lanczos with vector p_{m+1}, and do $m-(k+1)$ steps.

The first $k+1$ chunk is $A\overline{P}_{k+1} = \overline{P}_{k+1}\,\overline{T}_{k+1} + \beta_{k+1} p_{k+2} e_{k+1}^T$.



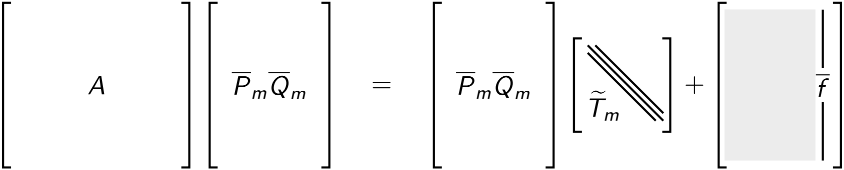
Question: Can the "arrowhead" $\overline{T}_{k+1}$ be "tridiagonalized" efficiently?

Is there a benefit?



We showed [Bag,Mon,Per25+] that we can tridiagonalize $\overline{T}_{k+1}$ via

$$\widetilde{T}_{k+1} = \overline{Q}_{k+1}^T \, \overline{T}_{k+1} \, \overline{Q}_{k+1} \qquad \overline{Q}_{k+1} = \begin{bmatrix} & \overline{Q}_k & & \begin{matrix} 0 \\ \vdots \\ 0 \end{matrix} \\ 0 & \dots & 0 & 1 \end{bmatrix}$$



Equating the first k columns: $A\overline{P}_k\overline{Q}_k = \overline{P}_k\overline{Q}_k\widetilde{T}_k + \widetilde{t}_{k+1,k}\, p_{m+1} e_k^T$

Another Restart Method — IRL

TRL Summary: Restart with k Ritz vectors.

Another Variation: Can we choose a new (better) initial vector $\widehat{p}_1$?

Goal: $$\widehat{p}_1 \approx \underbrace{c_1 x_1 + \cdots + c_k x_k}_{\text{large } c_i} + \underbrace{c_{k+1} x_{k+1} + \cdots + c_n x_n}_{\text{small } c_i}$$

IRL applies a polynomial filter to update the starting vector.

$$\widehat{p}_1 = q(A) p_1 = c_1 q(\lambda_1) x_1 + \cdots + c_k q(\lambda_k) x_k + c_{k+1} q(\lambda_{k+1}) x_{k+1} + \cdots + c_n q(\lambda_n) x_n$$

Which polynomial q?

Applying Shifts

Say (μ, x) is an approximate eigenpair of A. Then

$$x^T(A - \mu I)p_1 \quad \approx \quad 0.$$

So $(A - \mu I)p_1$ has (approx.) no components in direction of x.

Choose μ to be a "shift" from the undesired part of the spectrum.

Result: A vector $(A - \mu I)p_1$ *enriched* in direction of desired eigenvectors.

However, notice that

$$(A - \mu I)P_m \quad = \quad P_m(T_m - \mu I) + fe_m^T.$$

So, we shift T_m instead of A.

Applying Shifts

Given the undesired eigenvalues $\theta_{k+1}, \ldots, \theta_m$ of T_m, we consider

$$q(T_m) \quad = \quad (T_m - \theta_{k+1} I) \ldots (T_m - \theta_m I)$$

(*which is not formed explicitly!*).

Use the implicit *QR* algorithm with shifts $\theta_{k+1}, \ldots, \theta_m$ to obtain

$$T_m^+ \quad := \quad {Q_m^+}^T T_m Q_m^+, \qquad q(T_m) = Q_m^+ R_m^+.$$

Implicit QR Algorithm

Choose a shift μ.

Create a rotation matrix $\widetilde{G}_1 = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}$ such that $\widetilde{G}_1^T \begin{bmatrix} t_{11} - \mu \\ t_{21} \end{bmatrix} = \begin{bmatrix} \star \\ 0 \end{bmatrix}$.

Implicit QR Algorithm

Choose a shift μ.

