Accelerating Arnoldi Eigenvalue Computations with Polynomial Preconditioning

Mark Embree · Virginia Tech

Jennifer Loe · Sandia National Labs

Ron Morgan · Baylor University

SIAM Applied Linear Algebra Conference · May 2021

Thanks to: US National Science Foundation DMS-1720257.

"Polynomial Preconditioned Arnoldi with Stability Control"
SIAM J. Sci. Comp. 43 A1–A25, 2021.

Setting

- Suppose A is large, nonsymmetric, $n \times n$ matrix,
  and we seek the smallest magnitude eigenvalues of A.

- Shift-invert. Apply (restarted) Arnoldi to A^{-1}:

  eigenvalue $\lambda \in \sigma(A)$ becomes $1/\lambda \in \sigma(A^{-1})$

  The smallest $|\lambda|$ *becomes the largest* $|1/\lambda|$.

- Polynomial preconditioning gives an option when $A^{-1}v$ is too expensive,
  e.g., on distributed memory parallel machines.

- Polynomial preconditioning [Saad 84]. Apply (restarted) Arnoldi to $\pi(A)$:

  eigenvalue $\lambda \in \sigma(A)$ becomes $\pi(\lambda) \in \sigma(\pi(A))$

  Design π *so the smallest* $|\lambda|$ *is mapped to an "easy to find"* $\pi(\lambda)$.

- $\pi(A)v$ requires only matrix-vector products (easy in parallel) . . .
  . . . *but how do you design* π*, and apply it stably?*

Polynomial Preconditioning and Krylov Subspaces

Standard Arnoldi (k steps) extracts eigenvector estimates from

$$\mathcal{K}_k(A, v) = \text{span}\{v, Av, \ldots, A^{k-1}v\}.$$

Preconditioned Arnoldi (k steps) extracts eigenvector estimates from

$$\begin{aligned}\mathcal{K}_k(\pi(A), v) &= \text{span}\{v, \pi(A)v, \ldots, \pi(A)^{k-1}v\} \\ &\subseteq \text{span}\{v, Av, \ldots, A^{d(k-1)}v\} = \mathcal{K}_{dk-d+1}(A, v).\end{aligned}$$

Polynomial Preconditioning and Krylov Subspaces

Standard Arnoldi (k steps) extracts eigenvector estimates from

$$\mathcal{K}_k(A, v) = \text{span}\{v, Av, \ldots, A^{k-1}v\}.$$

Preconditioned Arnoldi (k steps) extracts eigenvector estimates from

$$\begin{aligned}\mathcal{K}_k(\pi(A), v) &= \text{span}\{v, \pi(A)v, \ldots, \pi(A)^{k-1}v\} \\ &\subseteq \text{span}\{v, Av, \ldots, A^{d(k-1)}v\} = \mathcal{K}_{dk-d+1}(A, v).\end{aligned}$$

- Polynomial preconditioning gives access to much higher powers of A, for a fixed subspace dimension k.

  However, each single matvec $\pi(A)v$ requires d matvecs with A.

- Given a good bound for the unwanted eigenvalues in $\sigma(A)$, one could design π using best approximating polynomials: "Arnoldi–Chebyshev" approach [Saad 1984; Ho, Chatelin, Bennani 1990 . . .]). Extensive literature for both polynomial filters for eigenvalue problems, and semi-iterative methods for linear systems.

GMRES-Based Polynomial Preconditioners

- "Hybrid GMRES" approach [Nachtigal, Reichel, Trefethen 1992].
  At step d, GMRES solves the optimization problem

$$\|\pi(A)b\| = \min_{\substack{p \in \mathcal{P}_d \\ p(0)=1}} \|p(A)b\|.$$

  Denote the optimal polynomial as $\pi \in \mathcal{P}_d$, $\pi(0) = 1$:

$$\pi(z) = \left(1 - \frac{z}{\theta_1}\right) \cdots \left(1 - \frac{z}{\theta_d}\right).$$

  Here $\theta_1, \ldots, \theta_d$ denote the roots of GMRES residual polynomial (*harmonic Ritz values*). (These roots are easy to compute; order matters.)

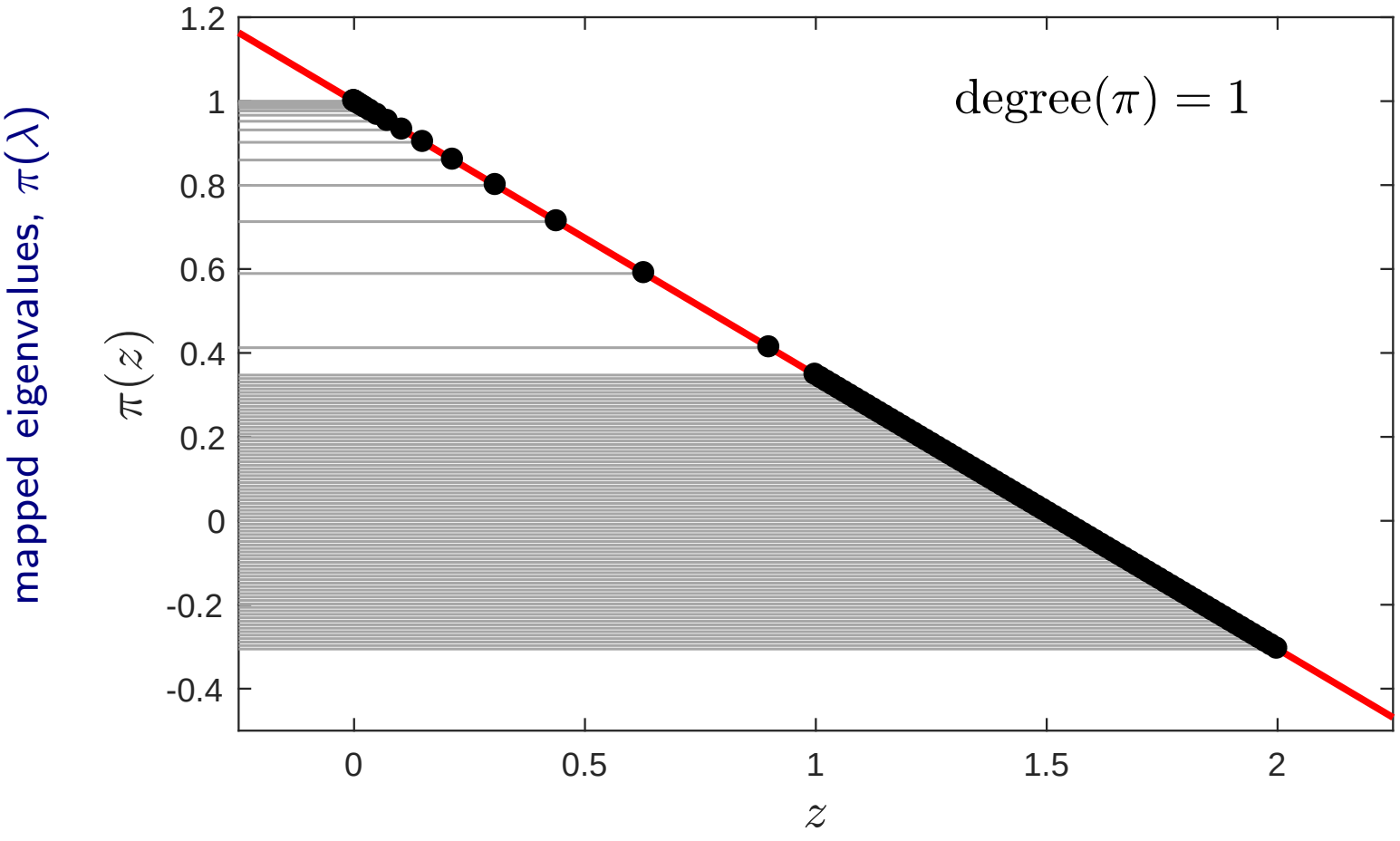
- Contrast with the approach of Heidi Thornquist [2006]:
  – Precondition with the *iteration polynomial* $q(z)$ from, e.g., GMRES:

$$\pi(z) = 1 - zq(z) \quad \Longrightarrow \quad \pi(A) = I - Aq(A) \quad \Longrightarrow \quad q(A) \approx A^{-1}.$$

GMRES Residual Polynomials: An Example

$d = 1$ GMRES iteration

Linear transformation is no help (shift-invariance of Krylov spaces)

