Orthogonal Similarity Reductions in Eigenvalue Algorithms

AMS Graduate Conference
Brown University

Kyle Monette

April 11, 2026

THE
UNIVERSITY
OF RHODE ISLAND

Introduction

Definition: An *eigenpair* (λ, x) of an $n \times n$ matrix A is a scalar $\lambda \in \mathbb{R}$ (*eigenvalue*) and nonzero vector $x \in \mathbb{R}^n$ (*eigenvector*) such that

$$Ax = \lambda x.$$

Our focus is on the *symmetric* eigenvalue problem—where $A = A^T$.

Introduction

Definition: An *eigenpair* (λ, x) of an $n \times n$ matrix A is a scalar $\lambda \in \mathbb{R}$ (*eigenvalue*) and nonzero vector $x \in \mathbb{R}^n$ (*eigenvector*) such that

$$Ax = \lambda x.$$

Our focus is on the *symmetric* eigenvalue problem—where $A = A^T$.

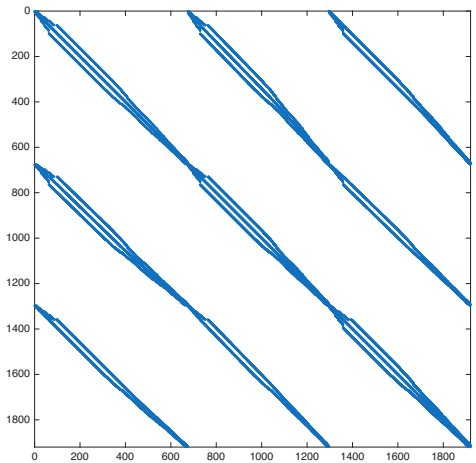
Small and Dense A

Iterative algorithms may be unnecessary

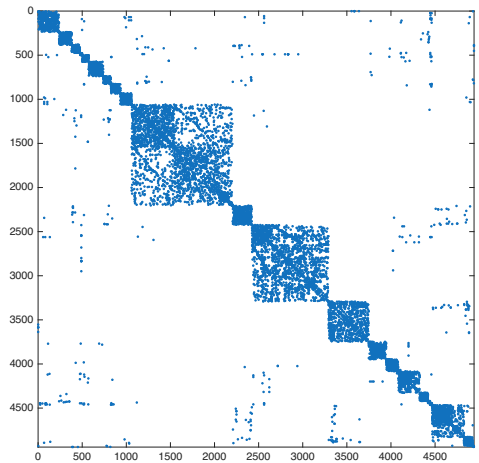
Large and Sparse A

Goal: Compute a few of the largest eigenpairs

Finite difference model ($n = 1919$)



Adjacency matrix ($n = 4941$)



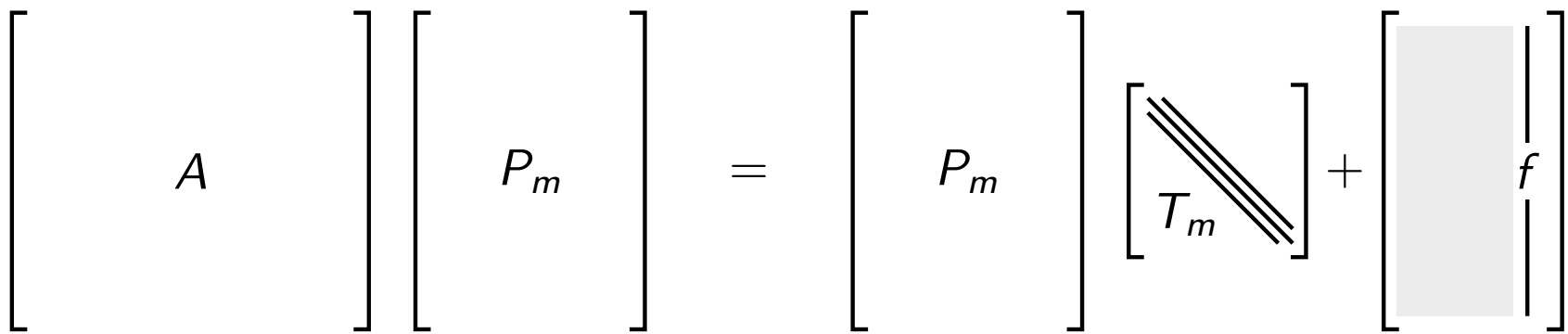
Lanczos Factorization

Goal: Approximate largest eigenpairs $\{(\lambda_i, x_i)\}_{i=1}^{k}$ of large, symmetric A.

Key Idea: Project A onto a much smaller m-dimensional subspace.

