Numerical-Integration Based Eigensolvers

Ping Tak Peter Tang
Intel Corporation

ARITH 22 Post-Conference Meeting
June 25, 2015

- Numerical linear algebra is a core subject in computational methods
- The two major themes within numerical linear algebra are
    1. Solutions of linear system: Given matrix A and vector $\mathbf{b}$, find vector $\mathbf{x}$ such that $A\mathbf{x} \approx \mathbf{b}$
    2. Eigenproblems:
        - Standard eigenproblem: Given matrix A, find scalar λ and vector $\mathbf{x}$ such that $A\mathbf{x} = \lambda\mathbf{x}$
        - Generalized eigenproblem: Given matrices A and B, find scalar λ and vector $\mathbf{x}$ such that $A\mathbf{x} = \lambda B\mathbf{x}$.

Eigenvalue problems is of major importance in many scientific and engineering applications. Quoting verbatim from Saad's textbook:

- Structural dynamics
- Electrical networks
- Combustion processes
- Macro economics
- Normal mode techniques
- Quantum chemistry
- Markov chain techniques
- Chemical reactions
- Magneto hydrodynamics
- Control theory

- When the matrix is of moderate size, there are well established method and robust software, such as LAPACK.
- When the underlying matrix is large and perhaps sparse, solving for part of the spectrum is still an active research area
- Recently, a new type of methods have been proposed that involve the use of analytic function theory and numerical integration
  - Sakurai and collaborators used a pole finding technique (SS algorithms)
  - Polizzi and collaborators used an approximate subspace projector (FEAST algorithm)
  - Both use numerical integration
- I will mainly use the FEAST algorithm to introduce these integration-based methods

High-Level Outline

- FEAST as filtered subspace iteration, for Hermitian problems
- FEAST for non-Hermitian problems
- SS method as filtered Krylov subspace method
- Function approximation and computer arithmetic studies related to the filter

FEAST for Hermitian Problems

- FEAST as filtered subspace iteration for standard Hermitian eigenproblem
- The approximate spectral projector filter
- Convergence analysis of filtered subspace iteration
- FEAST for generalized Hermitian problem

Review: Subspace Iteration for HEP

Recall the standard fact about Hermitian matrices.

For any Hermitian matrix M, n-by-n,
M can be diagonalized by a unitary matrix U:

$$M = U \Lambda U^H, \qquad \Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n), \quad U^H U = I$$

U consists of n orthogonal eigenvectors,
and the $\lambda_j \in \mathbb{R}$ are eigenvalues.

Review: Subspace Iteration for HEP

Recall the standard fact about Hermitian matrices.

For any Hermitian matrix M, n-by-n,
M can be diagonalized by a unitary matrix U:

$$M = U\Lambda U^H, \qquad \Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n), \quad U^H U = I$$

U consists of n orthogonal eigenvectors,
and the $\lambda_j \in \mathbb{R}$ are eigenvalues.

Matrix A

$$\begin{bmatrix} 4 & 3 & 3 \\ 3 & 4 & 3 \\ 3 & 3 & 4 \end{bmatrix}$$

Review: Subspace Iteration for HEP

Recall the standard fact about Hermitian matrices.

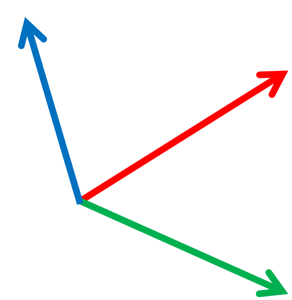
For any Hermitian matrix M, n-by-n,
M can be diagonalized by a unitary matrix U:

$$M = U \Lambda U^H, \qquad \Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n), \quad U^H U = I$$

U consists of n orthogonal eigenvectors,
and the $\lambda_j \in \mathbb{R}$ are eigenvalues.

Matrix A 10   1   1

$$\begin{bmatrix} 4 & 3 & 3 \\ 3 & 4 & 3 \\ 3 & 3 & 4 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$



Review: Subspace Iteration for HEP

Recall the standard fact about Hermitian matrices.

For any Hermitian matrix M, n-by-n,
M can be diagonalized by a unitary matrix U:

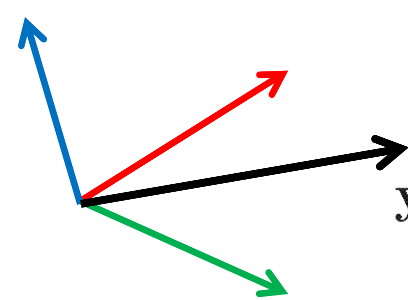
$$M = U\Lambda U^H, \qquad \Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n), \quad U^H U = I$$

U consists of n orthogonal eigenvectors,
and the $\lambda_j \in \mathbb{R}$ are eigenvalues.

Matrix A 10   1   1

$$\begin{bmatrix} 4 & 3 & 3 \\ 3 & 4 & 3 \\ 3 & 3 & 4 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

consider $\mathbf{y} = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3$



Review: Subspace Iteration for HEP

Recall the standard fact about Hermitian matrices.

For any Hermitian matrix M, n-by-n,
M can be diagonalized by a unitary matrix U:

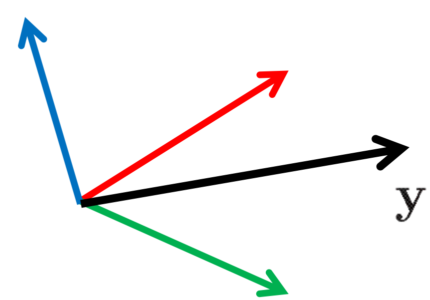
$$M = U \Lambda U^H, \qquad \Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n), \quad U^H U = I$$

U consists of n orthogonal eigenvectors,
and the $\lambda_j \in \mathbb{R}$ are eigenvalues.

$$\text{Matrix } A \quad \begin{bmatrix} 4 & 3 & 3 \\ 3 & 4 & 3 \\ 3 & 3 & 4 \end{bmatrix} \qquad \overset{10}{\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}} \quad \overset{1}{\begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix}} \quad \overset{1}{\begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}}$$

$$\text{consider } \mathbf{y} = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3$$

$$A\mathbf{y} = 10\,\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3$$



Review: Subspace Iteration for HEP

Recall the standard fact about Hermitian matrices.

For any Hermitian matrix M, n-by-n,
M can be diagonalized by a unitary matrix U:

$$M = U\Lambda U^H, \qquad \Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_n), \quad U^H U = I$$

U consists of n orthogonal eigenvectors,
and the $\lambda_j \in \mathbb{R}$ are eigenvalues.

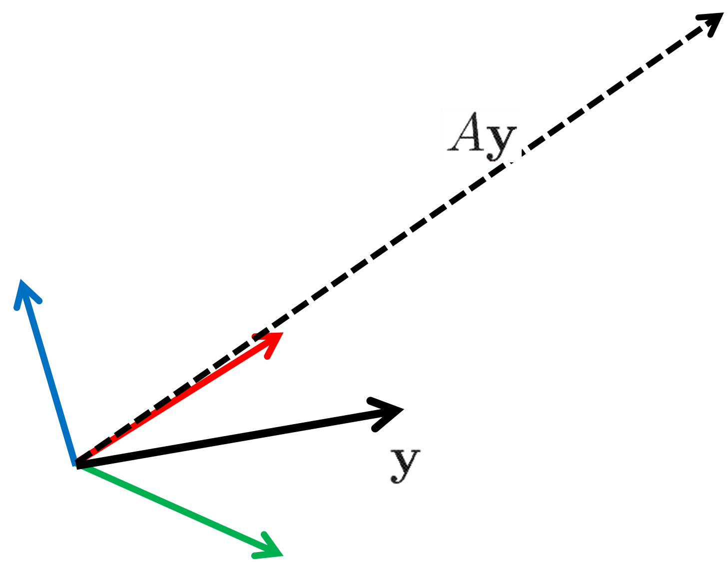
Matrix A 10   1   1

$$\begin{bmatrix} 4 & 3 & 3 \\ 3 & 4 & 3 \\ 3 & 3 & 4 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 1 \\ -1 \\ 1 \end{bmatrix} \quad \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix}$$

consider $\mathbf{y} = \mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3$

$A\mathbf{y} = 10\,\mathbf{x}_1 + \mathbf{x}_2 + \mathbf{x}_3$

$A\mathbf{y}$ is close to direction of $\mathbf{x}_1$



Review: Subspace Iteration for HEP

Matrix M, n-by-n

eigenvalues λ, eigenvectors $\mathbf{x}$

$M\mathbf{x} = \lambda\mathbf{x}$

Subspace Iteration:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$

Loop $k = 1, 2, 3 \ldots$

$Y_k \leftarrow \ M \ Q_{k-1}$

$Q_k \leftarrow \text{orthonormalize}(Y_k)$

$k \leftarrow k + 1$

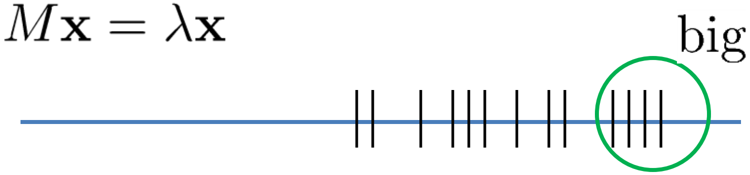
End Loop:

Review: Subspace Iteration for HEP

Matrix M, n-by-n

eigenvalues λ, eigenvectors $\mathbf{x}$

$M\mathbf{x} = \lambda\mathbf{x}$



Subspace Iteration:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$

Loop $k = 1, 2, 3 \ldots$

$Y_k \leftarrow M \; Q_{k-1}$

$Q_k \leftarrow \text{orthonormalize}(Y_k)$

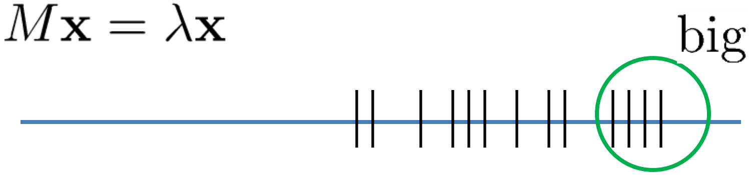
$k \leftarrow k + 1$

End Loop:

$Q_k^T M Q_k$ captures the "big"

Review: Subspace Iteration for HEP

Matrix M, n-by-n
eigenvalues λ, eigenvectors $\mathbf{x}$
$M\mathbf{x} = \lambda\mathbf{x}$



Informally:
If $|\lambda_i| > |\lambda_{p+1}|$
then distance$(\mathbf{x}_i, \mathcal{Q}_k) \to 0$
$\mathcal{Q}_k$ = subspace spanned by Q_k

Subspace Iteration:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$

Loop $k = 1, 2, 3 \ldots$

$Y_k \leftarrow M \; Q_{k-1}$

$Q_k \leftarrow \text{orthonormalize}(Y_k)$

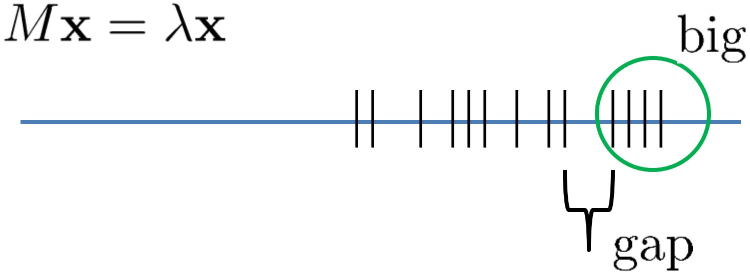
$k \leftarrow k + 1$

End Loop:

$Q_k^T M Q_k$ captures the "big"

Review: Subspace Iteration for HEP

Matrix M, n-by-n
eigenvalues λ, eigenvectors $\mathbf{x}$
$M\mathbf{x} = \lambda\mathbf{x}$



Subspace Iteration:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$
Loop $k = 1, 2, 3 \ldots$

$Y_k \leftarrow M \; Q_{k-1}$

$Q_k \leftarrow \text{orthonormalize}(Y_k)$

$k \leftarrow k+1$

End Loop:
$Q_k^T M Q_k$ captures the "big"

Informally:
If $|\lambda_i| > |\lambda_{p+1}|$
then distance$(\mathbf{x}_i, \mathcal{Q}_k) \to 0$
$\mathcal{Q}_k$ = subspace spanned by Q_k

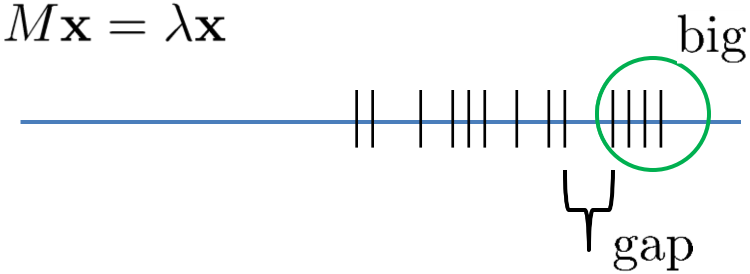
Formally:
If $|\lambda_i| > |\lambda_{p+1}|$
$\mathcal{Q}_k$ = subspace spanned by Q_k
P_k is orthogonal projection to $\mathcal{Q}_k$,
i.e. $P_k = Q_k Q_k^H$
$\|(I - P_k)\mathbf{x}_i\|_2 \le \alpha |\lambda_{p+1}/\lambda_i|^k$

Review: Subspace Iteration for HEP

Matrix M, n-by-n

eigenvalues λ, eigenvectors $\mathbf{x}$

$M\mathbf{x} = \lambda \mathbf{x}$



Subspace Iteration:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$

Loop $k = 1, 2, 3 \ldots$

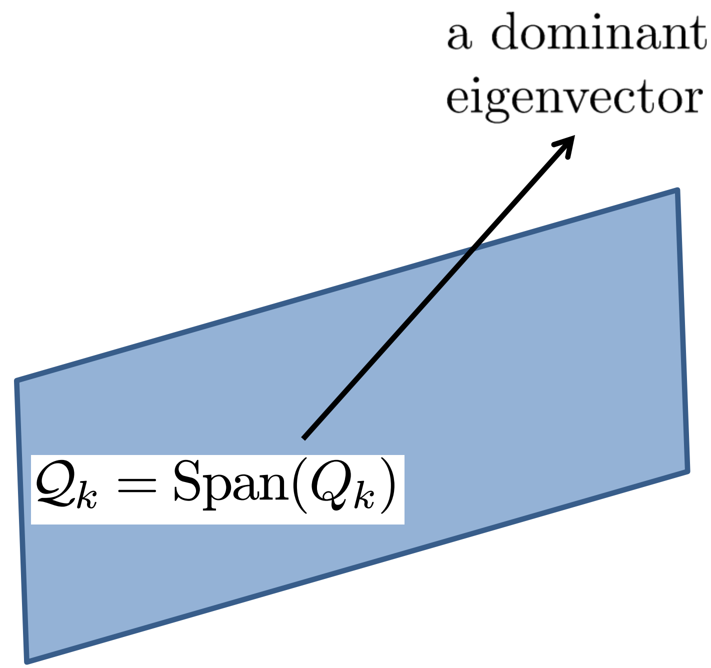
$Y_k \leftarrow M \; Q_{k-1}$

$Q_k \leftarrow \text{orthonormalize}(Y_k)$

$k \leftarrow k + 1$

End Loop:

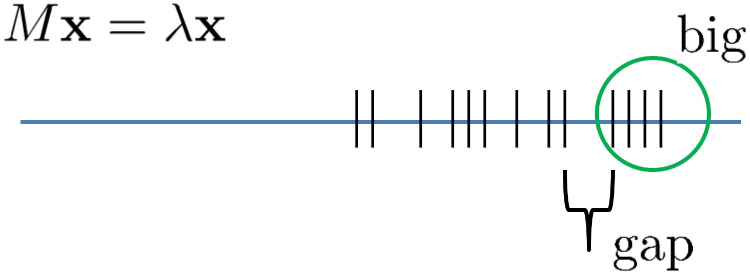
$Q_k^T M Q_k$ captures the "big"



Review: Subspace Iteration for HEP

Matrix M, n-by-n
eigenvalues λ, eigenvectors $\mathbf{x}$

$M\mathbf{x} = \lambda\mathbf{x}$



Subspace Iteration:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$

Loop $k = 1, 2, 3 \ldots$

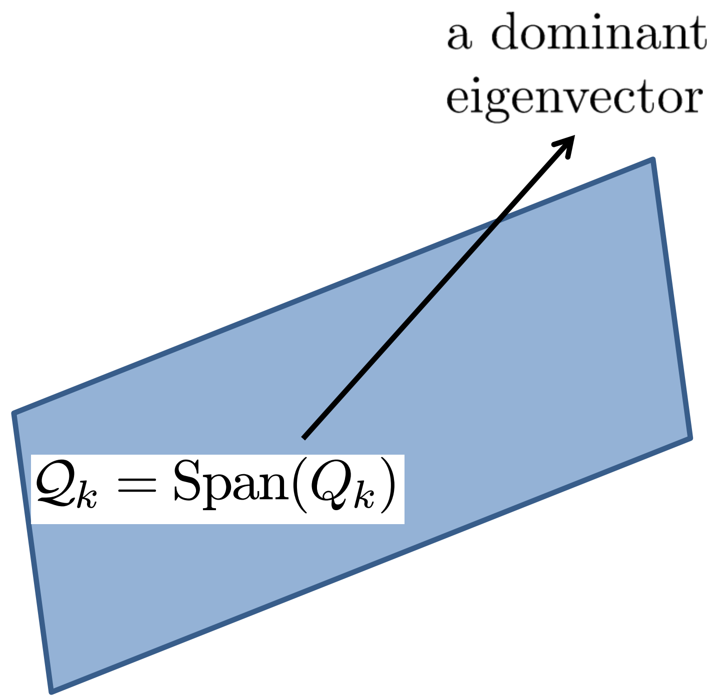
$Y_k \leftarrow M \; Q_{k-1}$

$Q_k \leftarrow \text{orthonormalize}(Y_k)$

$k \leftarrow k + 1$

End Loop:

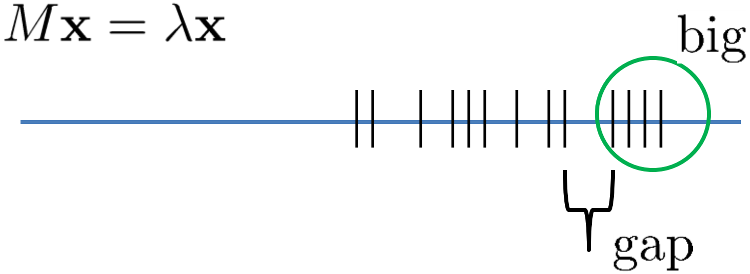
$Q_k^T M Q_k$ captures the "big"



Review: Subspace Iteration for HEP

Matrix M, n-by-n
eigenvalues λ, eigenvectors $\mathbf{x}$

$M\mathbf{x} = \lambda\mathbf{x}$



Subspace Iteration:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$

Loop $k = 1, 2, 3 \ldots$

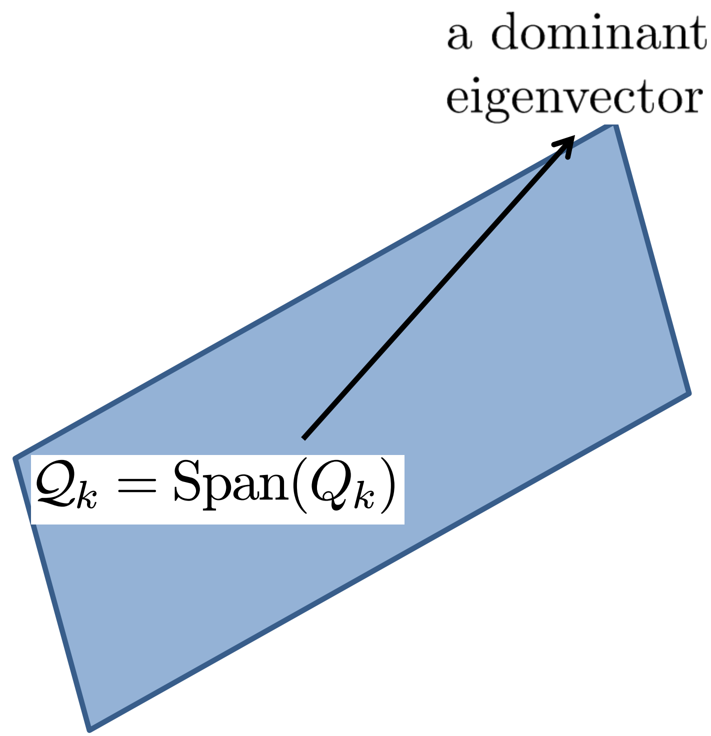
$Y_k \leftarrow M \; Q_{k-1}$

$Q_k \leftarrow \text{orthonormalize}(Y_k)$

$k \leftarrow k + 1$

End Loop:

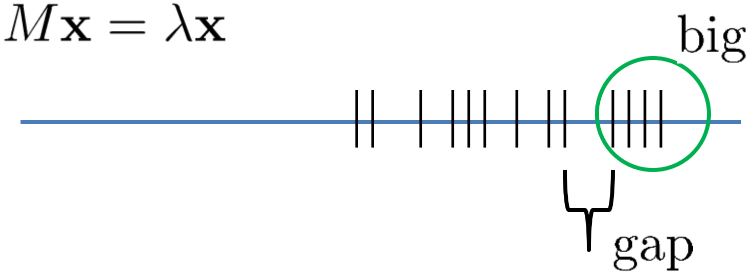
$Q_k^T M Q_k$ captures the "big"