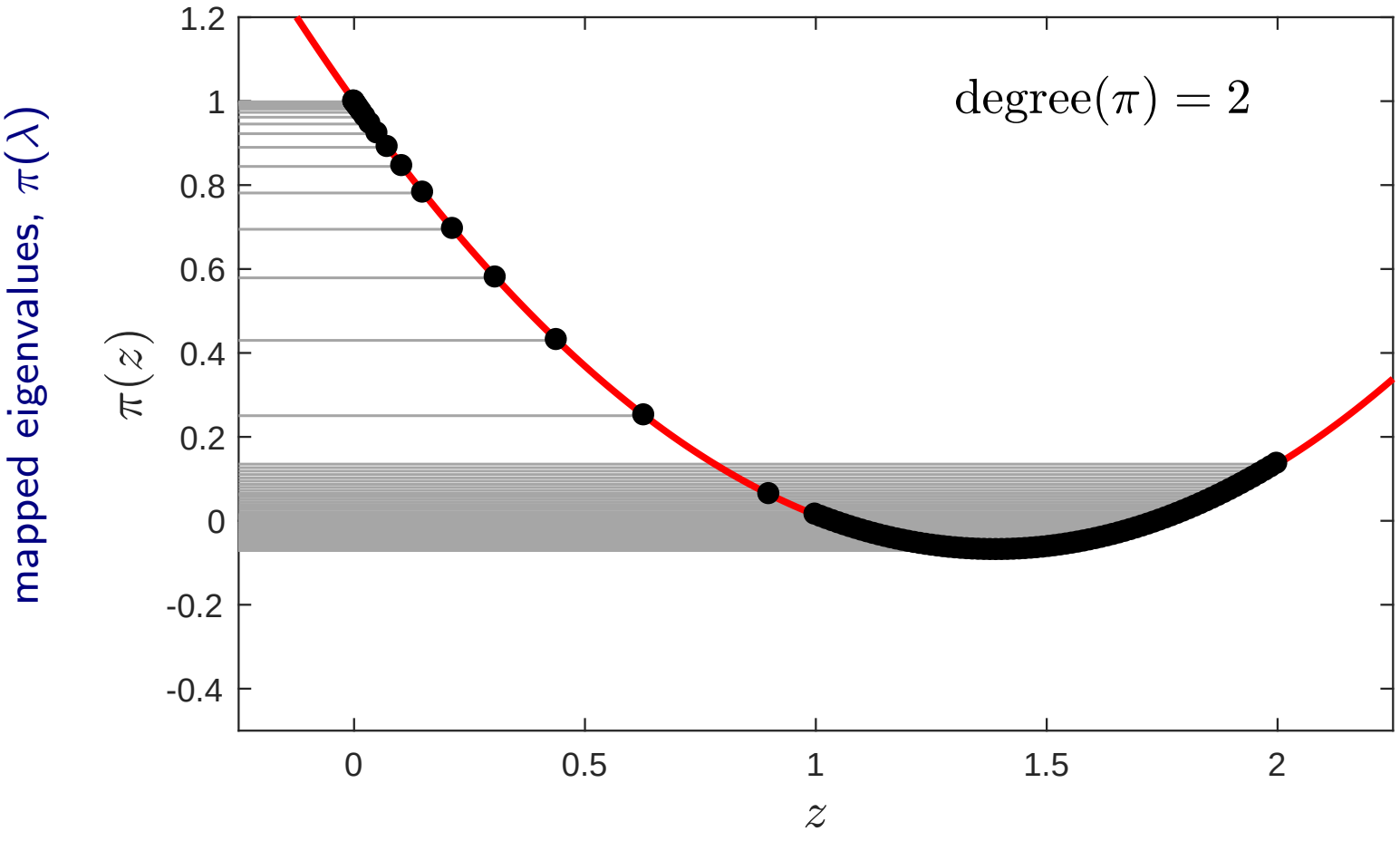


- λ = eigenvalues of A

GMRES Residual Polynomials: An Example

$d = 2$ GMRES iterations

Quadratic transformation separates small eigs, clusters larger eigs

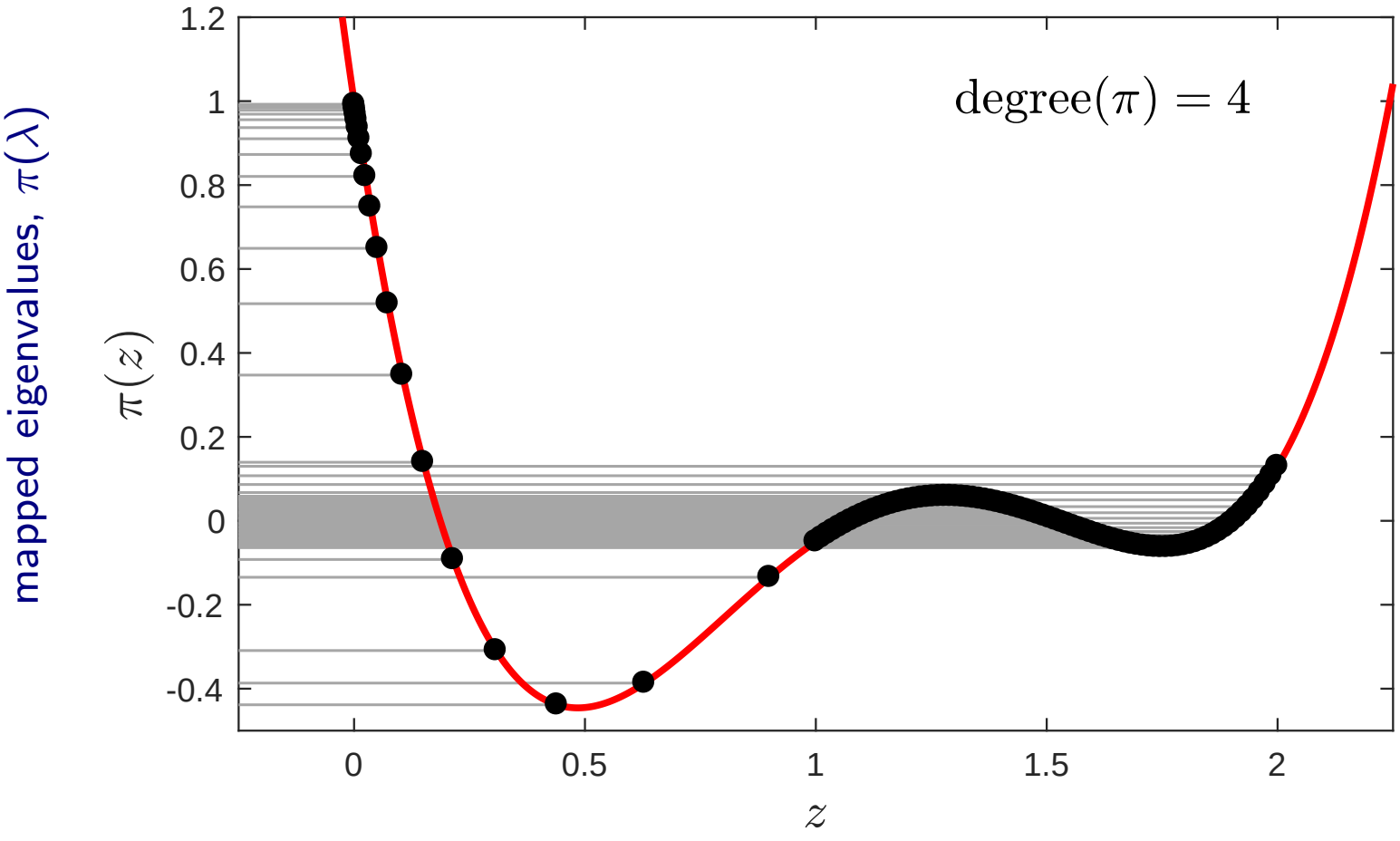


- λ = eigenvalues of A

GMRES Residual Polynomials: An Example

$d = 4$ GMRES iterations

Quartic transformation further separates small eigenvalues

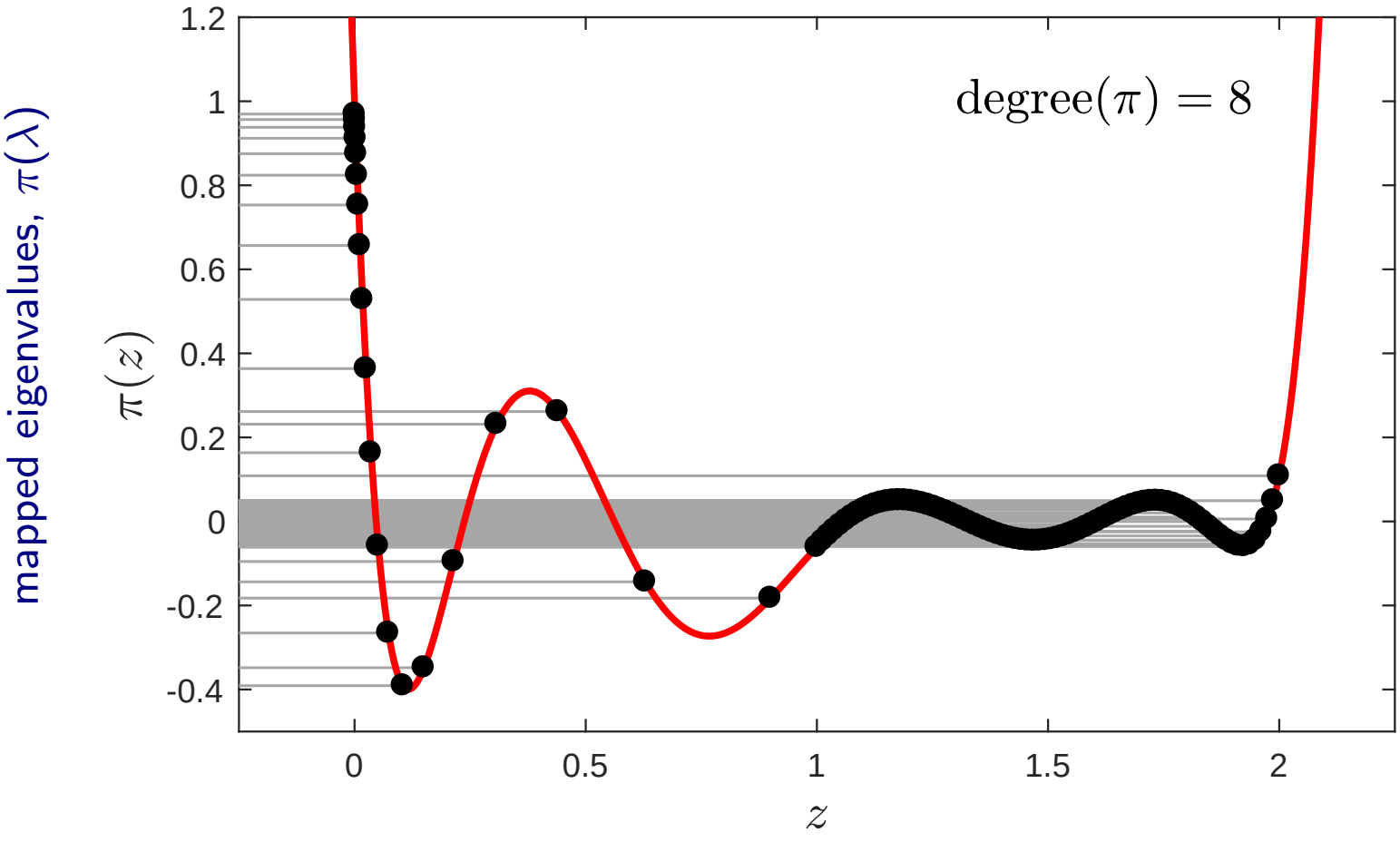


- λ = eigenvalues of A

GMRES Residual Polynomials: An Example

$d = 8$ GMRES iterations

Degree 8 transformation further separates small eigs, mixes up larger eigs



- λ = eigenvalues of A

GMRES Residual Polynomials

$$\pi(z) = \left(1 - \frac{z}{\theta_1}\right) \cdots \left(1 - \frac{z}{\theta_d}\right).$$

- When taking π as the GMRES residual polynomials, we expect

  $$\pi(\lambda) \approx 1$$

  for the eigenvalues λ of A nearest to the origin.
  (Recall that $\pi(0) = 1$ by construction.)

- Extreme case: if $\lambda = 0$, consider GMRES behavior for a singular system.