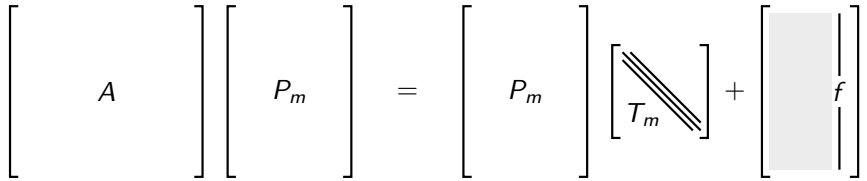
The *Lanczos Algorithm* constructs this subspace so that the projection is tridiagonal.

$$AP_m = P_m T_m + f e_m^T \quad \Leftrightarrow \quad T_m = P_m^T A P_m.$$



** This is the cornerstone of MATLAB's 'eigs' command!

Eigenpairs of T_m

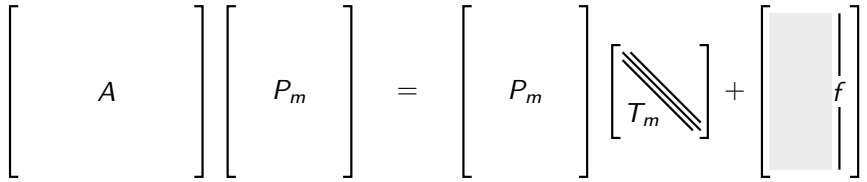


Eigenpairs (θ, y) of T_m are used to approximate those of A:

$$T_m y = \theta y \quad \Rightarrow \quad A(P_m y) = \theta(P_m y) + f e_m^T y.$$

We call $(\theta, P_m y)$ a Ritz pair of A.

Eigenpairs of T_m

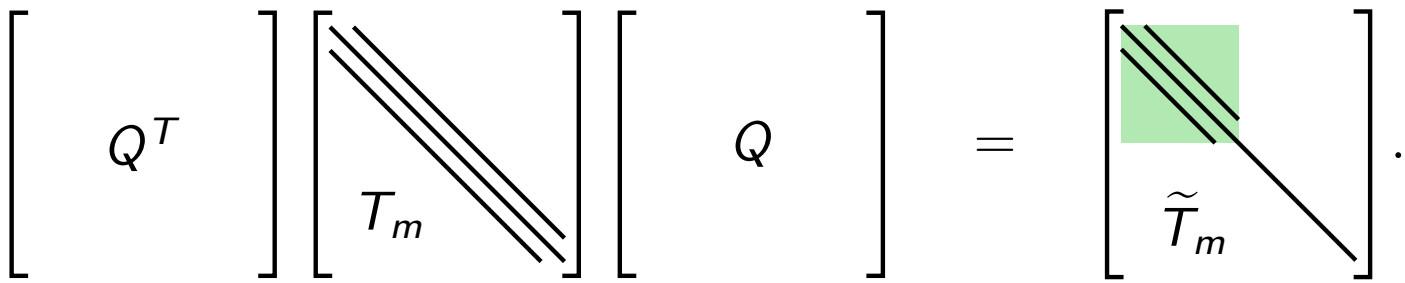


Eigenpairs (θ, y) of T_m are used to approximate those of A:

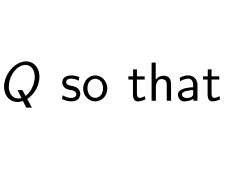
$$T_m y = \theta y \quad \Rightarrow \quad A(P_m y) = \theta(P_m y) + f e_m^T y.$$

We call $(\theta, P_m y)$ a Ritz pair of A.

Deflate out the $m - k$ "bad" eigenvalues of T_m via an orthogonal Q:



Creating the Arrowhead

How do we create Q so that $\begin{bmatrix} \\ Q^T \\ \\ \end{bmatrix}$  $\begin{bmatrix} \\ Q \\ \\ \end{bmatrix} = \begin{bmatrix} \\ \widetilde{T}_m \\ \\ \end{bmatrix}$?

Let $\overline{P}_{k+1} = \big[\underbrace{P_m Y_k}_{\textit{Ritz vectors}} \quad f/\beta\big]$ where $\beta = \|f\|$. One can show that

$$\overline{P}_{k+1}^T A \overline{P}_{k+1} \quad = \quad D_{k+1} \quad = \quad \begin{bmatrix} \theta_1 & & & & \beta y_{m,1} \\ & \theta_2 & & & \beta y_{m,2} \\ & & \ddots & & \vdots \\ & & & \theta_k & \beta y_{m,k} \\ \beta y_{m,1} & \beta y_{m,2} & \cdots & \beta y_{m,k} & * \end{bmatrix}.$$

So the projection of A on $\overline{P}_{k+1}$ basis is not tridiagonal ...