Create a rotation matrix $\widetilde{G}_1 = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}$ such that $\widetilde{G}_1^T \begin{bmatrix} t_{11} - \mu \\ t_{21} \end{bmatrix} = \begin{bmatrix} \star \\ 0 \end{bmatrix}$.

$$T_m = \begin{bmatrix} * & * & & & \\ * & * & * & & \\ & * & * & * & \\ & & * & * & * \\ & & & * & * \end{bmatrix} \quad \to \quad G_1^T T_m G_1 = \begin{bmatrix} \times & \times & \bullet & & \\ \times & \times & \times & & \\ \bullet & \times & * & * & \\ & & * & * & * \\ & & & * & * \end{bmatrix}.$$

Implicit QR Algorithm

Choose a shift μ.

Create a rotation matrix $\widetilde{G}_1 = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}$ such that $\widetilde{G}_1^T \begin{bmatrix} t_{11} - \mu \\ t_{21} \end{bmatrix} = \begin{bmatrix} \star \\ 0 \end{bmatrix}$.

$$T_m = \begin{bmatrix} * & * & & & \\ * & * & * & & \\ & * & * & * & \\ & & * & * & * \\ & & & * & * \end{bmatrix} \quad \to \quad G_1^T T_m G_1 = \begin{bmatrix} \times & \times & \bullet & & \\ \times & \times & \times & & \\ \bullet & \times & * & * & \\ & & * & * & * \\ & & & * & * \end{bmatrix}.$$

$$\underbrace{\begin{bmatrix} * & \times & & & \\ \times & \times & \times & \bullet & \\ & \times & \times & \times & \\ & \bullet & * & * & * \\ & & & * & * \end{bmatrix}}_{\text{Action of } G_2}$$

Implicit QR Algorithm

Choose a shift μ.

Create a rotation matrix $\widetilde{G}_1 = \begin{bmatrix} c & -s \\ s & c \end{bmatrix}$ such that $\widetilde{G}_1^T \begin{bmatrix} t_{11} - \mu \\ t_{21} \end{bmatrix} = \begin{bmatrix} \star \\ 0 \end{bmatrix}$.

$$T_m = \begin{bmatrix} * & * & & & \\ * & * & * & & \\ & * & * & * & \\ & & * & * & * \\ & & & * & * \end{bmatrix} \quad \to \quad G_1^T T_m G_1 = \begin{bmatrix} \times & \times & \bullet & & \\ \times & \times & \times & & \\ \bullet & \times & * & * & \\ & & * & * & * \\ & & & * & * \end{bmatrix}.$$

$$\underbrace{\begin{bmatrix} * & \times & & & \\ \times & \times & \times & \bullet & \\ & \times & \times & \times & \\ & \bullet & * & * & * \\ & & & * & * \end{bmatrix}}_{\text{Action of } G_2} \quad \underbrace{\begin{bmatrix} * & * & & & \\ * & * & \times & & \\ & \times & \times & \times & \bullet \\ & & \times & \times & \times \\ & & \bullet & \times & * \end{bmatrix}}_{\text{Action of } G_3} \quad \underbrace{\begin{bmatrix} * & * & & & \\ * & * & * & & \\ & * & * & \times & \\ & & \times & \times & \times \\ & & & \times & \times \end{bmatrix}}_{\text{Action of } G_4}$$

Implicit QR Algorithm

So after one shift μ, we obtain

$$T_m^+ \quad := \quad {Q_m^+}^T T_m Q_m^+, \qquad Q_m^+ = G_1 \dots G_{m-1}.$$

If μ is an (undesired) eigenvalue of T_m (an "exact shift"):

Then $\quad t_{m,m}^+ = \mu \quad$ and $\quad t_{m,m-1}^+ = t_{m-1,m}^+ = 0.$

Implicit QR Algorithm

So after one shift μ, we obtain

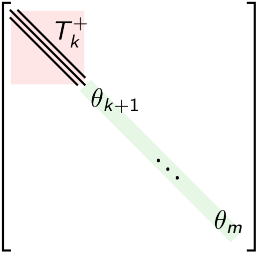
$$T_m^+ := {Q_m^+}^T T_m Q_m^+, \qquad Q_m^+ = G_1 \dots G_{m-1}.$$

If μ is an (undesired) eigenvalue of T_m (an "exact shift"):

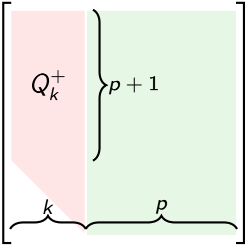
Then $t_{m,m}^+ = \mu$ and $t_{m,m-1}^+ = t_{m-1,m}^+ = 0.$