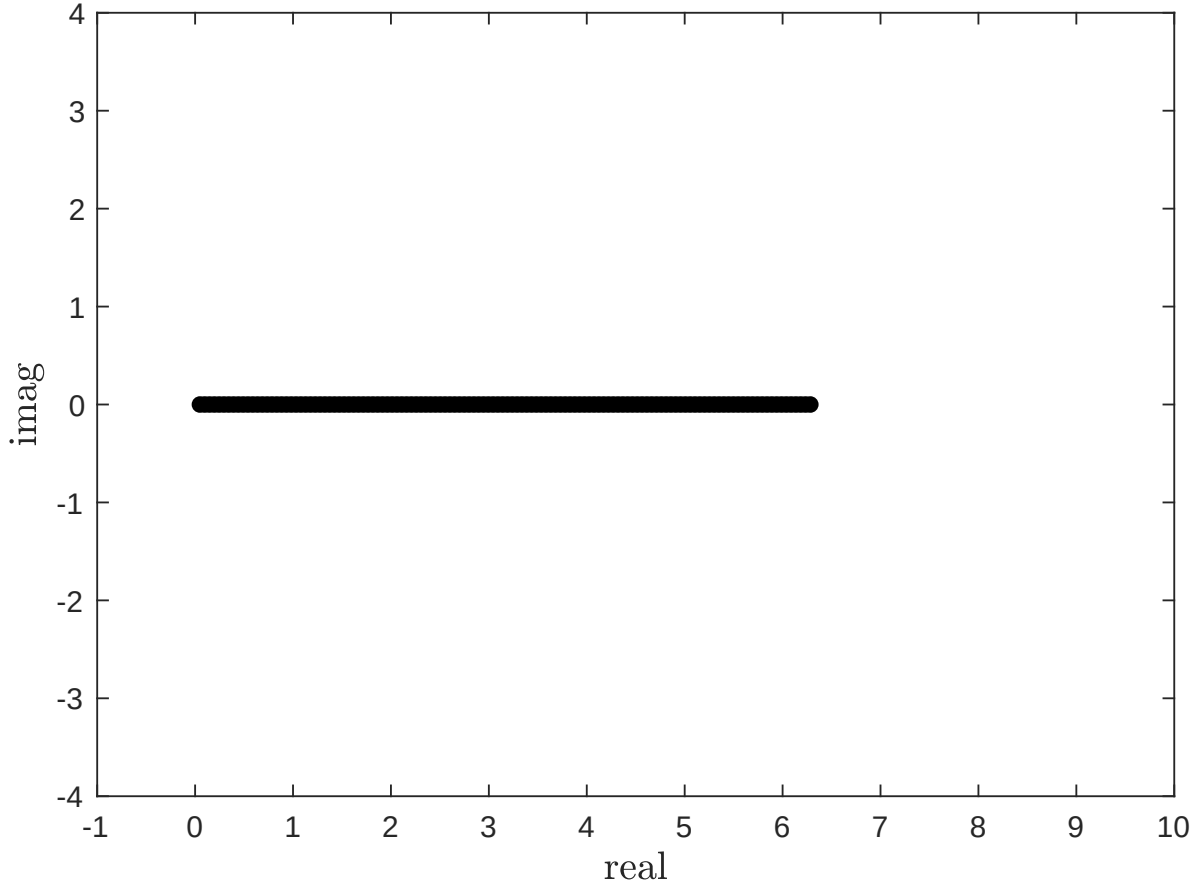
- GMRES handles the nonnormality of A naturally.

- Semi-iterative methods would typically use Ritz information to estimate the spectrum. We use Ritz information to build π directly.

- One must order $\theta_1, \ldots, \theta_d$ to promote stability when computing $\pi(A)v$; [Nachtigal, Reichel, Trefethen 1992] recommend a *modified Leja ordering*.

Roots of π and Nonnormality

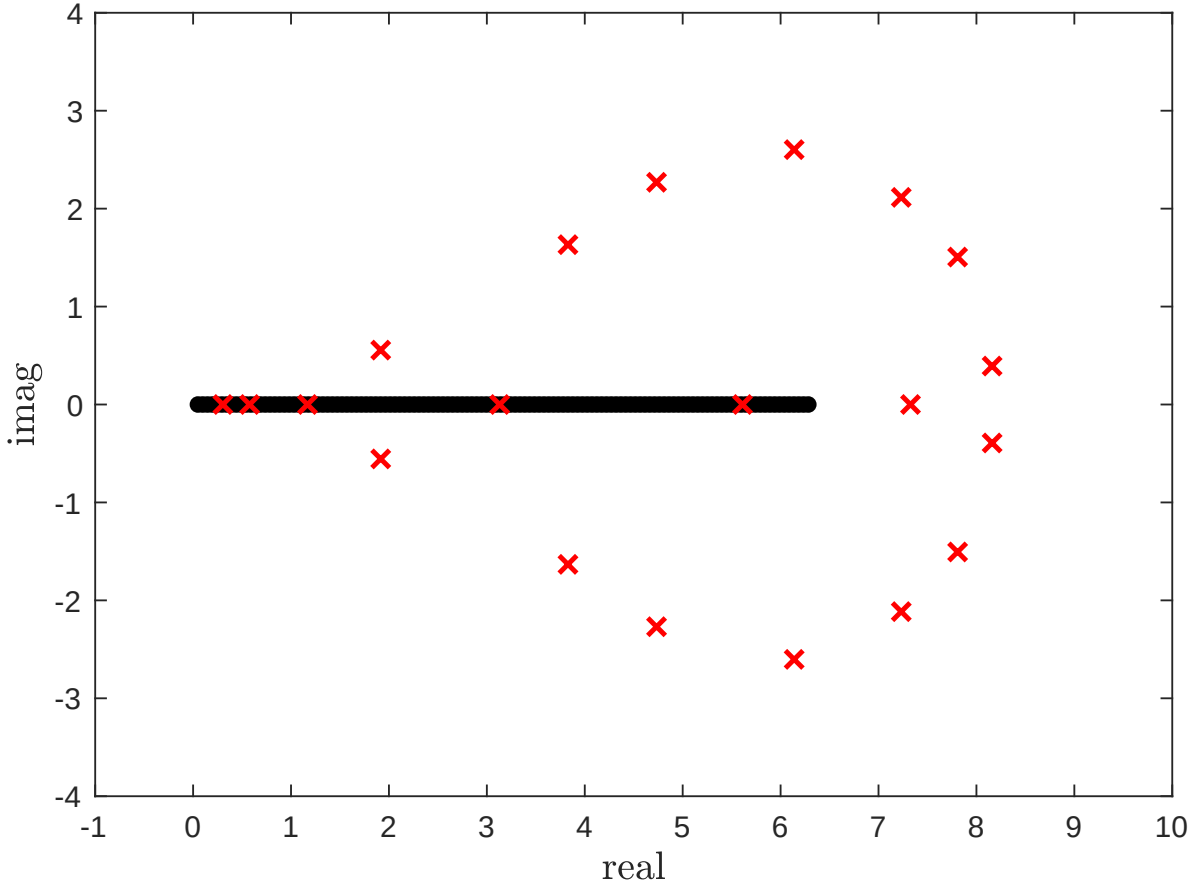
Bidiagonal "twisted Toeplitz" matrix, $N = 100$, real eigenvalues in $(0, 2\pi]$.
[Trefethen, Chapman 2004; Trefethen, E. 2005].



- eigenvalues of A

Roots of π and Nonnormality

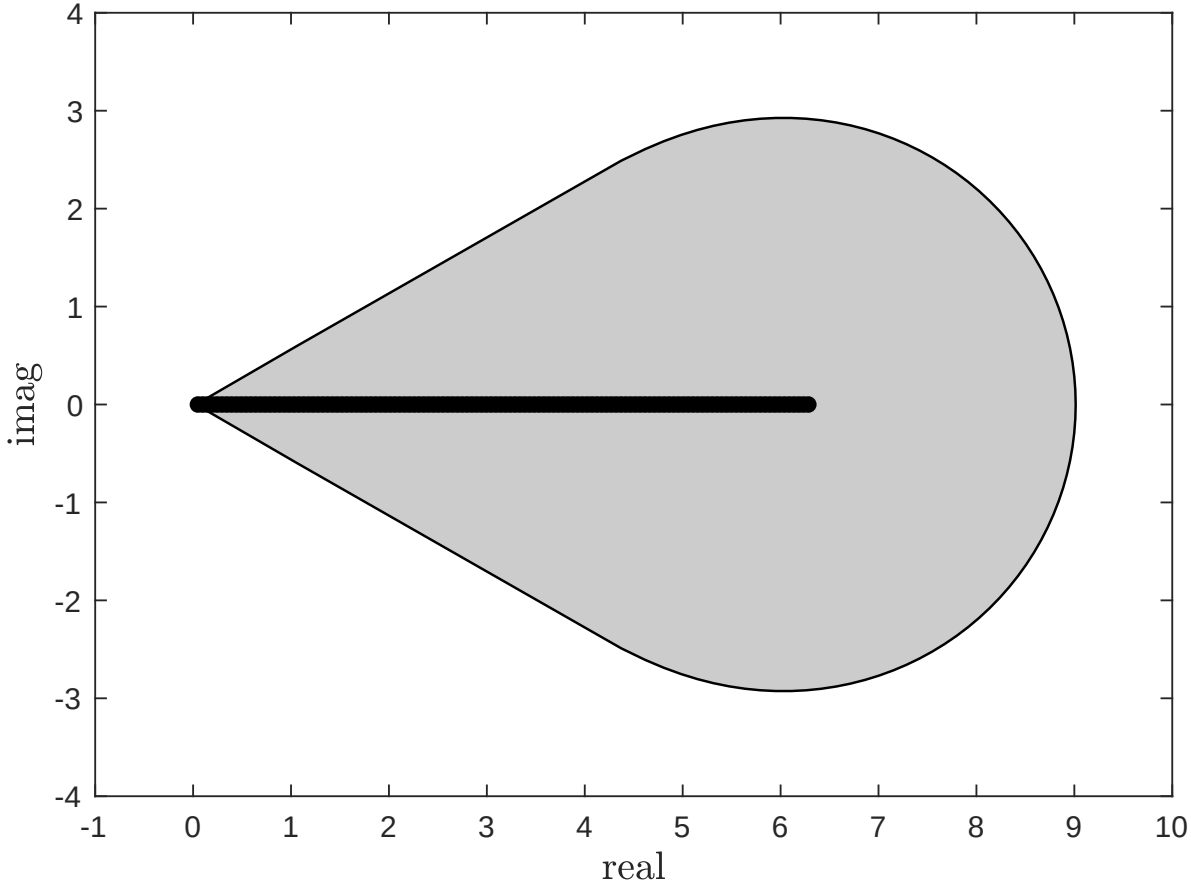
Bidiagonal "twisted Toeplitz" matrix, $N = 100$, real eigenvalues in $(0, 2\pi]$.
[Trefethen, Chapman 2004; Trefethen, E. 2005].