Review: Subspace Iteration for HEP

Matrix M, n-by-n
eigenvalues λ, eigenvectors $\mathbf{x}$

$M\mathbf{x} = \lambda\mathbf{x}$



Subspace Iteration:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$

Loop $k = 1, 2, 3 \ldots$

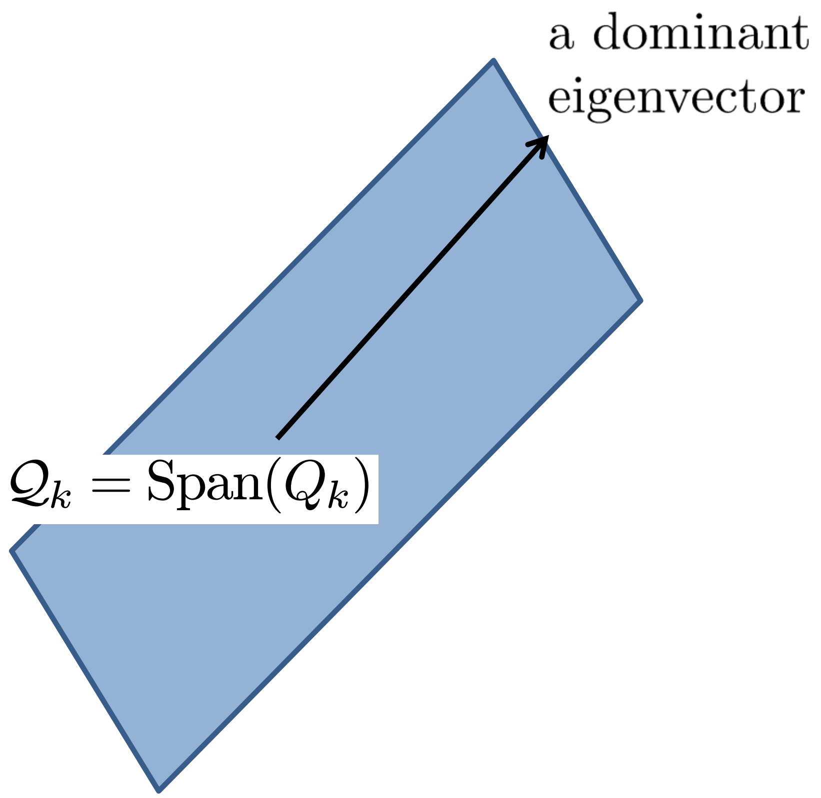
$Y_k \leftarrow M\ Q_{k-1}$

$Q_k \leftarrow \text{orthonormalize}(Y_k)$

$k \leftarrow k + 1$

End Loop:

$Q_k^T M Q_k$ captures the "big"

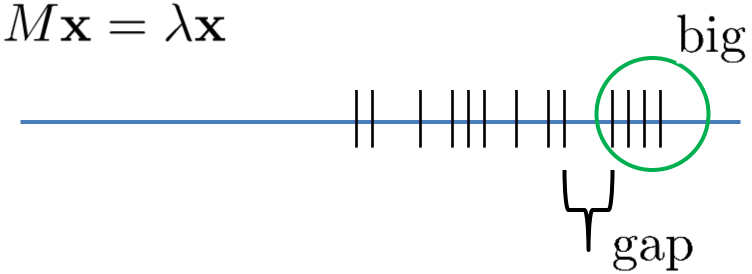


Review: Subspace Iteration for HEP

Matrix M, n-by-n

eigenvalues λ, eigenvectors $\mathbf{x}$

$M\mathbf{x} = \lambda\mathbf{x}$



Subspace Iteration:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$

Loop $k = 1, 2, 3 \ldots$

$Y_k \leftarrow \; M \; Q_{k-1}$

$Q_k \leftarrow \text{orthonormalize}(Y_k)$

$k \leftarrow k + 1$

End Loop:

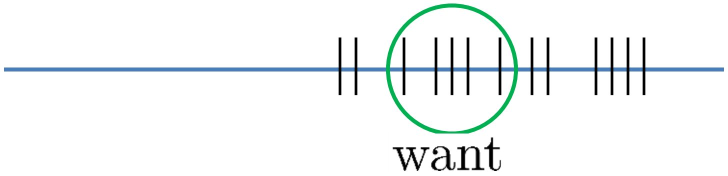
$Q_k^T M Q_k$ captures the "big"

- Only works for "big"
- Speed at the mercy of "gap"
- Can be VERY slow

Review: Subspace Iteration for HEP

Matrix M, n-by-n
eigenvalues λ, eigenvectors $\mathbf{x}$

$M\mathbf{x} = \lambda\mathbf{x}$



Subspace Iteration:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$

Loop $k = 1, 2, 3 \ldots$

$Y_k \leftarrow M \; Q_{k-1}$

$Q_k \leftarrow \text{orthonormalize}(Y_k)$

$k \leftarrow k + 1$

End Loop:

$Q_k^T M Q_k$ captures the "big"

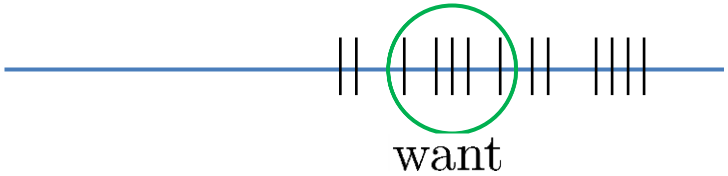
CANNOT give what you WANT

FEAST as subspace Iteration

Matrix M, n-by-n

eigenvalues λ, eigenvectors $\mathbf{x}$

$M\mathbf{x} = \lambda\mathbf{x}$



Subspace Iteration:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$

Loop $k = 1, 2, 3 \ldots$

$Y_k \leftarrow \ M \ Q_{k-1}$

$Q_k \leftarrow \text{orthonormalize}(Y_k)$

$k \leftarrow k + 1$

End Loop:

$Q_k^T M Q_k$ captures the "big"

CANNOT give what you WANT

FEAST:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$

Loop $k = 1, 2, 3 \ldots$

$Y_k \leftarrow \rho_{\text{quad}}(M) \ Q_{k-1}$

$Q_k \leftarrow \text{orthonormalize}(Y_k)$

$k \leftarrow k + 1$

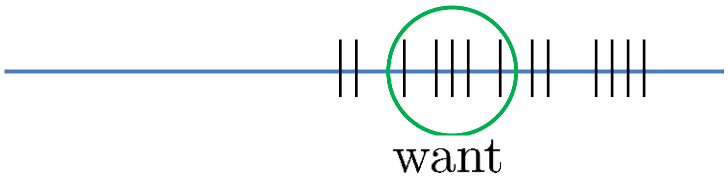
End Loop:

FEAST as subspace Iteration

Matrix M, n-by-n

eigenvalues λ, eigenvectors $\mathbf{x}$

$M\mathbf{x} = \lambda\mathbf{x}$



Subspace Iteration:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$

Loop $k = 1, 2, 3 \ldots$

$Y_k \leftarrow M \; Q_{k-1}$

$Q_k \leftarrow \text{orthonormalize}(Y_k)$

$k \leftarrow k + 1$

End Loop:

$Q_k^T M Q_k$ captures the "big"

CANNOT give what you WANT

FEAST:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$

Loop $k = 1, 2, 3 \ldots$

heart of FEAST $Y_k \leftarrow \rho_{\text{quad}}(M) \; Q_{k-1}$

$Q_k \leftarrow \text{orthonormalize}(Y_k)$

$k \leftarrow k + 1$

End Loop:

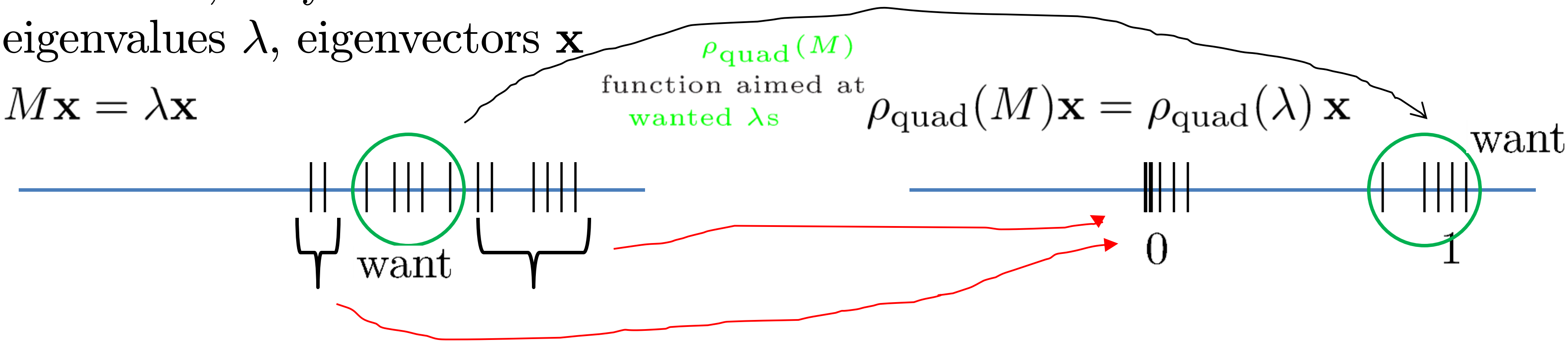
$Q_k^T M Q_k$ captures WANT

FEAST as subspace Iteration

Matrix M, n-by-n

eigenvalues λ, eigenvectors $\mathbf{x}$

$M\mathbf{x} = \lambda\mathbf{x}$



Subspace Iteration:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$

Loop $k = 1, 2, 3 \ldots$

$Y_k \leftarrow \; M \; Q_{k-1}$

$Q_k \leftarrow \text{orthonormalize}(Y_k)$

$k \leftarrow k + 1$

End Loop:

$Q_k^T M Q_k$ captures the "big"

CANNOT give what you WANT

FEAST:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$

Loop $k = 1, 2, 3 \ldots$

$Y_k \leftarrow \rho_{\text{quad}}(M) \; Q_{k-1}$

$Q_k \leftarrow \text{orthonormalize}(Y_k)$

$k \leftarrow k + 1$

End Loop:

$Q_k^T M Q_k$ captures WANT

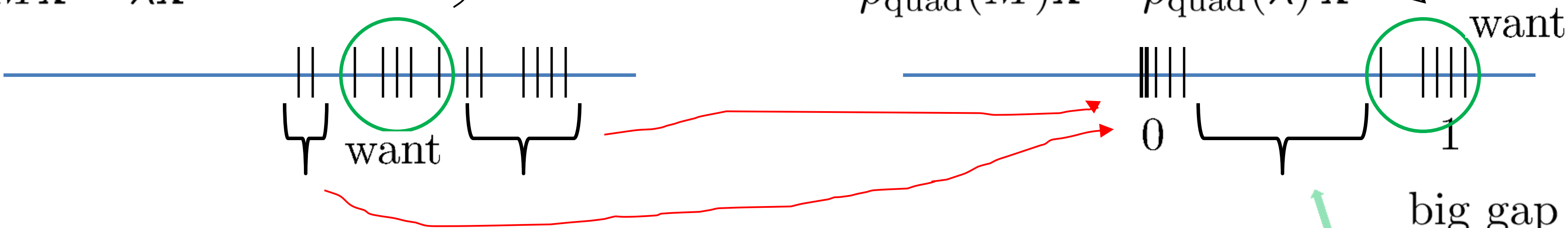
FEAST as subspace Iteration

Matrix M, n-by-n

eigenvalues λ, eigenvectors $\mathbf{x}$

$M\mathbf{x} = \lambda\mathbf{x}$

$\rho_{\text{quad}}(M)\mathbf{x} = \rho_{\text{quad}}(\lambda)\,\mathbf{x}$



Subspace Iteration:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$

Loop $k = 1, 2, 3 \ldots$

$Y_k \leftarrow \;M\; Q_{k-1}$

$Q_k \leftarrow \text{orthonormalize}(Y_k)$

$k \leftarrow k+1$

End Loop:

$Q_k^T M Q_k$ captures the "big"

CANNOT give what you WANT

FEAST:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$

Loop $k = 1, 2, 3 \ldots$

$Y_k \leftarrow \rho_{\text{quad}}(M)\; Q_{k-1}$

$Q_k \leftarrow \text{orthonormalize}(Y_k)$

$k \leftarrow k+1$

End Loop:

$Q_k^T M Q_k$ captures WANT, **FAST!**

FEAST for Hermitian Problems

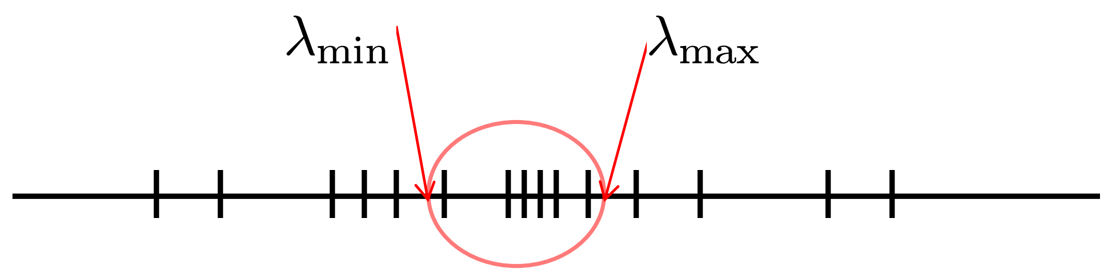
- FEAST as filtered subspace iteration for standard Hermitian eigenproblem
- The approximate spectral projector filter
- Convergence analysis of filtered subspace iteration
- FEAST for generalized Hermitian problem

Spectral Projector and $\rho_{\text{quad}}(M)$

$M = X\,\Lambda\,X^{-1}$, X is orthonormal

- we are interested in λ inside $[\lambda_{\min}, \lambda_{\max}]$

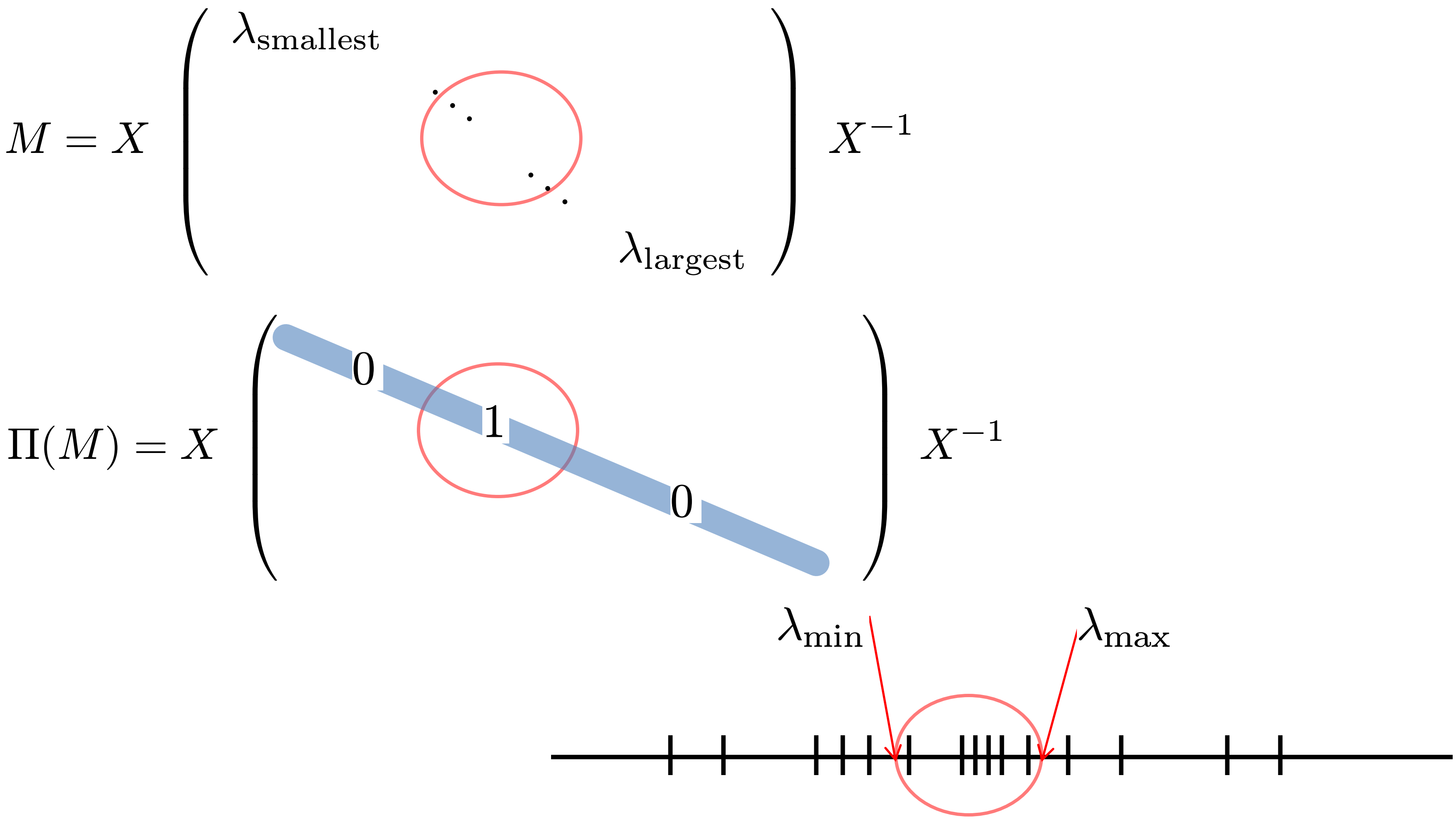
$$M = X \begin{pmatrix} \lambda_{\text{smallest}} & & & & \\ & \ddots & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & \lambda_{\text{largest}} \end{pmatrix} X^{-1}$$



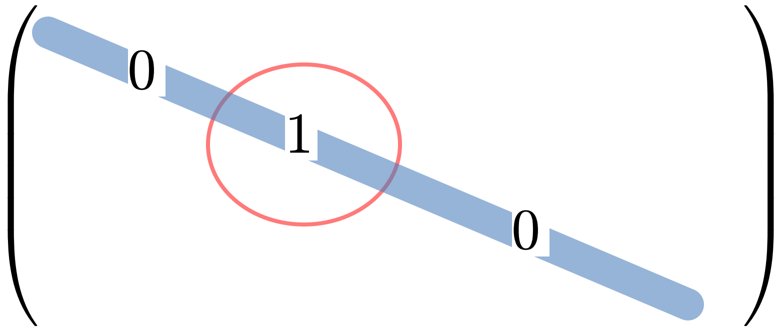
Spectral Projector and $\rho_{\text{quad}}(M)$

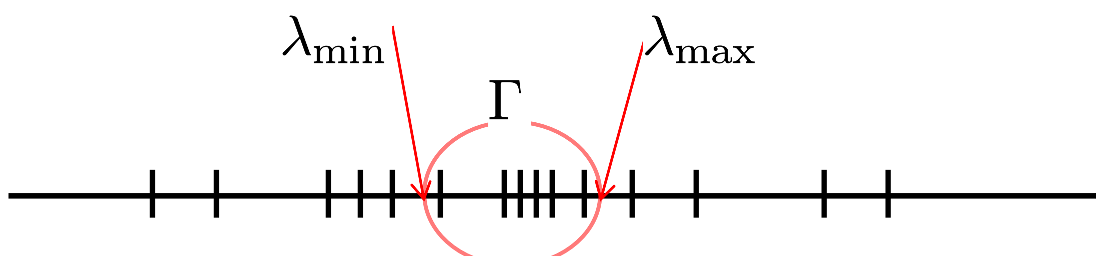
$M = X\,\Lambda\,X^{-1}$, X is orthonormal

- we are interested in λ inside $[\lambda_{\min}, \lambda_{\max}]$



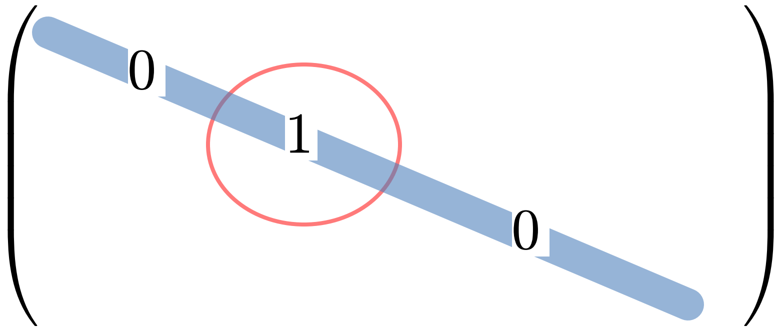
$M = X\,\Lambda\,X^{-1}$, X is orthonormal

$\Pi(M) = X$  X^{-1}



$M = X\,\Lambda\,X^{-1}$, X is orthonormal

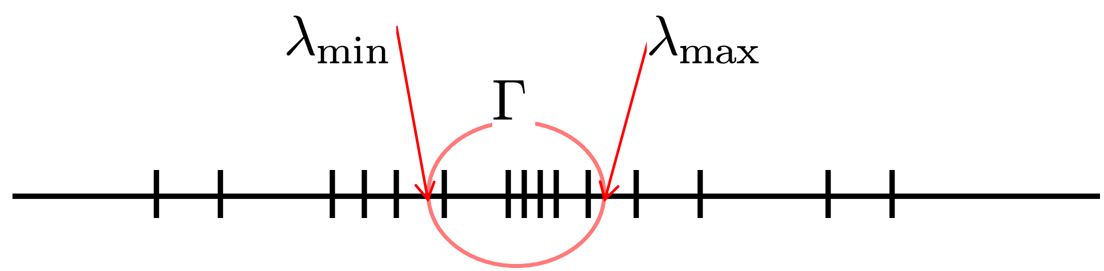
$\Pi(M) = X$



X^{-1}

Want scalar function $\rho(\lambda)$ where

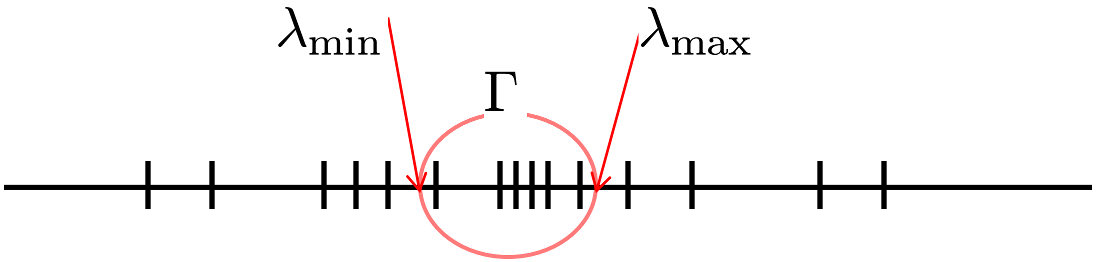
- $\rho(\lambda)$ is close to 1 for λ inside interval
- $\rho(\lambda)$ is close to 0
- The matrix function $\rho(M)Y$ is practical to compute