Repeat for $p := m - k$ exact shifts:

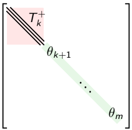
$T_m^+ =$

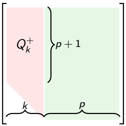


$Q_m^+ =$



$$
\begin{aligned}
AP_mQ_m^+ &= P_mQ_m^+T_m^+ + fe_m^TQ_m^+ \\
A\underbrace{P_k^+}_{P_mQ_k^+} &= P_k^+T_k^+ + \underbrace{q_{m,k}^+ f}_{f^+} e_k^T
\end{aligned}
$$

$T_m^+ =$ 

$Q_m^+ =$ 

Now, restart Lanczos with f^+ and build to size m.

Revisiting the initial vector:

From $AP_m = P_mT_m + fe_m^T$ one can show $q(A)P_me_1 = P_m\underbrace{q(T_m)}_{Q_m^+R_m^+}e_1$.

Then $q(A)p_1 = P_mQ_m^+r_{11}^+e_1 \quad \Rightarrow \quad q(A)p_1 = r_{11}^+\widehat{p}_1$.

Therefore, $\widehat{p}_1 \propto q(A)p_1$ is a “polynomial filter” update to p_1.

Which has stronger components in desired Ritz vectors.

Equivalence

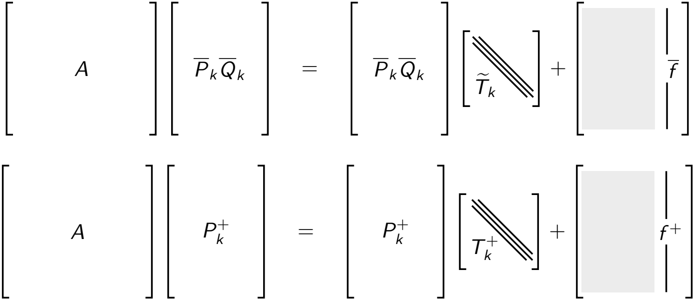
TRL and IRL (exact shifts) are “mathematically equivalent”.

Equivalence

TRL and IRL (exact shifts) are “mathematically equivalent”.

By considering the k-Lanczos factorizations …

$$
\begin{array}{lrcl}
\text{TRL} & A\overline{P}_k\overline{Q}_k & = & \overline{P}_k\overline{Q}_k\widetilde{T}_k + \widetilde{t}_{k+1,k}\,p_{m+1}e_k^T \\
\text{IRL} & AP_k^+ & = & P_k^+T_k^+ + q_{m,k}^+\,p_{m+1}e_k^T
\end{array}
$$

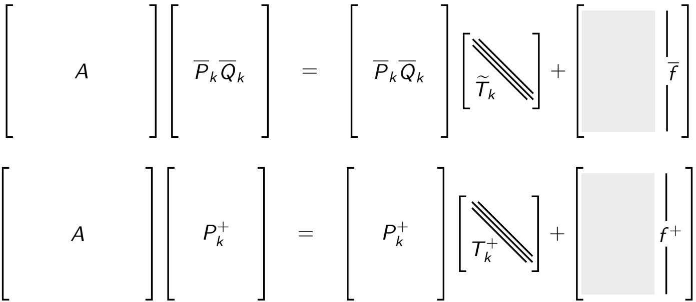


TRL and IRL (exact shifts) are "mathematically equivalent".

By considering the k-Lanczos factorizations . . .

$$\text{TRL} \qquad A\overline{P}_k\overline{Q}_k = \overline{P}_k\overline{Q}_k\widetilde{T}_k + \widetilde{t}_{k+1,k}p_{m+1}e_k^T$$

$$\text{IRL} \qquad AP_k^+ = P_k^+T_k^+ + q_{m,k}^+p_{m+1}e_k^T$$



Our New Result: $\overline{P}_k\overline{Q}_k = \pm P_k^+, \qquad \widetilde{T}_k = \pm T_k^+, \qquad \overline{f} = \pm f^+.$

In Summary . . .

Our new method [Bag,Mon,Per25+] (arrowhead to tridiagonal) . . .