• eigenvalues of A × roots of π

Roots of π and Nonnormality

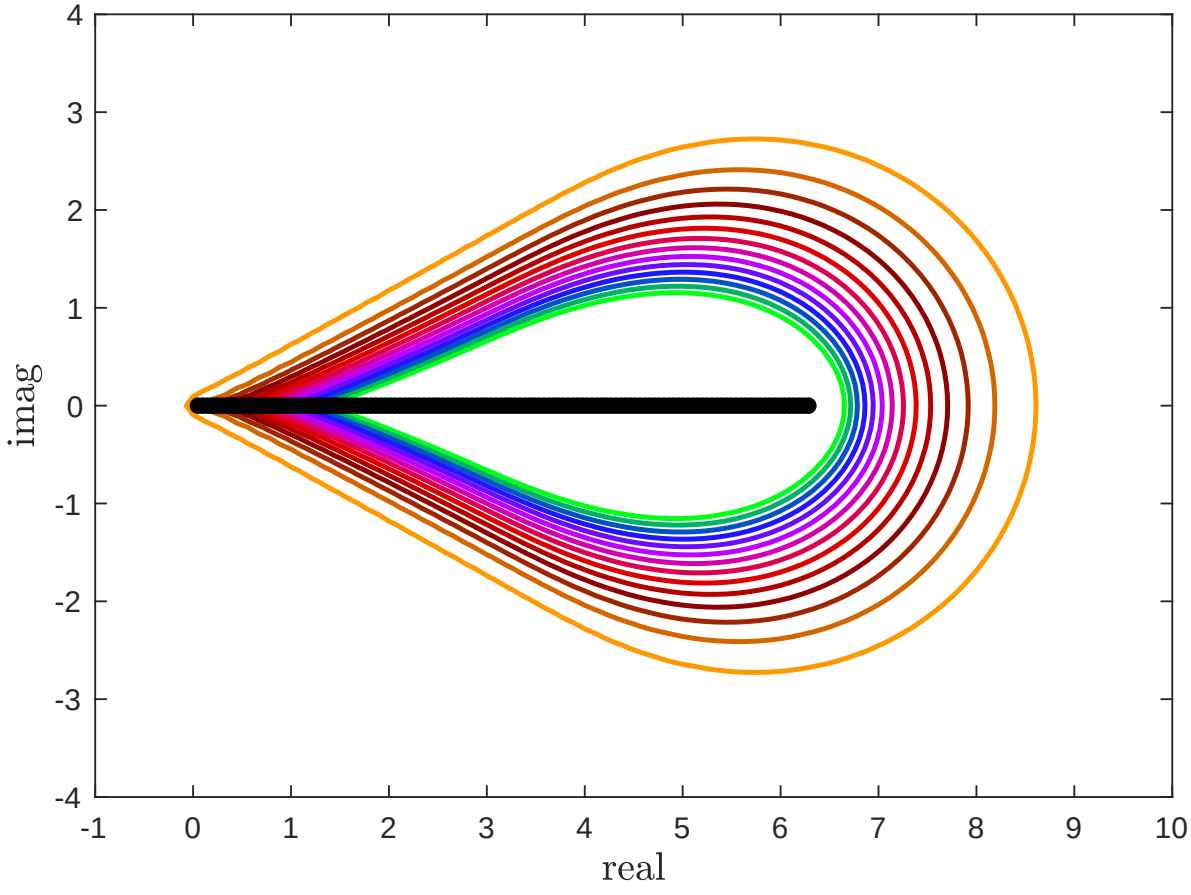
Bidiagonal "twisted Toeplitz" matrix, $N = 100$, real eigenvalues in $(0, 2\pi]$.
[Trefethen, Chapman 2004; Trefethen, E. 2005].



• eigenvalues of A numerical range of A (gray)

Roots of π and Nonnormality

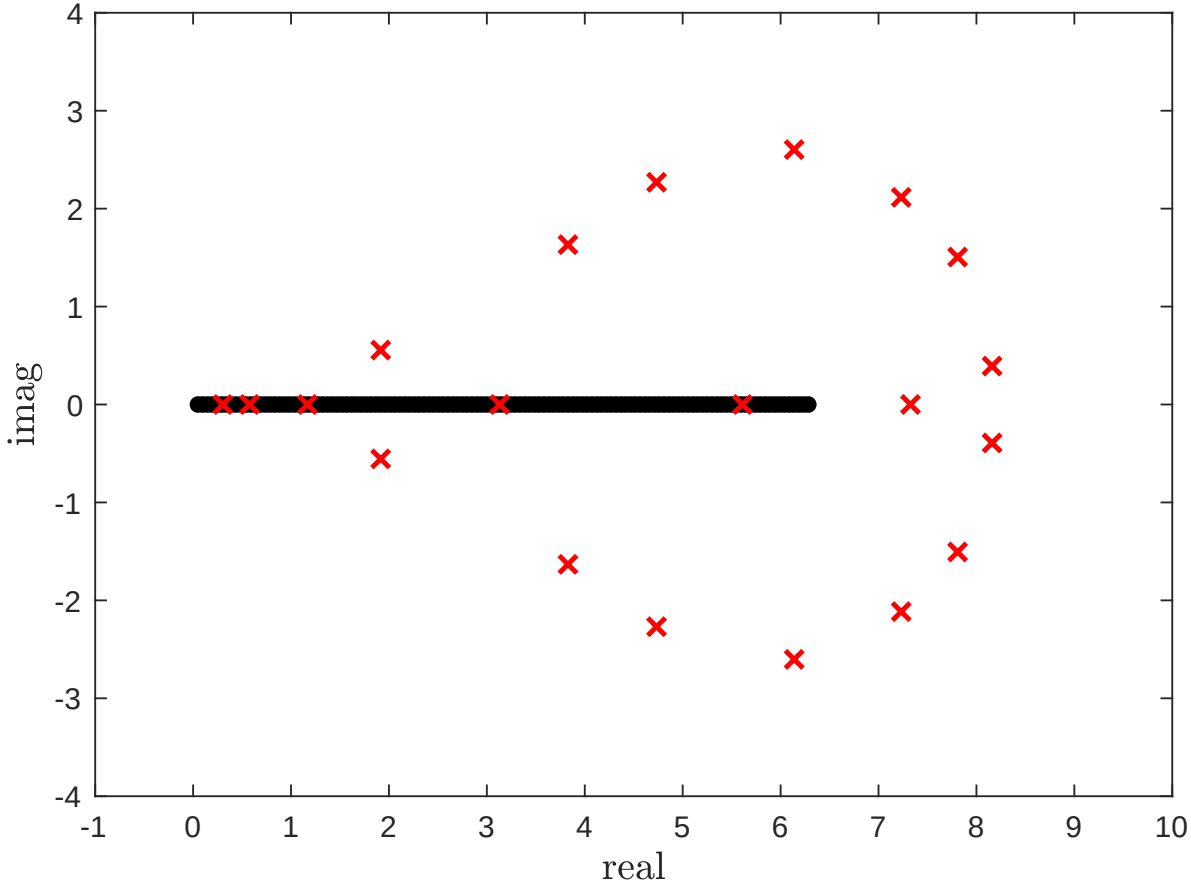
Bidiagonal "twisted Toeplitz" matrix, $N = 100$, real eigenvalues in $(0, 2\pi]$.
[Trefethen, Chapman 2004; Trefethen, E. 2005].



- eigenvalues of A ε-pseudospectra ($\varepsilon = 10^{-1}, 10^{-2}, \ldots, 10^{-14}$)

Roots of π and Nonnormality

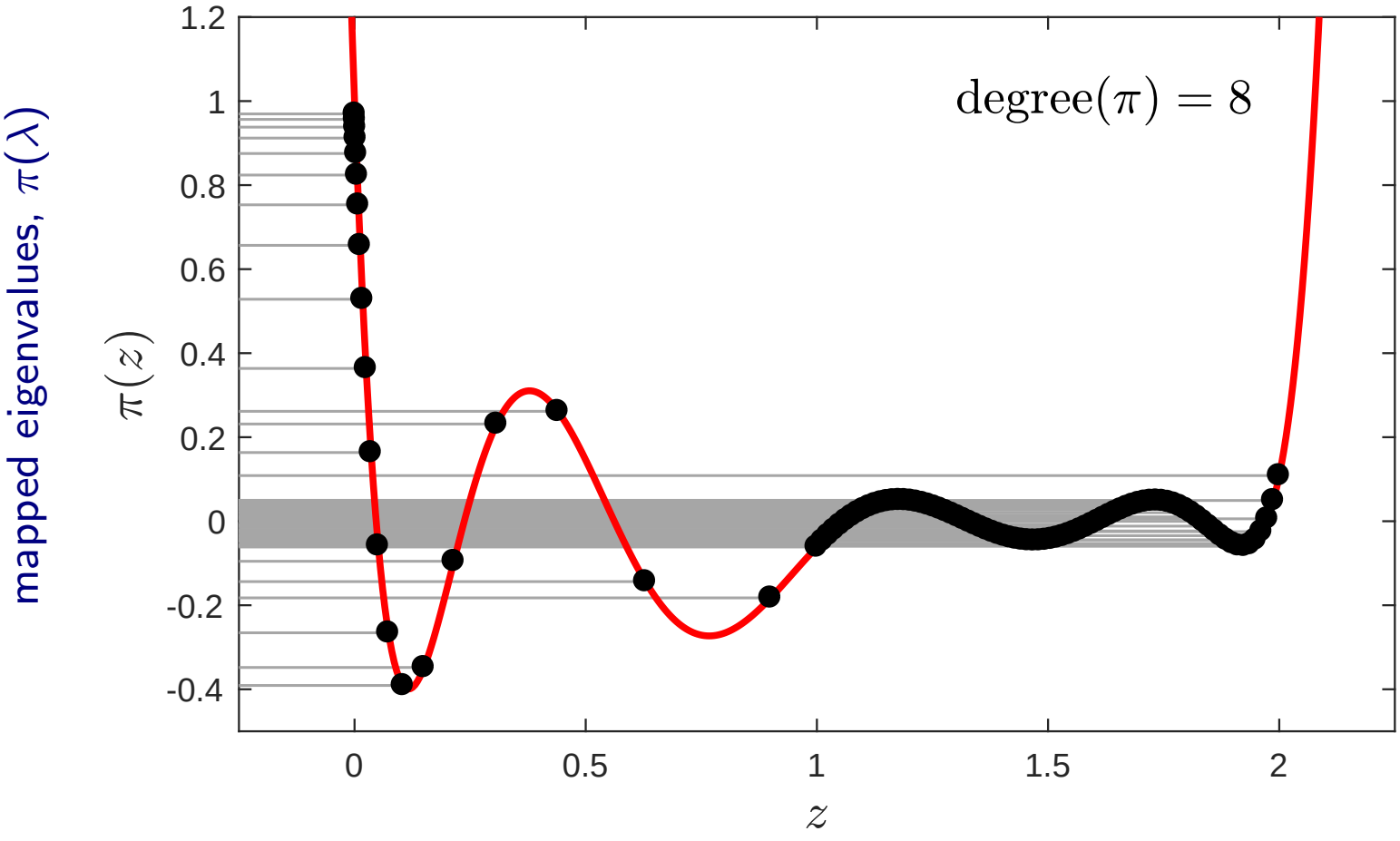
Bidiagonal "twisted Toeplitz" matrix, $N = 100$, real eigenvalues in $(0, 2\pi]$.
[Trefethen, Chapman 2004; Trefethen, E. 2005].



• eigenvalues of A × roots of π

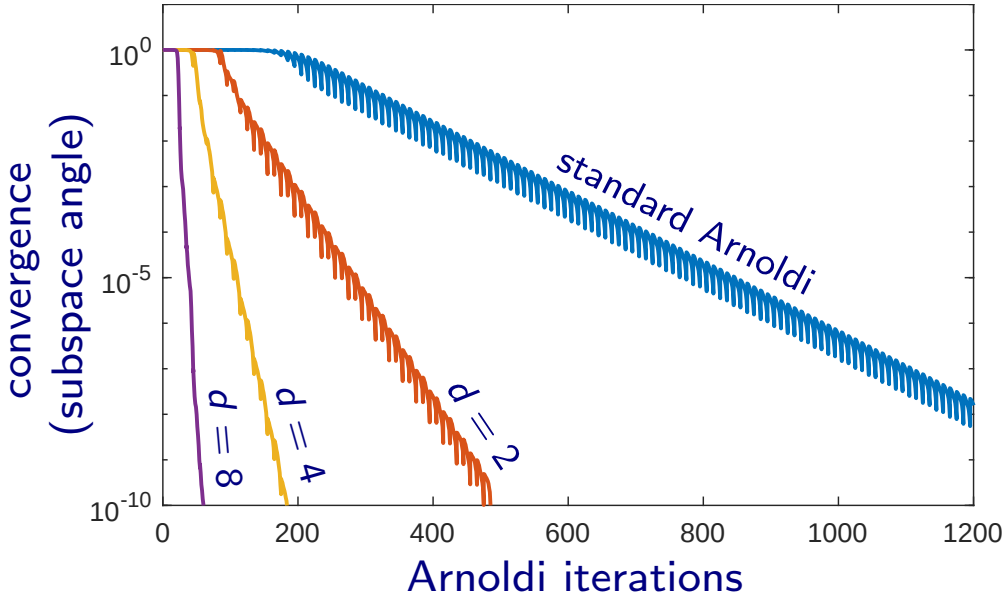
How Does Restarted Arnoldi with π Perform?