Arrowhead $\rightarrow$ Tridiagonal

Remedy: "Tridiagonalize" the arrowhead D_{k+1} so that

$$\begin{bmatrix} \overline{Q}_k^T & \\ & 1 \end{bmatrix} \begin{bmatrix} D_{k+1} \end{bmatrix} \begin{bmatrix} \overline{Q}_k & \\ & 1 \end{bmatrix} = \begin{bmatrix} \overline{T}_{k+1} \end{bmatrix}.$$

Arrowhead $\rightarrow$ Tridiagonal

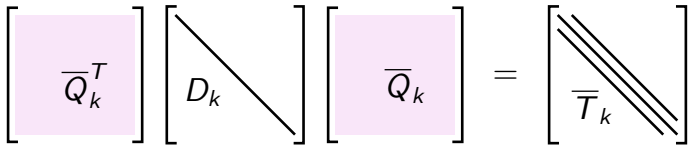
Remedy: "Tridiagonalize" the arrowhead D_{k+1} so that

$$\begin{bmatrix} \overline{Q}_k^T & \\ & 1 \end{bmatrix} \begin{bmatrix} D_{k+1} \end{bmatrix} \begin{bmatrix} \overline{Q}_k & \\ & 1 \end{bmatrix} = \begin{bmatrix} \overline{T}_{k+1} \end{bmatrix}.$$

With the special 1 in $\overline{Q}_k$, we can truncate and obtain

$$\overline{Q}_k^T D_k \overline{Q}_k = \overline{T}_k \quad \Rightarrow \quad \begin{bmatrix} \overline{Q}_k^T \end{bmatrix} \begin{bmatrix} D_k \end{bmatrix} \begin{bmatrix} \overline{Q}_k \end{bmatrix} = \begin{bmatrix} \overline{T}_k \end{bmatrix}.$$

All Together . . .



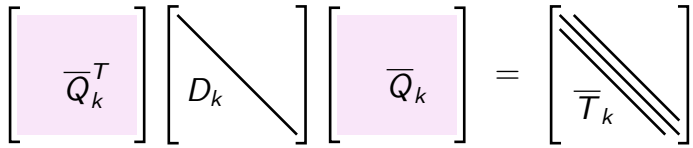
From $T_m Y_m = Y_m \Theta_m$ we get

$$Y_m^T T_m Y_m = \begin{bmatrix} D_k = \Theta_k & \\ & \Theta_{m-k} \end{bmatrix} = \begin{bmatrix} \overline{Q}_k \overline{T}_k \overline{Q}_k^T & \\ & \Theta_{m-k} \end{bmatrix}.$$

After some rearranging:

$$\begin{bmatrix} Y_k \overline{Q}_k & \\ & Y_{m-k} \end{bmatrix}^T T_m \begin{bmatrix} Y_k \overline{Q}_k & \\ & Y_{m-k} \end{bmatrix} = \begin{bmatrix} \overline{T}_k & \\ & \Theta_{m-k} \end{bmatrix}$$

All Together . . .



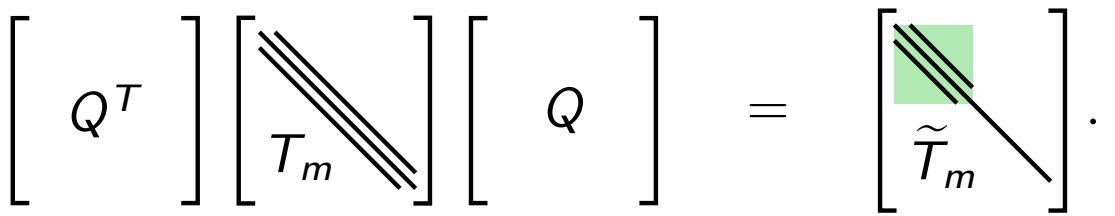
From $T_m Y_m = Y_m \Theta_m$ we get

$$Y_m^T T_m Y_m = \begin{bmatrix} D_k = \Theta_k & \\ & \Theta_{m-k} \end{bmatrix} = \begin{bmatrix} \overline{Q}_k \overline{T}_k \overline{Q}_k^T & \\ & \Theta_{m-k} \end{bmatrix}.$$