- **Maintains** tridiagonal structure.
- **Not** sensitive to numerical round-off.
- **Equivalent** to a new method [Mas&Dor18].
- Provides new, **stronger** insight: TRL $=$ IRL (not just equivalent!)
- Adapts to the bidiagonal case & singular value problem.
  A seminar presentation (and another paper!) to come . . .

New Topic — Spectral Counts

Let $A \in \mathbb{R}^{n \times n}$ be a (large!) symmetric matrix and $[a, b] \subset \mathbb{R}$.

Question: How many eigenvalues of A are in $[a, b]$? c.f. [Nap16]

Our (Future) Interest: Accelerating algorithms with this count.

Potential Insight: Form the projection matrix P

$$P \quad := \quad \sum_{\substack{i \\ \lambda_i \in [a,b]}} u_i u_i^T$$

which satisfies $P^2 = P$, so eigenvalues of P are 0 and 1.

Key Result: $\mathrm{trace}(P)$ is the number of eigenvalues of A in $[a, b]$.

Approximate trace(P)

Let $h(t)$ be the indicator function

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

and consider its representation in the Chebyshev basis

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

Linearly map eigenvalues of A and $[a, b]$ into $[-1, 1]$. Then

$$P \quad \approx \quad h(A) \quad \approx \quad \sum_{j=0}^{p} \gamma_j T_j(A)$$

has eigenvalues of 1 if $\lambda_i \in [a, b]$ and eigenvalues of 0 otherwise.

Therefore: $\quad \text{trace}(P) \; = \; \#\,\text{eigenvalues of } A \text{ in } [a, b] \; \approx \; \text{trace}(h(A)).$

Hutchinson's Theorem
Suppose each component of $v \in \mathbb{R}^n$ is drawn independently from a distribution with mean 0 and variance 1; that is, $\mathbb{E}[v] = 0$ and $\mathbb{E}[vv^T] = I_n$.

Then for any symmetric matrix A and matrix function f defined on A,

$$\mathbb{E}\left[v^T f(A) v\right] \quad = \quad \text{trace}\, f(A).$$

In particular, we'll use $v \sim \mathrm{unif}\{-1, 1\}^n$, called Rademacher vectors (for the least variance estimator).

Therefore, $$\text{trace}(P) \approx \text{trace}(h(A)) = \mathbb{E}\left[v^T h(A) v\right].$$

Computing $\mathrm{trace}(P) \approx \mathbb{E}\left[v^T h(A) v\right]$

From before ...

$$
\begin{aligned}
\#\text{ eigenvalues of } A \text{ in } [a, b] \quad &= \quad \mathrm{trace}(P) \\
&\approx \quad \mathrm{trace}(h(A)) \\
&= \quad \mathbb{E}\left[v^T h(A) v\right] \\
&= \quad \mathbb{E}\left[\sum_{j=0}^{p} \gamma_j v^T T_j(A) v\right] \\
&= \quad n \cdot \mathbb{E}\left[\sum_{j=0}^{p} \gamma_j v^T T_j(A) v\right] \qquad \text{(normalize } v\text{)} \\
&\approx \quad \frac{n}{n_v} \sum_{k=1}^{n_v} \sum_{j=0}^{p} \gamma_j v_k^T T_j(A) v_k
\end{aligned}
$$

Future Work

$T_p(A)v$ is a (modest size) polynomial of A times $v \dots$

Then $w = T_p(A)v$ has components in the eigenvectors of A from $[a, b]$.

Future work: Optimizing existing algorithms by using w.

- Solving $A(x + w) = b + Aw$ instead of $Ax = b$.
- Symmetric eigenvalue problems.
- Singular value thresholding (since $\sigma(A) = \sqrt{\lambda(A^T A)}$).
- $\dots$