Return to the simple example shown earlier.



- λ = eigenvalues of A

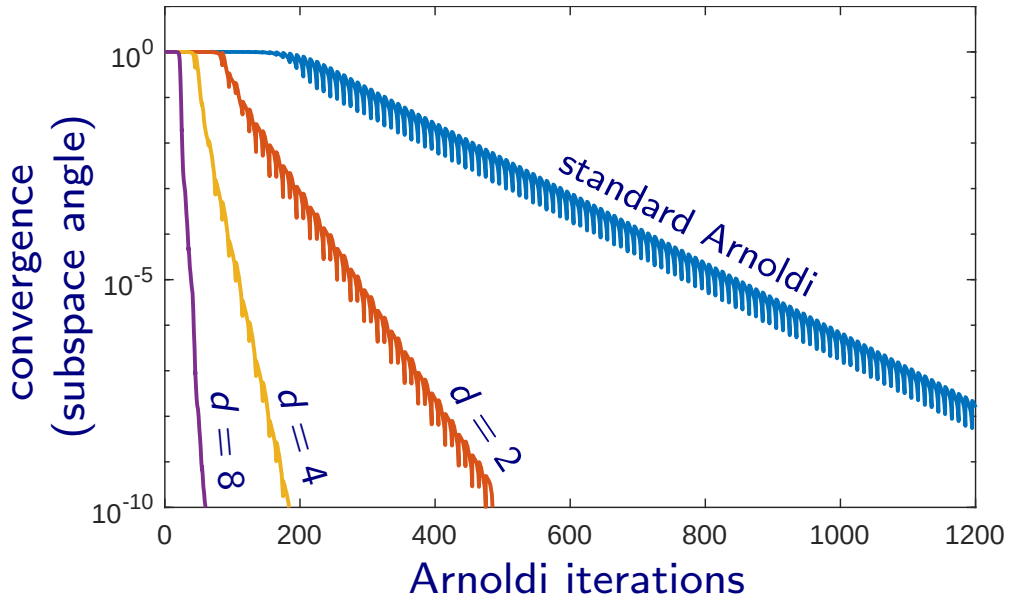
How Well Does it Work?



Compute 5 smallest eigs of A using *restarted* Arnoldi with max subspace dimension = 10: "Arnoldi(10,5)"

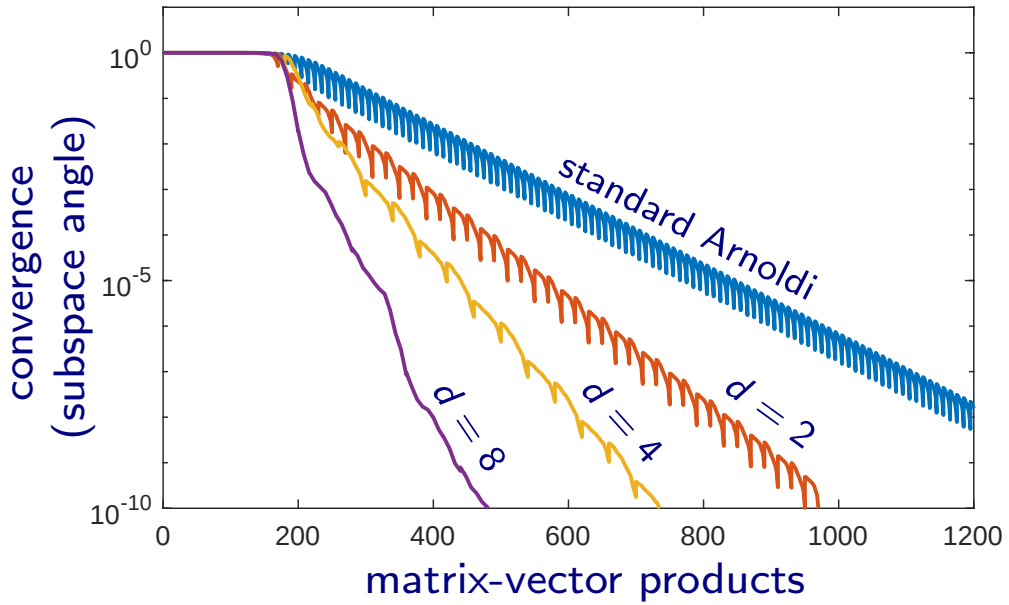
Increasing d can vastly reduce iterations.

How Well Does it Work?



Compute 5 smallest eigs of A using *restarted* Arnoldi with max subspace dimension = 10: "Arnoldi(10,5)"

Increasing d can vastly reduce iterations.



For restarted Arnoldi, preconditioning can even reduce matvecs significantly.

A Simple Convergence Result

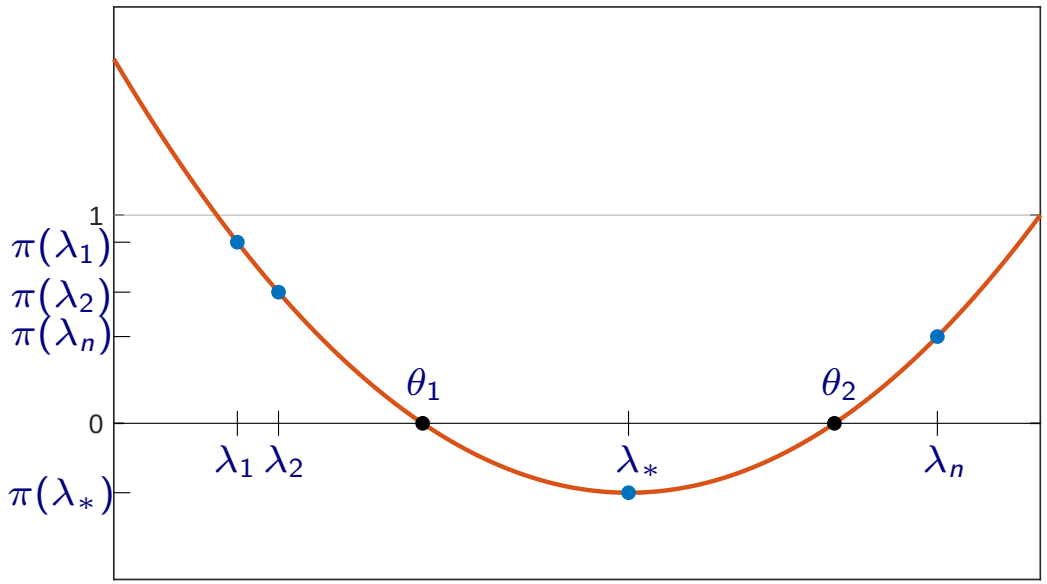
Proposition. Suppose A is symmetric positive definite,

$$0 < \lambda_1 < \lambda_2 < \cdots \leq \lambda_n.$$

Let $\pi(z) = (1 - z/\theta_1)(1 - z/\theta_2)$ be the GMRES polynomial of degree $d = 2$. If

$$\theta_1 + \theta_2 > \lambda_2 + \lambda_n,$$

then the asymptotic convergence rate for hull$(\{\pi(\lambda_j)\}_{j=2}^n)$ is faster than the asymptotic convergence rate for $[\lambda_2, \lambda_n]$.



A Simple Convergence Result

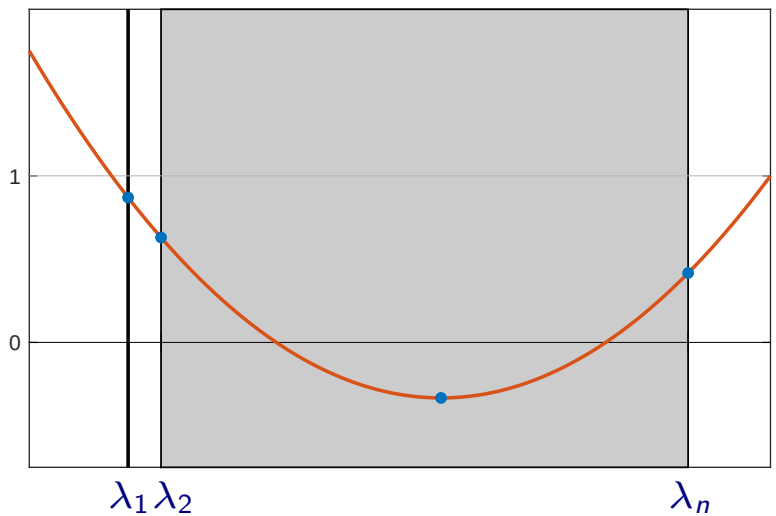
Proposition. Suppose A is symmetric positive definite,

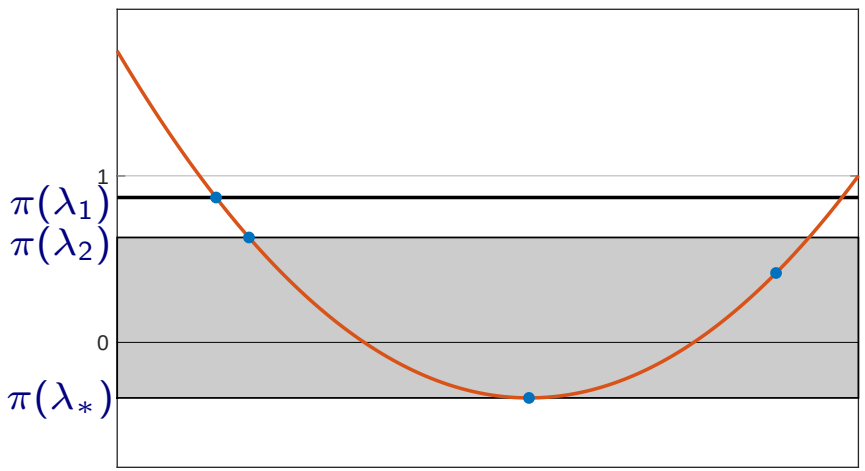
$$0 < \lambda_1 < \lambda_2 < \cdots \leq \lambda_n.$$

Let $\pi(z) = (1 - z/\theta_1)(1 - z/\theta_2)$ be the GMRES polynomial of degree $d = 2$. If

$$\theta_1 + \theta_2 > \lambda_2 + \lambda_n,$$

then the asymptotic convergence rate for hull$(\{\pi(\lambda_j)\}_{j=2}^n)$ is faster than the asymptotic convergence rate for $[\lambda_2, \lambda_n]$.