$M = X\,\Lambda\,X^{-1}$, X is orthonormal

Want scalar function $\rho(\lambda)$ where

- $\rho(\lambda)$ is close to 1 for λ inside interval
- $\rho(\lambda)$ is close to 0
- The matrix function $\rho(M)Y$ is practical to compute



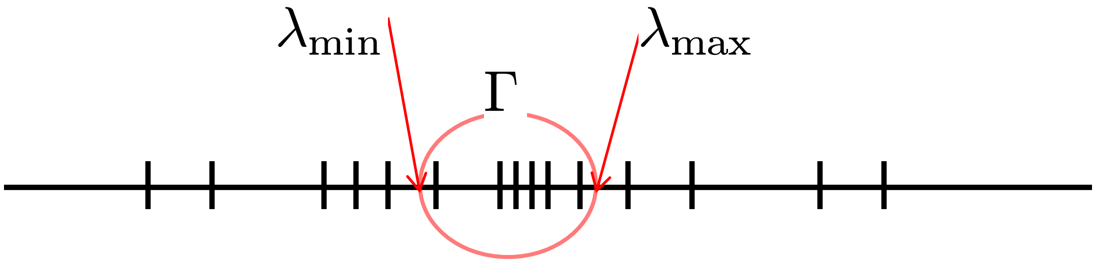
$M = X \, \Lambda \, X^{-1}$, X is orthonormal

Want scalar function $\rho(\lambda)$ where

- $\rho(\lambda)$ is close to 1 for λ inside interval
- $\rho(\lambda)$ is close to 0
- The matrix function $\rho(M)Y$ is practical to compute

Consider the scalar function $\pi(\lambda)$ defined as

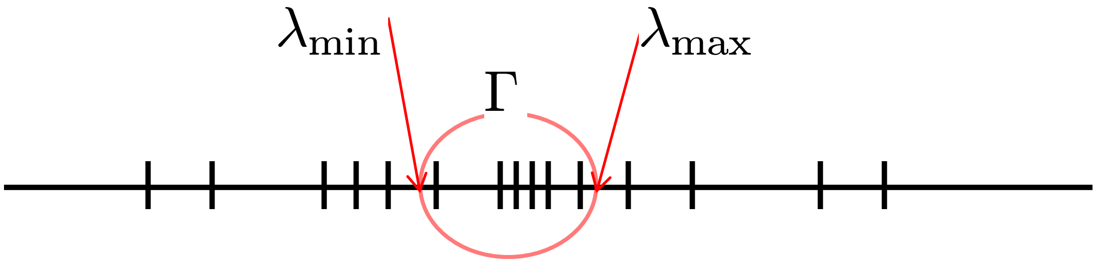
$$\pi(\lambda) = \frac{1}{2\pi\iota} \oint_{\Gamma} (z-\lambda)^{-1} \, dz = \begin{cases} 1 & \lambda \in \Gamma\text{'s interior,} \\ 0 & \text{otherwise.} \end{cases}$$



$M = X\,\Lambda\,X^{-1}$, X is orthonormal

Consider the scalar function $\pi(\lambda)$ defined as

$$\pi(\lambda) = \frac{1}{2\pi\iota}\oint_{\Gamma} (z-\lambda)^{-1}\,dz = \begin{cases} 1 & \lambda \in \Gamma\text{'s interior,} \\ 0 & \text{otherwise.} \end{cases}$$



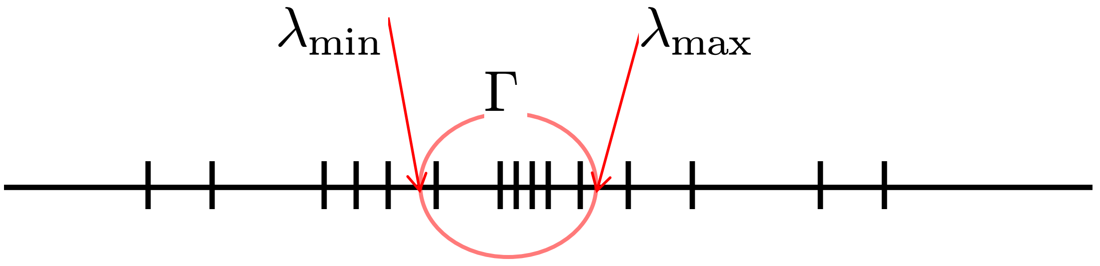
$M = X\,\Lambda\,X^{-1}$, X is orthonormal

Consider the scalar function $\pi(\lambda)$ defined as

$$\pi(\lambda) = \frac{1}{2\pi\iota}\oint_{\Gamma}(z-\lambda)^{-1}\,dz = \begin{cases} 1 & \lambda \in \Gamma\text{'s interior,} \\ 0 & \text{otherwise.} \end{cases}$$

Therefore define $\rho_{\text{quad}}(\lambda)$ as a quadrature rule that approximates $\pi(\lambda)$:

$$\rho_{\text{quad}}(\lambda) \stackrel{\text{def}}{=} \sum_{k=1}^{q} w_k (z_k - \lambda)^{-1}$$



$M = X\,\Lambda\,X^{-1}$, X is orthonormal

Consider the scalar function $\pi(\lambda)$ defined as

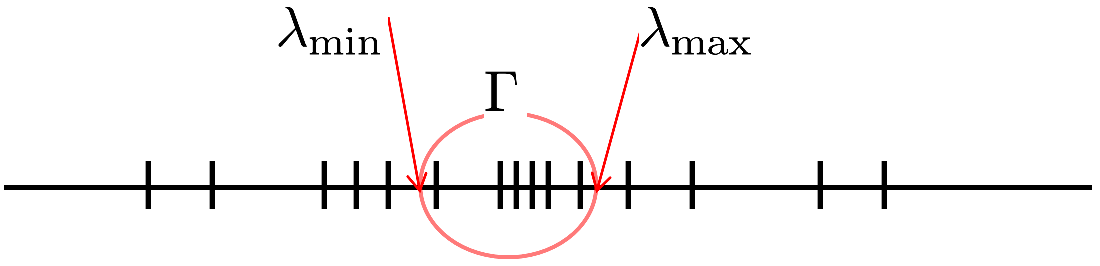
$$\pi(\lambda) = \frac{1}{2\pi\iota}\oint_\Gamma (z-\lambda)^{-1}\,dz = \begin{cases} 1 & \lambda \in \Gamma\text{'s interior,} \\ 0 & \text{otherwise.} \end{cases}$$

Therefore define $\rho_{\text{quad}}(\lambda)$ as a quadrature rule that approximates $\pi(\lambda)$:

$$\rho_{\text{quad}}(\lambda) \stackrel{\text{def}}{=} \sum_{k=1}^{q} w_k (z_k - \lambda)^{-1}$$

Easy to see

$$\rho_{\text{quad}}(M)Y = \sum_{k=1}^{q} w_k (z_k I - M)^{-1} Y$$

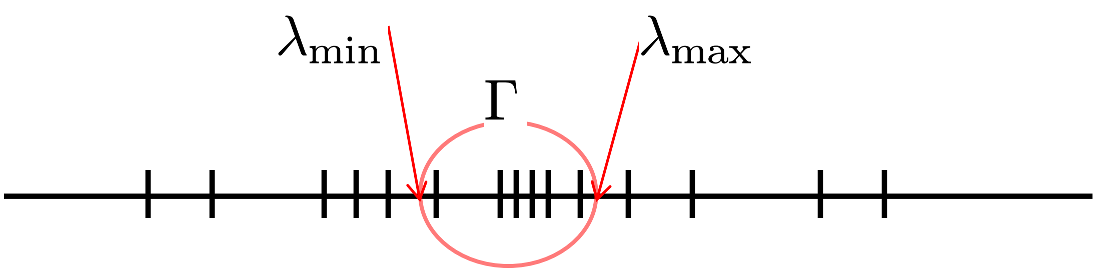


Approximate Spectral Projector

$M = X\,\Lambda\,X^{-1}$, X is orthonormal

$$\rho_{\text{quad}}(M) = X \rho_{\text{quad}}(\Lambda) X^{-1} \qquad \rho_{\text{quad}}(\lambda) = \sum_{k=1}^{q} w_k (z_k - \lambda)^{-1}$$

- $\rho_{\text{quad}}(\lambda)$ is close to 1 for λ inside interval
- $\rho_{\text{quad}}(\lambda)$ is close to 0
- Once quadrature rule is fixed, we know ρ_{quad} exactly