References

- [Bag,Mon,Per25+] Baglama, J., Monette, K., & Perovic. V. (to appear) Application of Transformation Matrices of Arrowhead Matrices in Connection with Symmetric Tridiagonal Matrices.
- [Cal&Sor94] Calvetti, D., Reichel, L., & Sorensen, D. C. (1994). An implicitly restarted Lanczos method for large symmetric eigenvalue problems. Electronic Transactions on Numerical Analysis, 2(1), 21.
- [Nap16] Di Napoli, E., Polizzi, E., & Saad, Y. (2016). Efficient estimation of eigenvalue counts in an interval. Numerical Linear Algebra with Applications, 23(4), 674-692.
- [Mas&Dor18] Mastronardi, N., & Dooren, P. V. (2018). The QR steps with perfect shifts. SIAM Journal on Matrix Analysis and Applications, 39(4), 1591-1615.
- [Sor92] Sorensen, D. C. (1992). Implicit application of polynomial filters in a k-step Arnoldi method. Siam journal on matrix analysis and applications, 13(1), 357-385.
- [Ste02] Stewart, G. W. (2002). A Krylov–Schur algorithm for large eigenproblems. SIAM Journal on Matrix Analysis and Applications, 23(3), 601-614.
- [Wu&Sim00] Wu, K., & Simon, H. (2000). Thick-restart Lanczos method for large symmetric eigenvalue problems. SIAM Journal on Matrix Analysis and Applications, 22(2), 602-616.

Thank you!

Questions?

Lanczos Recurrence

Three term recurrence: Arnoldi gives $T_{ij} = p_i^T A p_j$, for $1 \leq i \leq j$.

If A is symmetric, then $T_{ij} = (Ap_i)^T p_j$.

For $i < j - 1$, then $Ap_i \in \text{span}\{p_1, \ldots, p_{j-1}\}$ and p_j is perpendicular to this span, so it must be that $(Ap_i)^T p_j = 0$.

Therefore, T has zero 2nd off-diagonals.

Ritz Extraction

Let $\mathcal{K}_m$ be an m-dimensional subspace of $\mathbb{R}^n$.

We say (θ, x), where $x \in \mathcal{K}_m$, is a *Ritz pair* of A if $Ax - \theta x \perp \mathcal{K}_m$.

Let $P_m \in \mathbb{R}^{n \times m}$ be an ONB for $\mathcal{K}_m$.

Then $\exists\, y \in \mathbb{R}^m$ such that $x = P_m y$.

Let $T_m = P_m^T A P_m \in \mathbb{R}^{m \times m}$.

Then (θ, x) is a Ritz pair iff (θ, y) is an eigenpair of T_m.

TRL Step

Let $\Theta_k = \text{diag}(\theta_1, \ldots, \theta_k)$ and $Y_k = Y_{:,1:k}$. Then $A\overline{P}_k = \overline{P}_k \Theta_k + \beta_m p_{m+1} e_m^T Y_k$, or equivalently $A\overline{P}_k = \overline{P}_{k+1} \overline{T}_{k+1,k}$. Orthogonalize Ap_{m+1} against $\overline{P}_{k+1}$:

$$
\begin{aligned}
r &= (I - \overline{P}_{k+1}\overline{P}_{k+1}^T)Ap_{m+1} = Ap_{m+1} - \begin{bmatrix} \overline{P}_k & p_{m+1} \end{bmatrix} \begin{bmatrix} \overline{P}_k^T \\ p_{m+1}^T \end{bmatrix} Ap_{m+1} \\
&= Ap_{m+1} - \overline{P}_k \overline{P}_k^T Ap_{m+1} - p_{m+1}p_{m+1}^T Ap_{m+1} \\
&= Ap_{m+1} - \overline{P}_k (A\overline{P}_k)^T p_{m+1} - p_{m+1}p_{m+1}^T Ap_{m+1} \\
&= Ap_{m+1} - \overline{P}_k(\Theta_k \overline{P}_k^T p_{m+1}) - \overline{P}_k \beta_m Y_k^T e_m p_{m+1}^T p_{m+1} - p_{m+1}p_{m+1}^T Ap_{m+1} \\
&= Ap_{m+1} - p_{m+1}p_{m+1}^T Ap_{m+1} - \overline{P}_k \beta_m Y_k^T e_m.
\end{aligned}
$$

Let $p_{k+2} = r/\|r\| = r/\beta_{k+1}$, and $\alpha_{k+1} = p_{m+1}^T Ap_{m+1}$. Then

$$
Ap_{m+1} = \beta_{k+1} p_{k+2} + \alpha_{k+1} p_{m+1} + \overline{P}_k \overline{\beta}
$$

where $\overline{\beta}$ is a vector of $(\overline{\beta}_1, \ldots, \overline{\beta}_k)$ and $\overline{\beta}_i = \beta_m y_i^T e_m$.