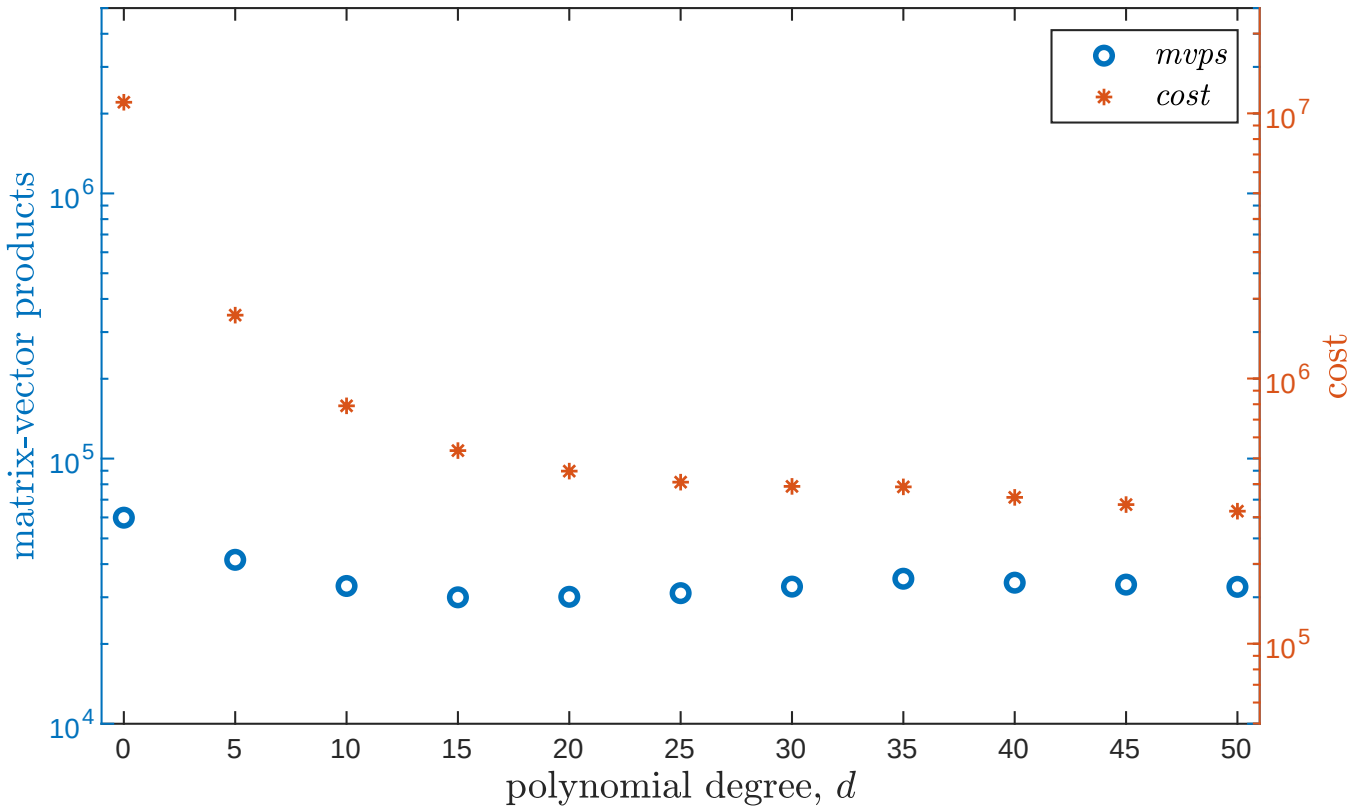
$$\kappa = \frac{\lambda_n - \lambda_1}{\lambda_n - \lambda_2} \qquad \text{rate} = \frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \qquad \kappa = \frac{\pi(\lambda_*) - \pi(\lambda_1)}{\pi(\lambda_*) - \pi(\lambda_2)}$$

Example: Convection–Diffusion Problem

2d Convection-diffusion problem, $n = 160{,}000$; moderately nonnormal.

We seek 15 eigenvalues with a buffer of 5 eigenvalues for restarted Arnoldi, and a maximum subspace dimension of 50: Arnoldi(50,20).

Results averaged over 10 trial runs.



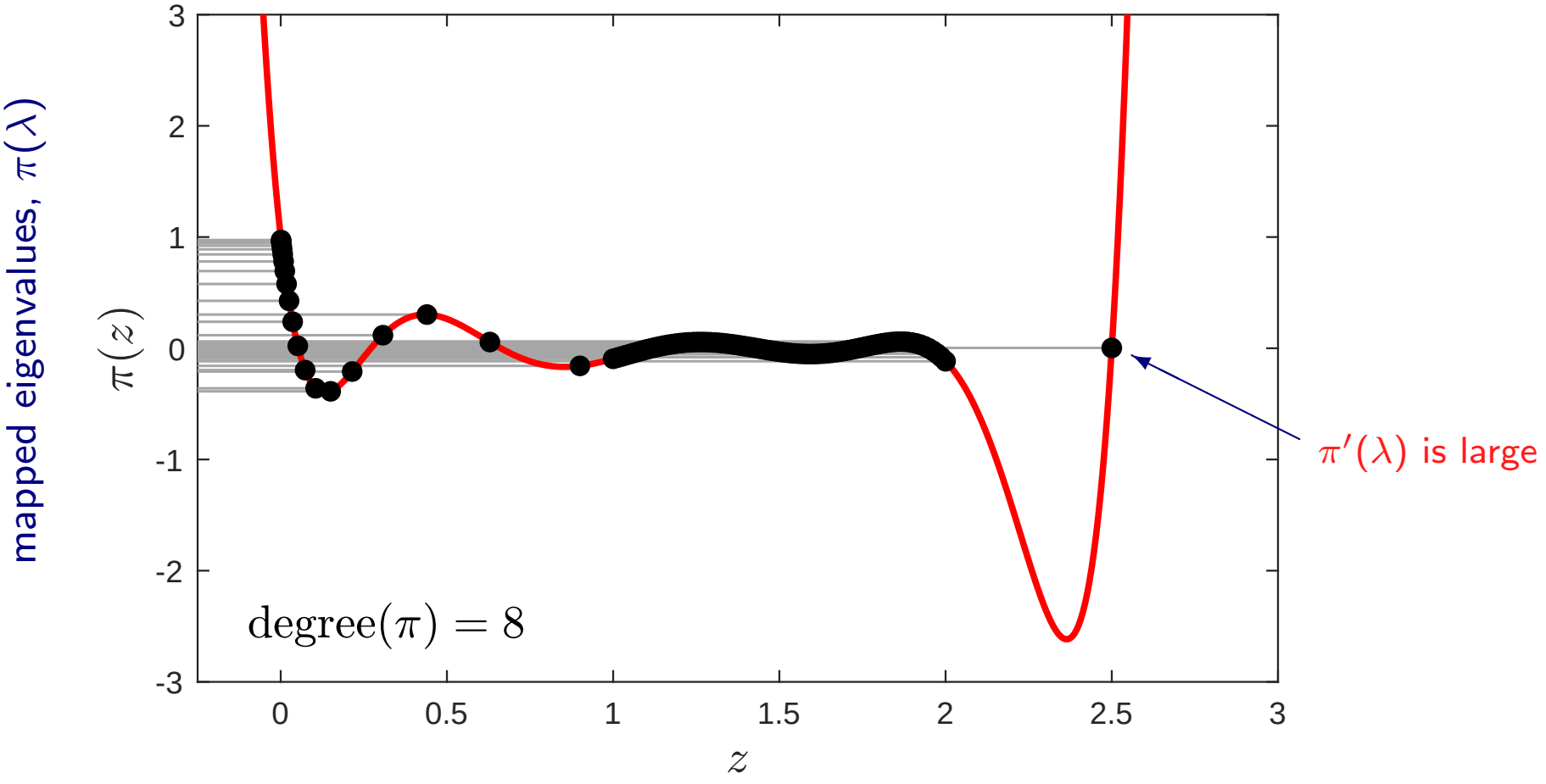
Three Ways to Improve Stability

The basic algorithm is very simple, but several minor adjustments can improve its applicability and reliability.

- Distant eigenvalues of A lead to isolated roots of π, which can complicate the evaluation of $\pi(A)v$.
- Polynomials that oscillate too much over the spectrum can mix up the desired and undesired eigenvalues of $\pi(A)$.
- A poor choice of b could have small components in the desired eigenvectors. (Easy fix: see the paper for details.)

Handle Outlying Eigenvalues by Duplicating Roots

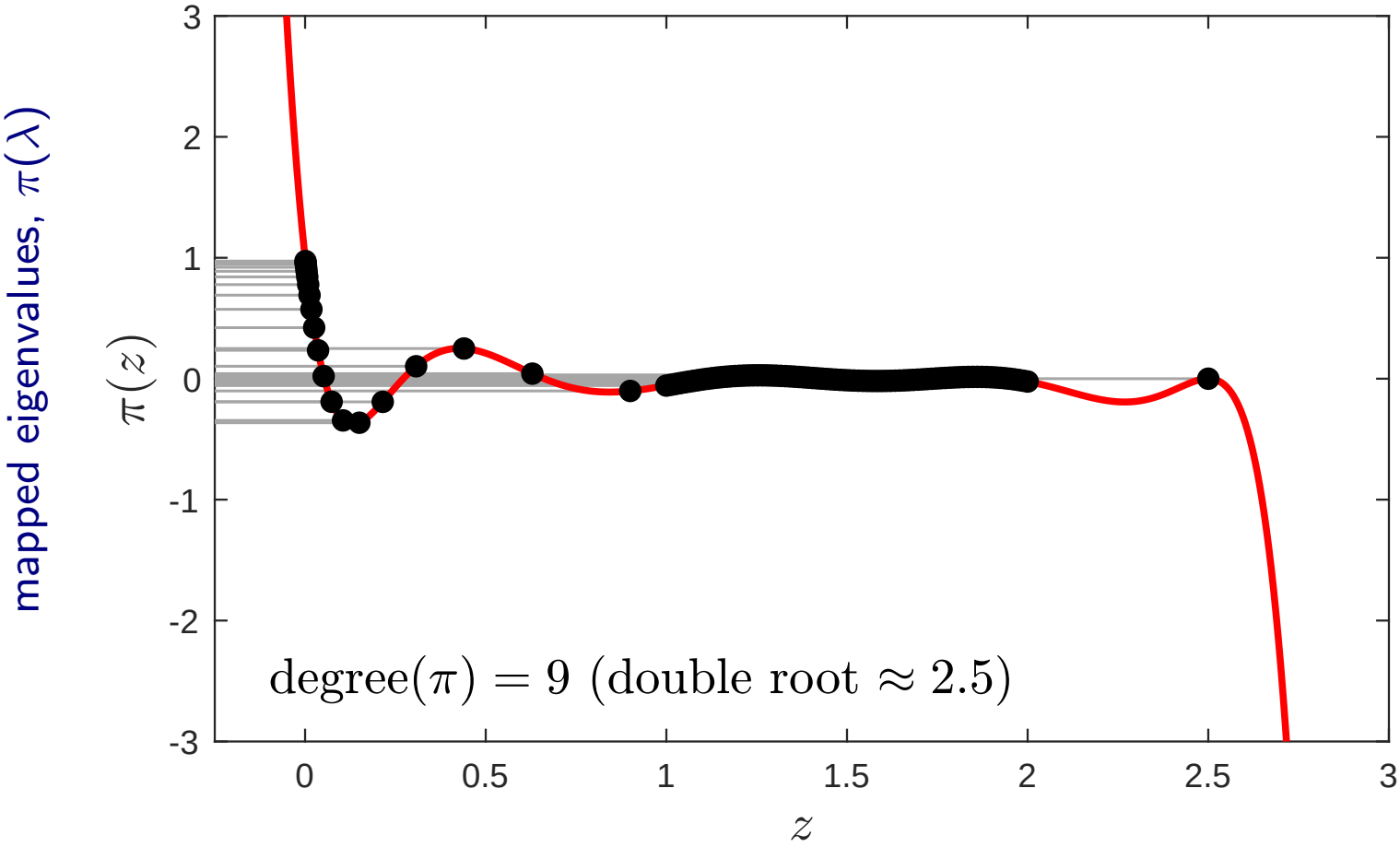
GMRES quickly puts roots very close to isolated "outlier" eigenvalues, but such roots can cause problems due to large $\pi'(\lambda)$ behavior.