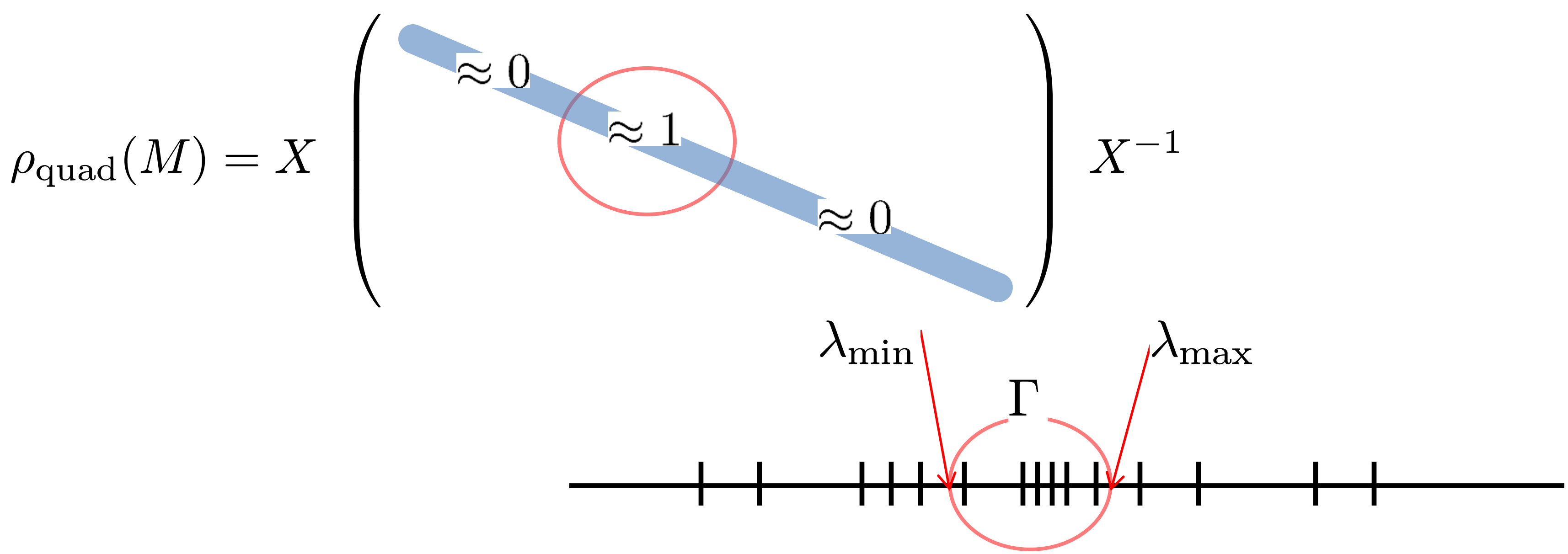


Approximate Spectral Projector

$M = X\,\Lambda\,X^{-1}$, X is orthonormal

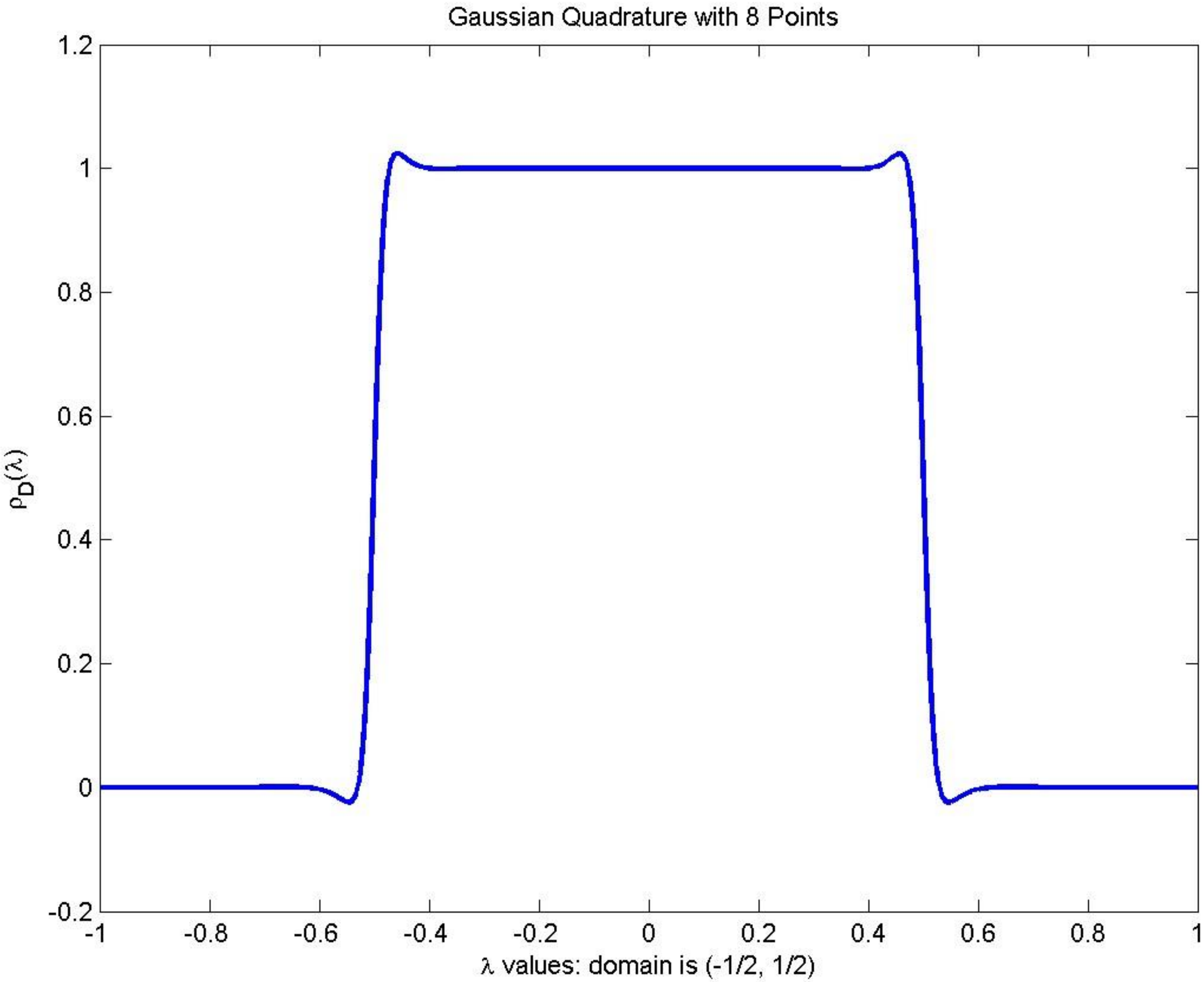
$$\rho_{\text{quad}}(M) = X\rho_{\text{quad}}(\Lambda)X^{-1} \qquad \rho_{\text{quad}}(\lambda) = \sum_{k=1}^{q} w_k(z_k - \lambda)^{-1}$$

- $\rho_{\text{quad}}(\lambda)$ is close to 1 for λ inside interval
- $\rho_{\text{quad}}(\lambda)$ is close to 0
- Once quadrature rule is fixed, we know ρ_{quad} exactly