TRL Step

$$
\begin{aligned}
A\overline{P}_{k+1} &= \left[\overline{P}_{k+1}\overline{T}_{k+1,k} \mid Ap_{m+1}\right] \\
&= \left[\ \overline{P}_{k+1}\overline{T}_{k+1,k} \ \middle|\ \beta_{k+1}p_{k+2} + \alpha_{k+1}p_{m+1} + \overline{P}_k\overline{\beta}\ \right] \\
&= \left[\ \overline{P}_{k+1}\overline{T}_{k+1,k} \ \middle|\ \overline{P}_{k+1}\begin{bmatrix} \overline{\beta} \\ \alpha_{k+1} \end{bmatrix}\ \right] + \beta_{k+1}p_{k+2}e_{k+1}^T \\
&= \overline{P}_{k+1}\overline{T}_{k+1} + \beta_{k+1}p_{k+2}e_{k+1}^T
\end{aligned}
$$

which also shows that $\overline{P}_{k+1}^T A\overline{P}_{k+1} = \overline{T}_{k+1}$ by orthogonality.

TRL Truncation

From $A\overline{P}_{k+1} = \overline{P}_{k+1}\overline{T}_{k+1} + \beta_{k+1}p_{k+2}e_{k+1}^T$, then

$$A\overline{P}_{k+1}\overline{Q}_{k+1} = \overline{P}_{k+1}\overline{Q}_{k+1}\widetilde{T}_{k+1} + \beta_{k+1}p_{k+2}e_{k+1}^T\overline{Q}_{k+1}.$$

Note that $e_{k+1}^T\overline{Q}_{k+1} = e_{k+1}^T$ and the first k columns of $p_{k+2}e_{k+1}^T$ are zeros. Therefore,

$$A\begin{bmatrix} \overline{P}_k & p_{m+1} \end{bmatrix}\begin{bmatrix} \overline{Q}_k & 0 \\ 0 & 1 \end{bmatrix} = \underbrace{\begin{bmatrix} \overline{P}_k & p_{m+1} \end{bmatrix}\begin{bmatrix} \overline{Q}_k & 0 \\ 0 & 1 \end{bmatrix}}_{=\begin{bmatrix} \overline{P}_k\overline{Q}_k & p_{m+1} \\ 0 & 0 \end{bmatrix}}\begin{bmatrix} \widetilde{T}_k & \widetilde{t}_{k+1,k}e_k \\ \widetilde{t}_{k+1,k}e_k^T & \widetilde{t}_{k,k} \end{bmatrix} + \beta_{k+1}p_{k+2}e_{k+1}^T$$

which gives us

$$A\overline{P}_k\overline{Q}_k = \overline{P}_k\overline{Q}_k\widetilde{T}_k + \widetilde{t}_{k+1,k}p_{m+1}e_k^T.$$

IRL Truncation

From $AP_m = P_m T_m + f e_m^T$, then $AP_m Q_m^+ = P_m Q_m^+ T_m^+ + f e_m^T Q_m^+$.

$$AP_m \left[\begin{array}{c|c} Q_k^+ & Q^+ \end{array}\right] = P_m \left[\begin{array}{c|c} Q_k^+ & Q^+ \end{array}\right] \left[\begin{array}{cc} T_k^+ & t_{k,k+1}^+ e_k e_1^T \\ t_{k+1,k}^+ e_1 e_k^T & \Theta \end{array}\right] + f e_m^T \left[\begin{array}{c|c} Q_k^+ & Q^+ \end{array}\right]$$

Then $AP_m Q_k^+ = P_m Q_k^+ T_k^+ + P_m Q^+ t_{k+1,k}^+ e_1 e_k^T + f \underbrace{e_m^T Q_k^+}_{= q_{m,k}^+ e_k^T}$.

Mathematically, $t_{k+1,k}^+ = 0$. But we need to include this term, so

$$AP_m Q_k^+ = P_m Q_k^+ T_k^+ + \left(P_m Q_m^+ e_{k+1} t_{k+1,k}^+ + q_{m,k}^+ f\right) e_k^T$$

since $Q^+ e_1 \coloneqq Q_m^+ e_{k+1}$.

Example

200×200 symmetric matrix $\mu^* = 15$ (exact)

(Left) Varying p, $n_v = 80$ fixed

(Right) Varying n_v, $p = 20$ fixed