- $\lambda =$ eigenvalues of A

Handle Outlying Eigenvalues by Duplicating Roots

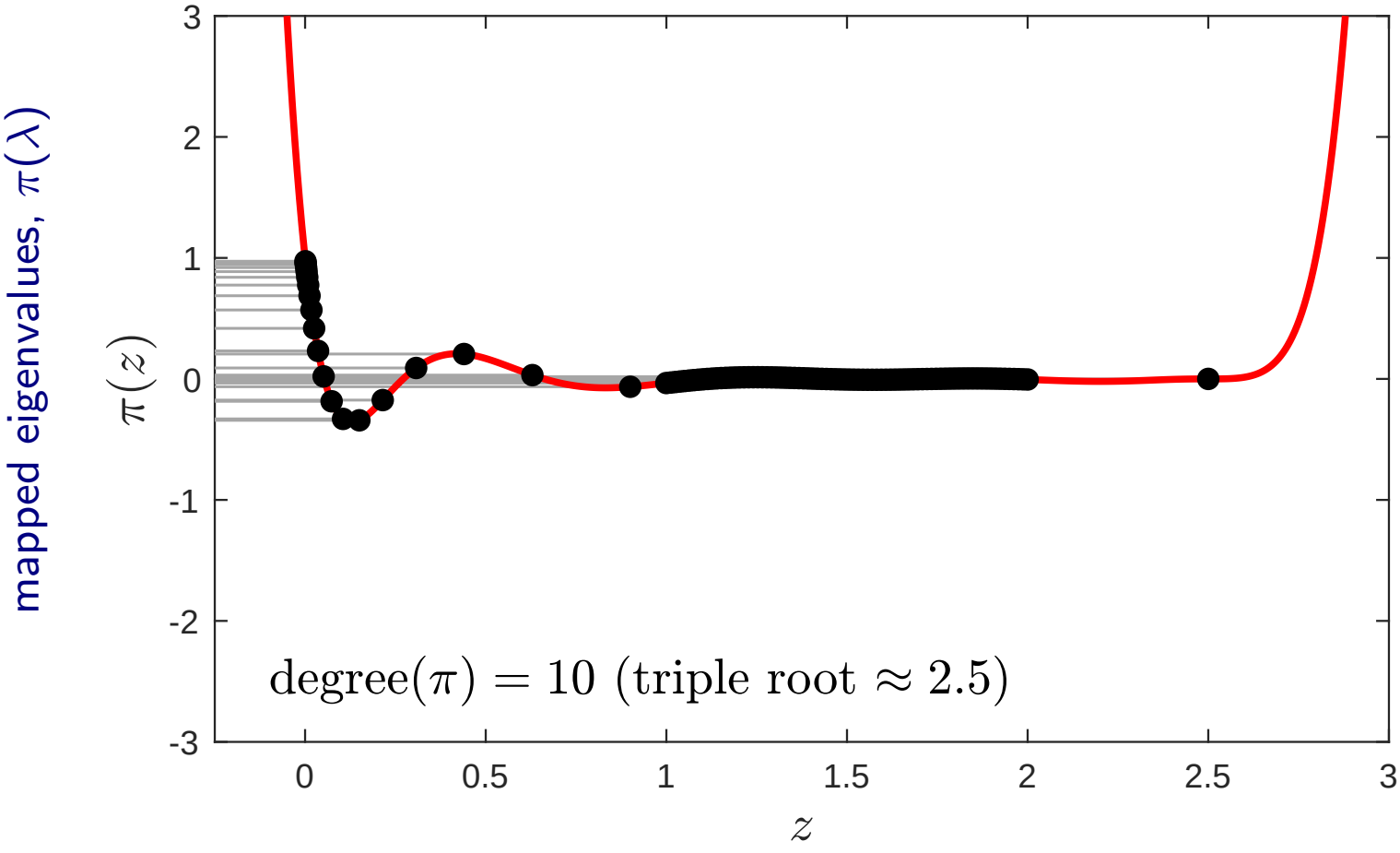
GMRES quickly puts roots very close to isolated "outlier" eigenvalues, but such roots can cause problems due to large $\pi'(\lambda)$ behavior.
Remedy: Add an extra copy (or more) of the problematic root.



- λ = eigenvalues of A

Handle Outlying Eigenvalues by Duplicating Roots

GMRES quickly puts roots very close to isolated "outlier" eigenvalues, but such roots can cause problems due to large $\pi'(\lambda)$ behavior.
Remedy: Add an extra copy (or more) of the problematic root.



- λ = eigenvalues of A

Handle Outlying Eigenvalues by Duplicating Roots

When should you add extra roots, and how many should you add?

How large would $\pi(\theta_j)$ be, were it not for the $(1 - z/\theta_j)$ term?

We define the *product of other factors (pof)* of the jth root:

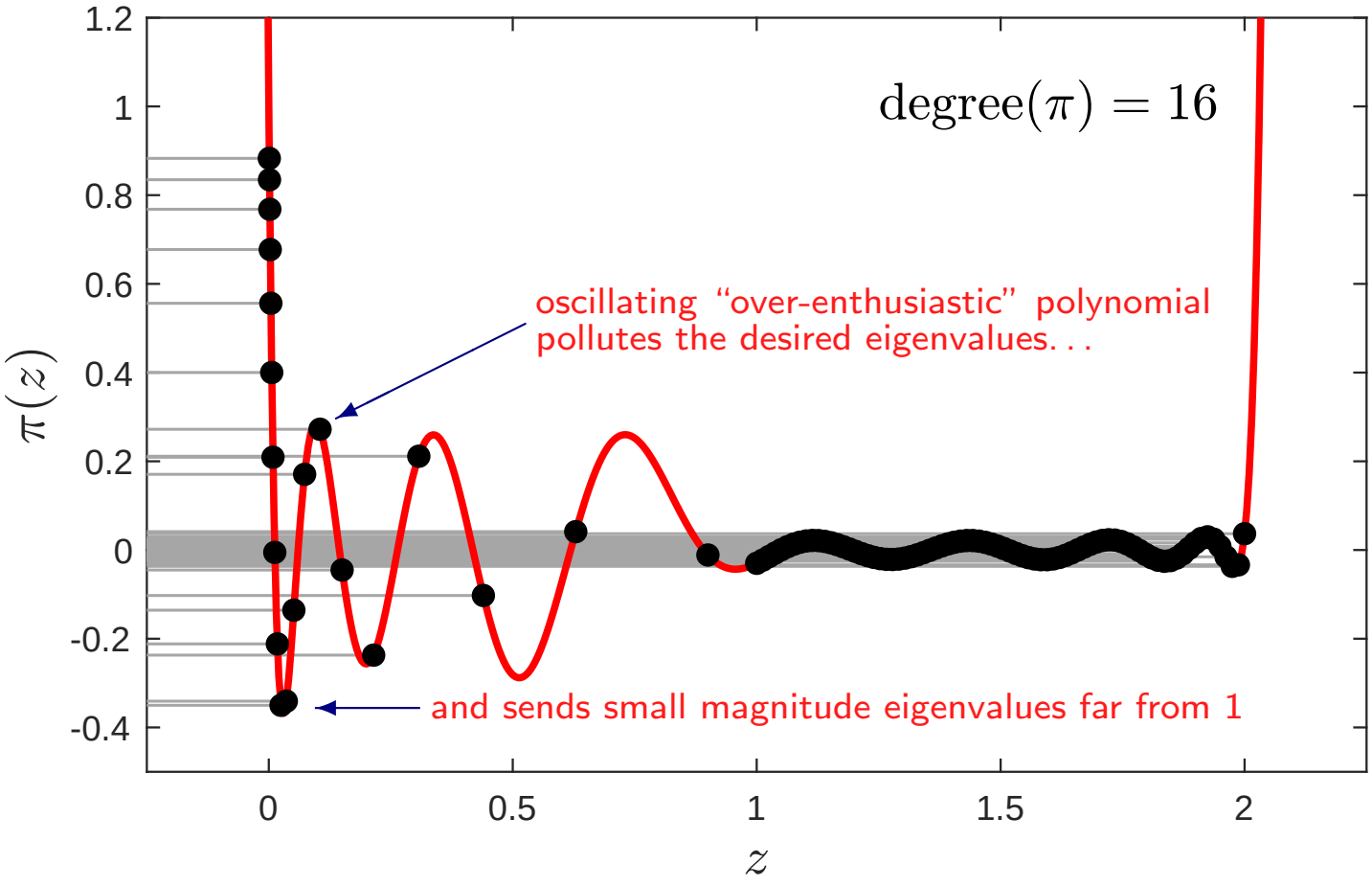
$$\text{pof}(j) := \prod_{i=1, i\neq j}^{d} \left|1 - \frac{\theta_j}{\theta_i}\right| = |\pi'(\theta_j)| \cdot |\theta_j|.$$

Rule of thumb (for double precision computations):

$$\text{Add } \left\lceil \frac{\log_{10}(\text{pof}(j)) - 4}{14} \right\rceil \text{ additional } \left(1 - \frac{z}{\theta_j}\right) \text{ factors to } \pi(z).$$

Damping to Enhance Smallest Magnitude Eigenvalues

For some spectral distributions, larger d can give π that are "*over-enthusiastic*".
Oscillations near the origin mix up the order of eigenvalues in $\sigma(\pi(A))$.



Taming Over-Enthusiastic Polynomials with Damping

For some spectral distributions, larger d can give π that are "*over-enthusiastic*". Oscillations near the origin mix up the order of eigenvalues in $\sigma(\pi(A))$.

Suppose A is diagonalizable, having eigenvectors $v_1, \ldots, v_n$.