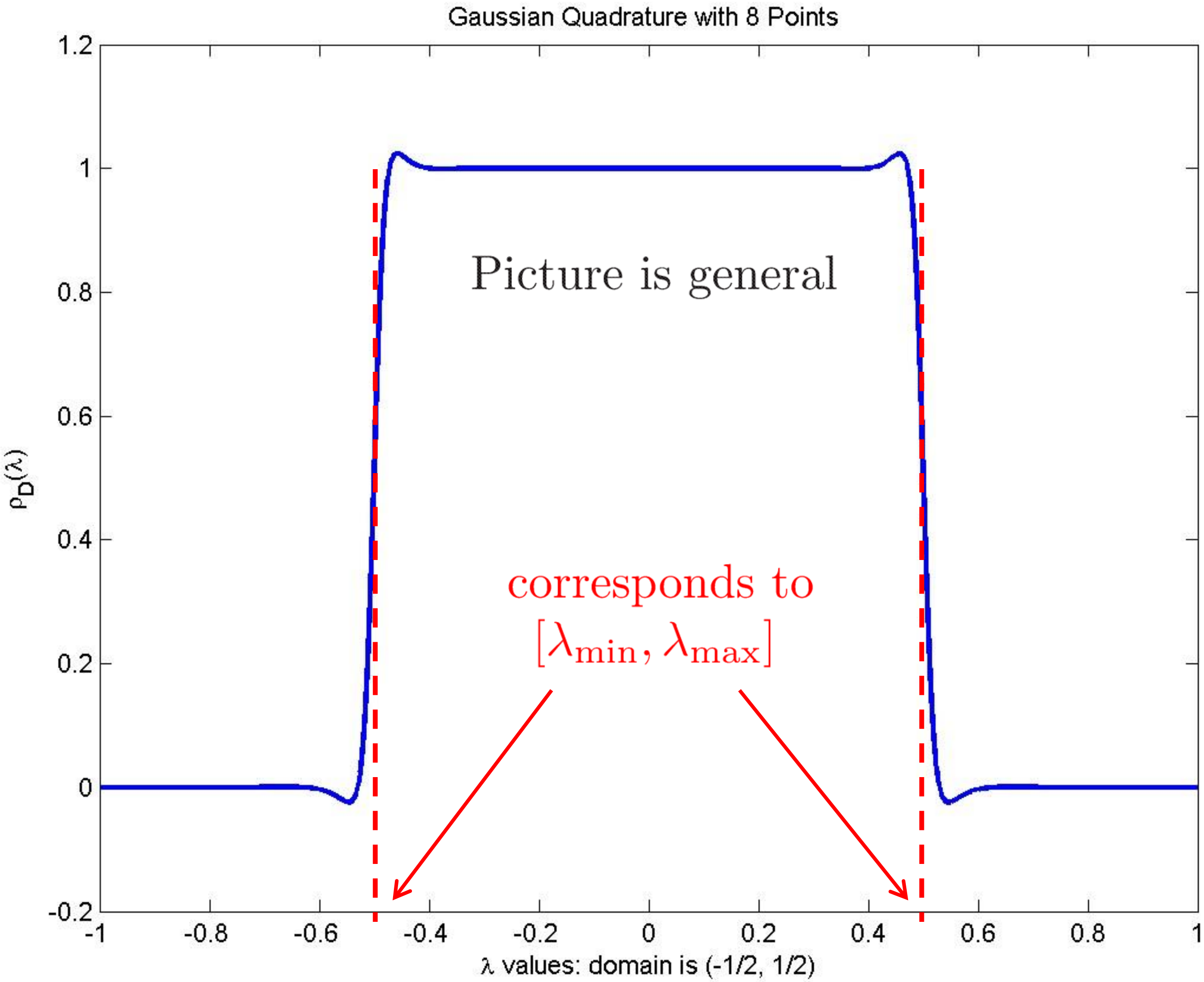
Approximate Spectral Projector

$$\rho_{\text{quad}}(\lambda) = \sum_{k=1}^{q} w_k (z_k - \lambda)^{-1}$$



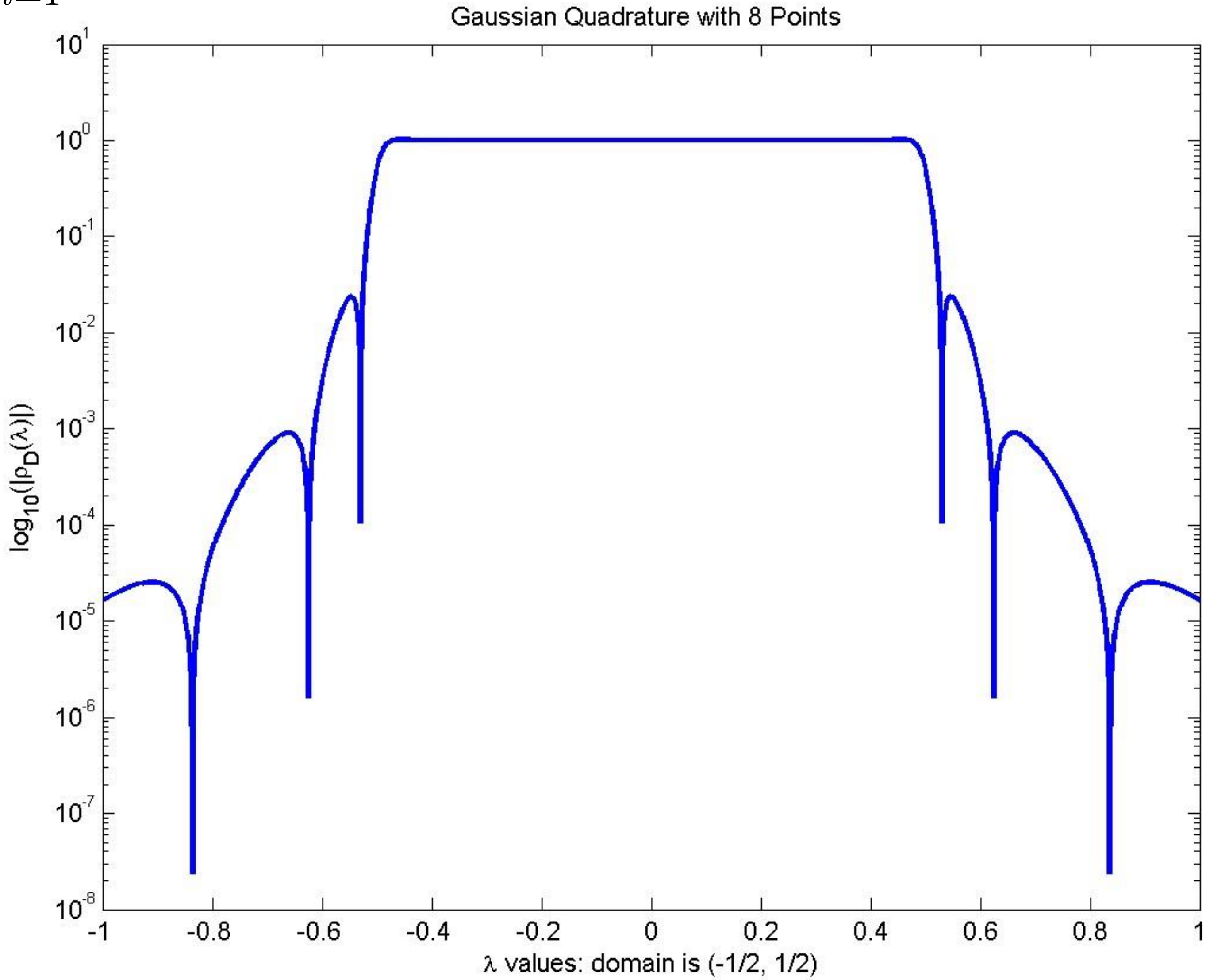
Approximate Spectral Projector

$$\rho_{\text{quad}}(\lambda) = \sum_{k=1}^{q} w_k (z_k - \lambda)^{-1}$$

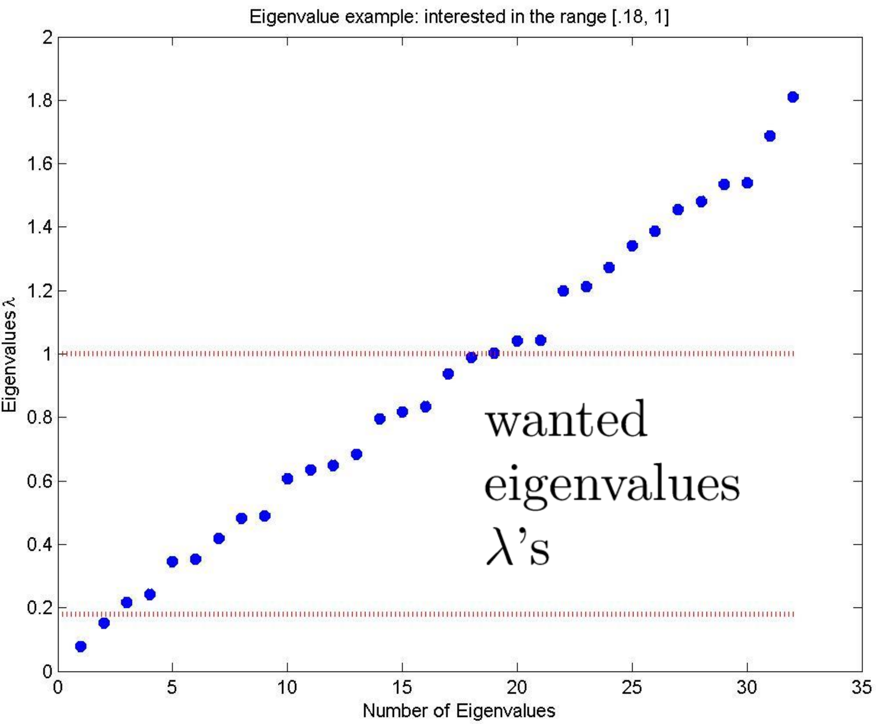


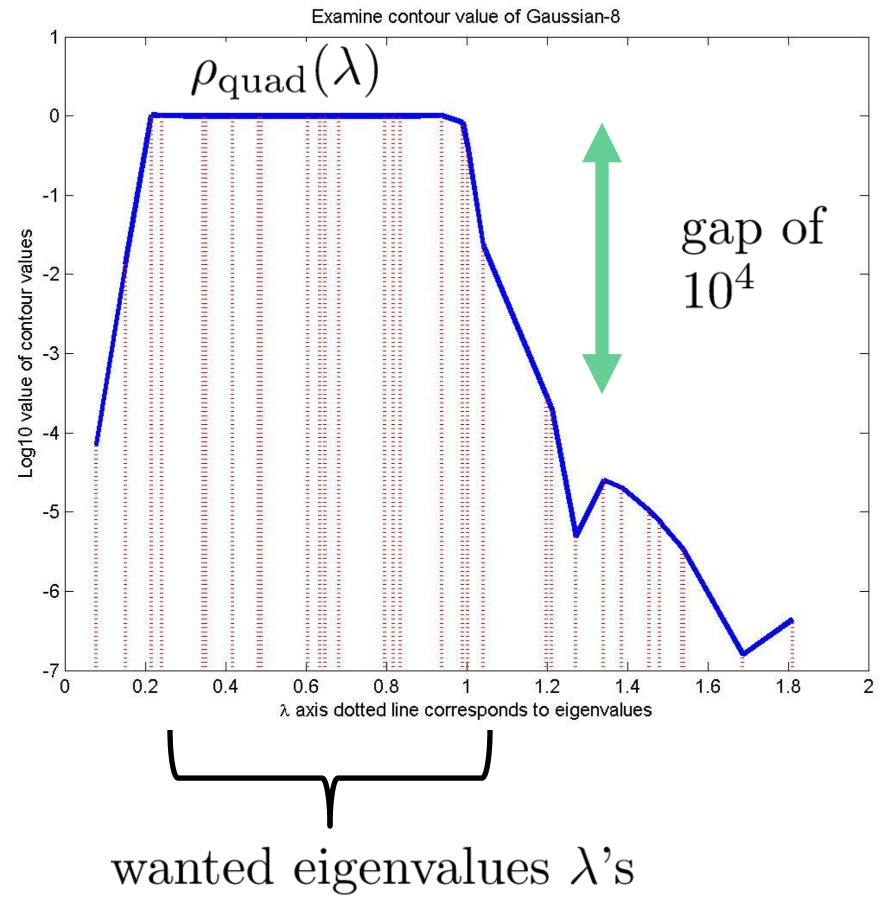
Approximate Spectral Projector

$$\rho_{\text{quad}}(\lambda) = \sum_{k=1}^{q} w_k (z_k - \lambda)^{-1}$$



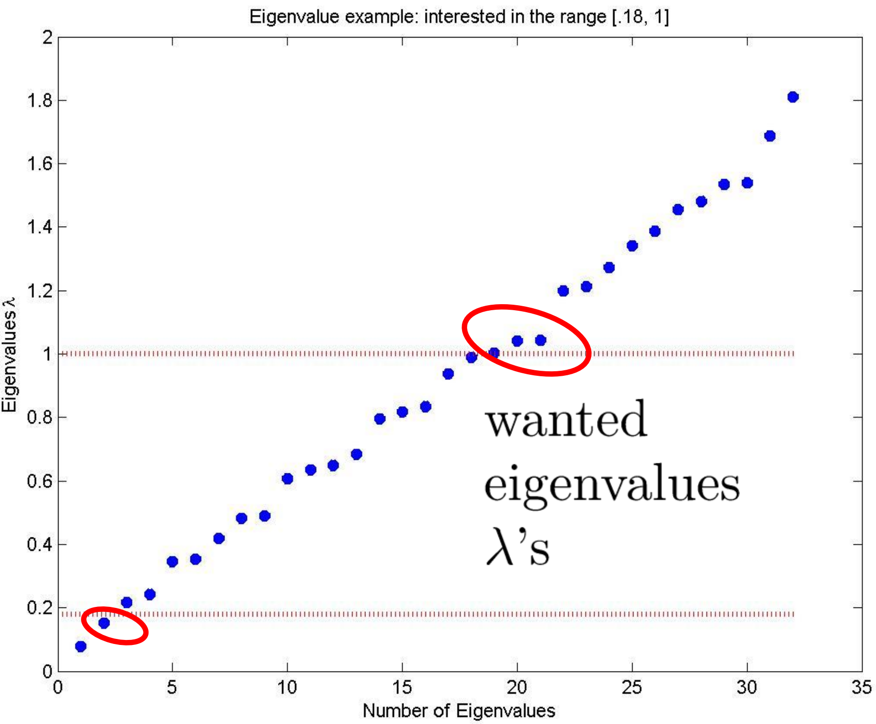
- 16 eigenvalues in search interval

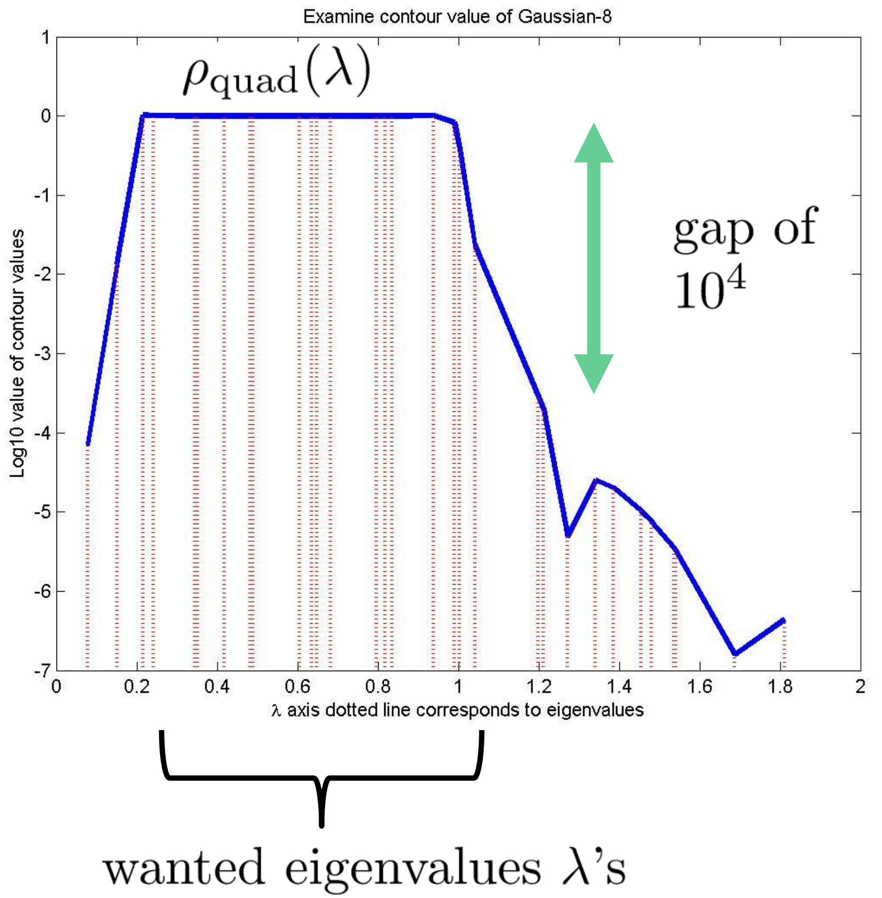




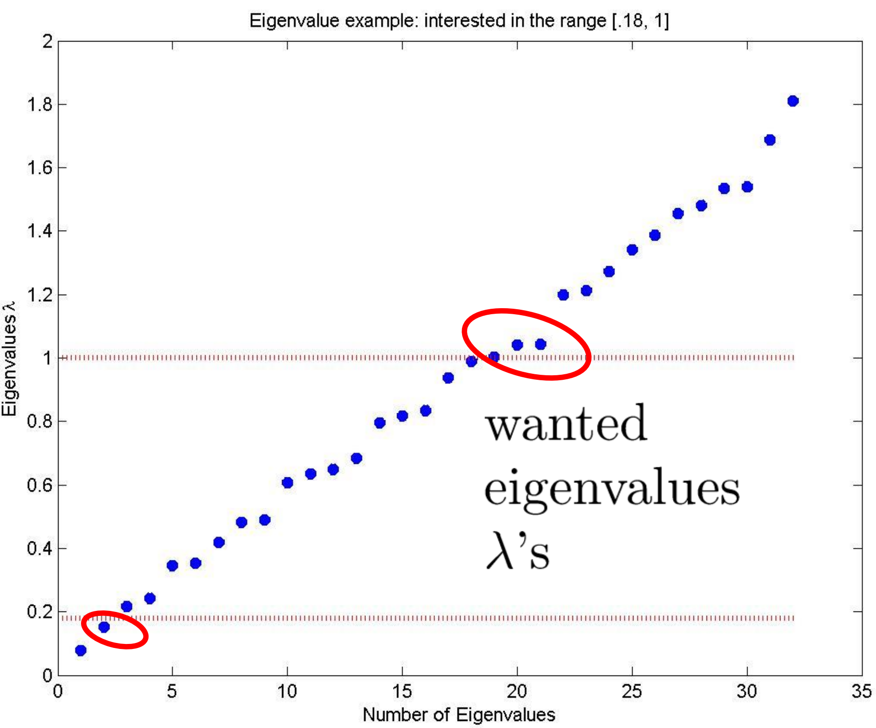
Example from an actual application

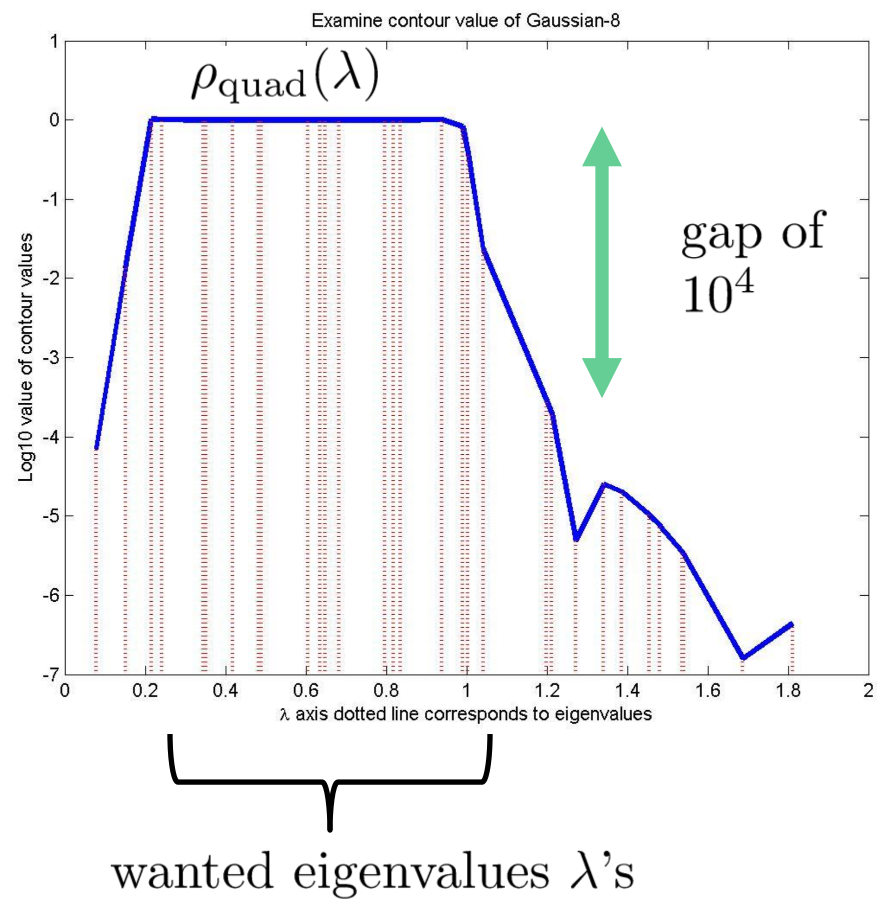
- 16 eigenvalues in search interval