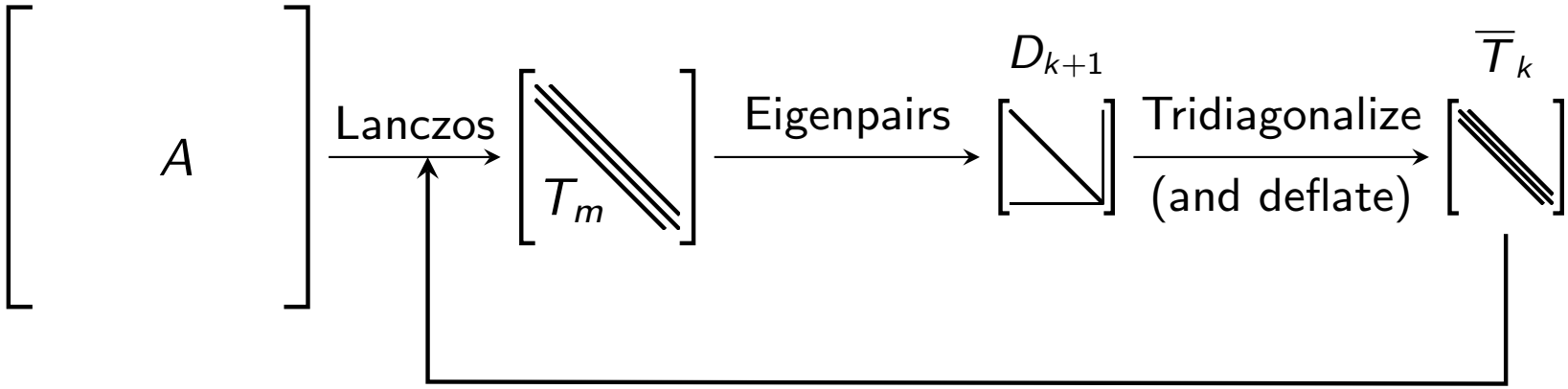
After some rearranging:

$$\begin{bmatrix} Y_k \overline{Q}_k & \\ & Y_{m-k} \end{bmatrix}^T T_m \begin{bmatrix} Y_k \overline{Q}_k & \\ & Y_{m-k} \end{bmatrix} = \begin{bmatrix} \overline{T}_k & \\ & \Theta_{m-k} \end{bmatrix}$$

This accomplishes our goal: find orthogonal $Q \in \mathbb{R}^{m \times m}$ so that



In Summary


A
Lanczos
T_m
Eigenpairs
D_{k+1}
Tridiagonalize
(and deflate)
$\overline{T}_k$

In Summary

- The arrowhead $\rightarrow$ tridiagonal conversion gives a new insight into the mathematically equivalent methods for restarting Lanczos:

$$\underbrace{\text{Keeping Ritz vectors } Y_k}_{\text{“Thick Restarts”}} \quad = \quad \underbrace{\text{Deflating } m-k \text{ “bad” Ritz values}}_{\text{“Implicit Restarts”}}$$

In Summary

- The arrowhead $\rightarrow$ tridiagonal conversion gives a new insight into the mathematically equivalent methods for restarting Lanczos:

$$\underbrace{\text{Keeping Ritz vectors } Y_k}_{\text{“Thick Restarts”}} \quad = \quad \underbrace{\text{Deflating } m-k \text{ “bad” Ritz values}}_{\text{“Implicit Restarts”}}$$

- Moreover, numerical tests show greater accuracy at deflating multiple eigenvalues compared to, e.g., implicit QR algorithm.

$$\begin{bmatrix} \\ Q^T \\ \\ \end{bmatrix} \begin{bmatrix} \\ T_m \\ \\ \end{bmatrix} \begin{bmatrix} \\ Q \\ \\ \end{bmatrix} = \begin{bmatrix} \\ \widetilde{T}_m \\ \\ \end{bmatrix}$$

  Example: Tridiagonal T_{31} with main diagonal $15, \ldots, 1, 0, 1, \ldots, 15$ and sub/super-diagonal 1's.
  For $k = 15$: deflation using QR algorithm is 10^{-4}
  deflation using arrowhead method is 10^{-14}!

In Summary

- The arrowhead $\rightarrow$ tridiagonal conversion gives a new insight into the mathematically equivalent methods for restarting Lanczos:

$$\underbrace{\text{Keeping Ritz vectors } Y_k}_{\text{“Thick Restarts”}} \quad = \quad \underbrace{\text{Deflating } m-k \text{ “bad” Ritz values}}_{\text{“Implicit Restarts”}}$$

- Moreover, numerical tests show greater accuracy at deflating multiple eigenvalues compared to, e.g., implicit QR algorithm.

$$\begin{bmatrix} \\ Q^T \\ \\ \end{bmatrix} \begin{bmatrix} \\ \diagdown \\ T_m \end{bmatrix} \begin{bmatrix} \\ Q \\ \\ \end{bmatrix} = \begin{bmatrix} \blacksquare \\ \diagdown \\ \widetilde{T}_m \end{bmatrix}$$

  Example: Tridiagonal T_{31} with main diagonal $15, \ldots, 1, 0, 1, \ldots, 15$ and sub/super-diagonal 1's.
  For $k = 15$: deflation using QR algorithm is 10^{-4}
  deflation using arrowhead method is 10^{-14}!

- We extended this to bidiagonalization and the SVD for restarting the GKLB algorithm.

Thank you!