- Let $v_1, \ldots, v_n$ be a basis of eigenvectors of A, and write

$$b = \sum_{j=1}^{n} (c_j) v_j.$$

- Then

$$Ab = \sum_{j=1}^{n} (\lambda_j c_j) v_j.$$

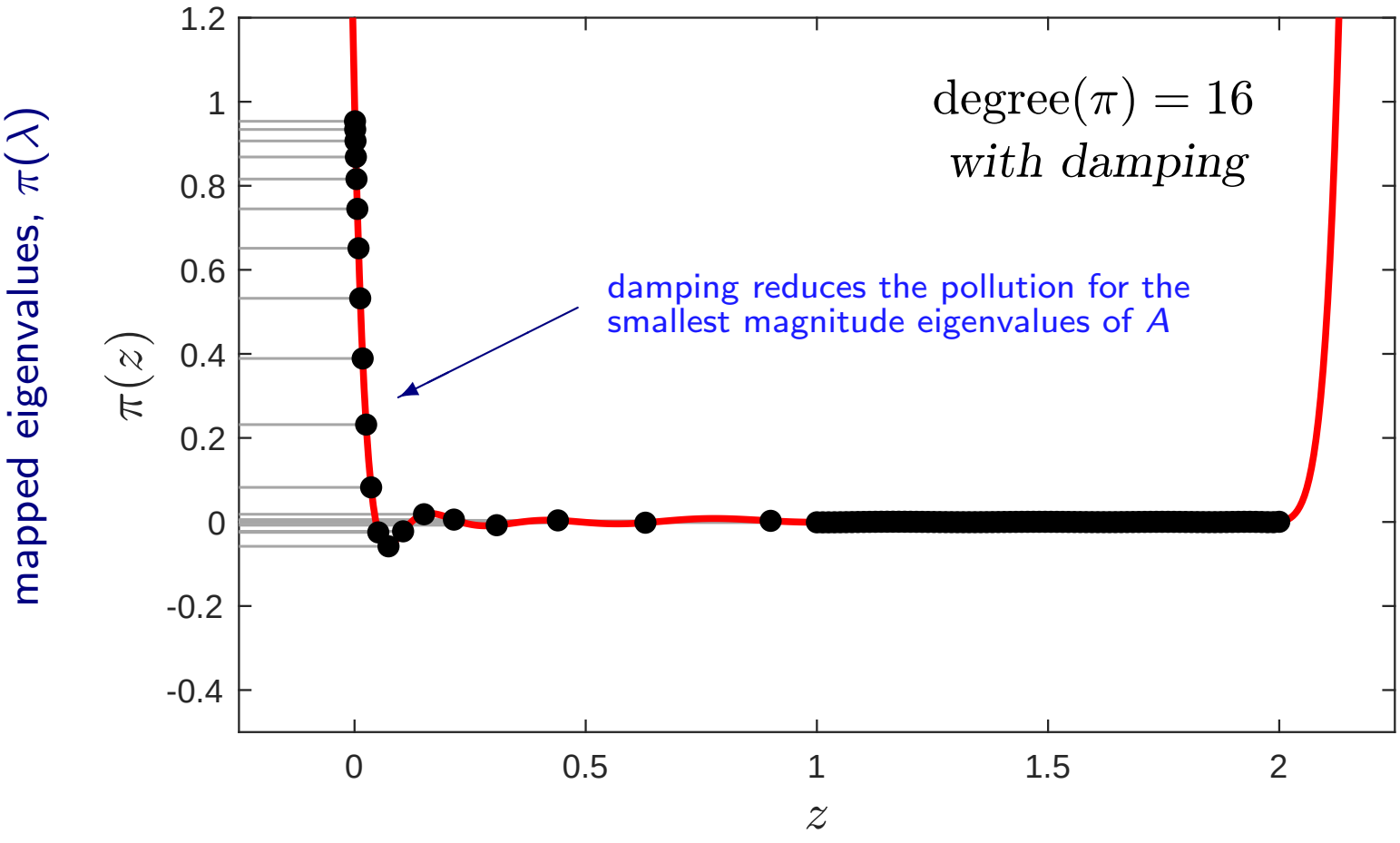
  Premultiplying by A *damps* the components of b associated with the smallest magnitude eigenvalues.

- Generate π by applying GMRES to (A, Ab) instead of (A, b).

- Compare to damping Gibbs phenomenon in Dirac filters in [Li, Xi, Vecharynski, Yang, Saad 2016]

Damping to Enhance Smallest Magnitude Eigenvalues

For some spectral distributions, larger d can give π that are "*over-enthusiastic*".
Oscillations near the origin mix up the order of eigenvalues in $\sigma(\pi(A))$.



- $\lambda =$ eigenvalues of A

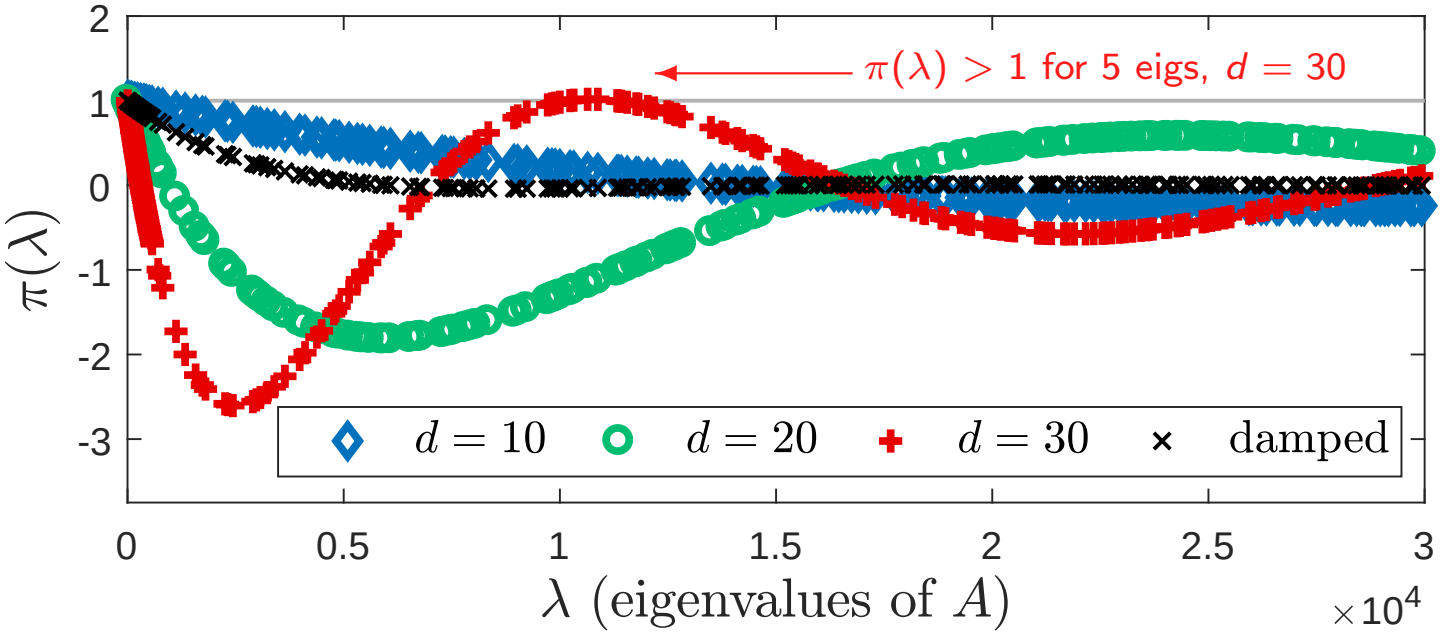
Damping to Enhance Smallest Magnitude Eigenvalues

An example from SuiteSparse:

S1rmq4m1 ($n = 5489$, symmetric positive definite)

Degree $d = 30$ polynomial is *over-enthusiastic* (red $+$).

Damping (generating π with GMRES on (A, Ab)) fixes the problem (black $\times$).



Double Polynomial Preconditioning

For large d, the GMRES run to form π can incur many dot products.

Here is a strategy to leverage very high powers of A with fewer dot products.

- Form the GMRES polynomial π_1 of degree d_1 for (A, b).
  The desired eigenvalues of $\pi(A)$ should be near 1.
- Define the matrix $\tau(A) = I - \pi(A)$.
  The desired eigenvalues of $\tau(A)$ should be near 0.
- Form the GMRES polynomial π_2 of degree d_2 for $(\tau(A), b)$.
  The desired eigenvalues of $\pi_2(\tau(A))$ should be near 1.
- Run restarted Arnoldi on $\pi_2(\tau(A)) = \pi_2(I - \pi_1(A))$.
  Notice that $\pi_2(1 - \pi_1(z))$ is a degree $d_1 d_2$ polynomial.

Double Polynomial Preconditioning: Example

2d Convection–Diffusion problem, $n = 640{,}000$.

degree d or $d_1 \times d_2$	cycles	matvecs (thousands)	time (minutes)	dot products (thousands)
Polynomial Preconditioned Arnoldi				
0	6924.5	207.8	243.4	15999.9
10	253.2	76.6	19.4	561.9
25	82.7	63.0	11.3	185.1
50	41.2	63.6	9.7	95.4
100	20.6	64.8	9.2	57.8
125	16.5	65.3	8.8	56.3
150	14.0	67.0	9.1	60.4
Double Polynomial Preconditioning				
$15 \times 20 = 300$	3.8	41.0	1.4	9.7
$15 \times 40 = 600$	2.0	48.9	1.6	6.9
$15 \times 50 = 750$	2.0	61.2	2.0	7.9
$25 \times 40 = 1000$	1.0	51.0	1.6	5.2
$25 \times 60 = 1500$	1.0	76.5	2.4	8.0

Double Polynomial Preconditioning: Example

2d Convection–Diffusion problem, $n = 640{,}000$.
Run Arnoldi(50,20) to compute the smallest 15 eigenvalues.

degree d or $d_1 \times d_2$	cycles	matvecs (thousands)	time (minutes)	dot products (thousands)
Polynomial Preconditioned Arnoldi				
0	6924.5	207.8	243.4	15999.9
10	253.2	76.6	19.4	561.9
25	82.7	63.0	11.3	185.1
50	41.2	63.6	9.7	95.4
100	20.6	64.8	9.2	57.8
125	16.5	65.3	8.8	56.3
150	14.0	67.0	9.1	60.4
Double Polynomial Preconditioning				
$15 \times 20 = 300$	3.8	41.0	1.4	9.7
$15 \times 40 = 600$	2.0	48.9	1.6	6.9
$15 \times 50 = 750$	2.0	61.2	2.0	7.9
$25 \times 40 = 1000$	1.0	51.0	1.6	5.2
$25 \times 60 = 1500$	1.0	76.5	2.4	8.0

Summary

Polynomial Preconditioning for Eigenvalue Computations
Using the GMRES Residual Polynomial

- The GMRES residual polynomial can be an appealing choice of precondition for Arnoldi eigenvalue computations.
- This choice of π adapts naturally to nonnormality and does not require an initial estimate of the spectrum.
- Good choices of π can reduce the *matrix-vector products*, *dot products*, *Arnoldi iterations*, and *computation time*.
- Simple modifications can address stability considerations: *duplicating tricky roots*, *damping b*, *using multiple b*.
- Double polynomial preconditioning gives access to very high powers of A, and can be especially helpful for minimizing dot products.