- note $\rho(\lambda_{20}) \approx 1$ and $|\rho(\lambda_{21})| \approx 10^{-4}$
- As long as p is chosen ≥ 20



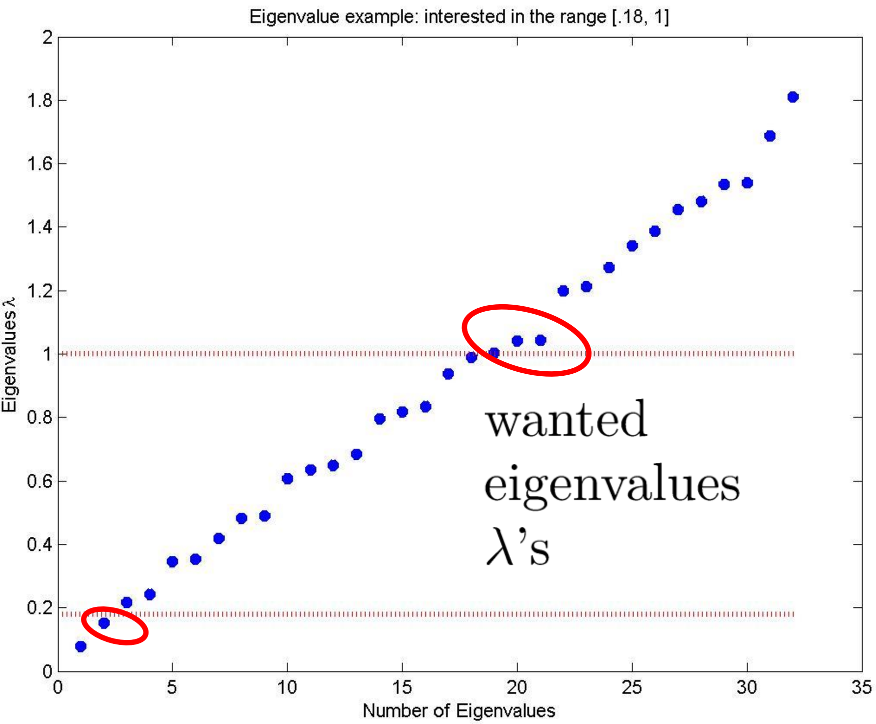


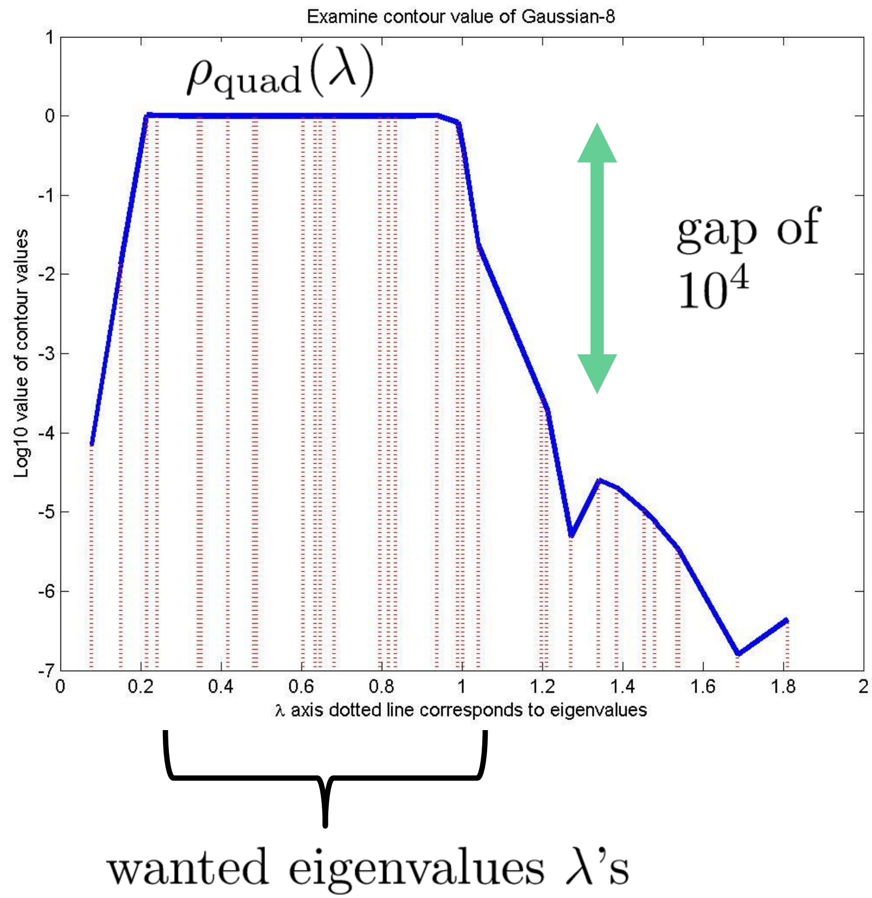
- 16 eigenvalues in search interval
- note $\rho(\lambda_{20}) \approx 1$ and $|\rho(\lambda_{21})| \approx 10^{-4}$
- As long as p is chosen ≥ 20
- covergence rate $\approx$ 4 digits/iteration
- Also, we will capture more then the target space



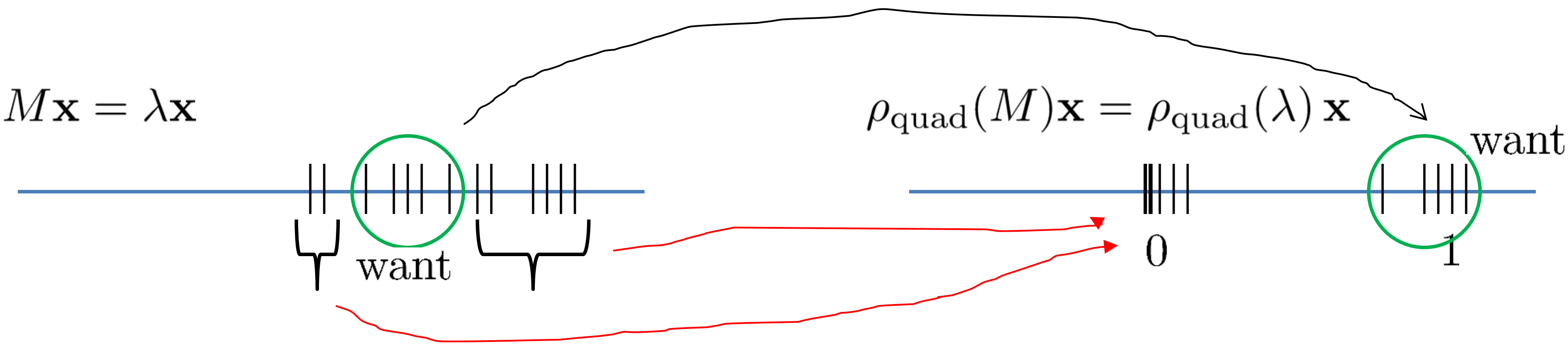


- If $p < 20$, $|\nu_{p+1}/\nu_i| \approx 1$ for $i = 1, 2, \ldots, p$.
- Iterations will almost surely fail to converge!





FEAST as subspace Iteration



FEAST:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$

Loop $k = 1, 2, 3 \ldots$

$Y_k \leftarrow \rho_{\text{quad}}(M)\, Q_{k-1}$

$Q_k \leftarrow \text{orthonormalize}(Y_k)$

$k \leftarrow k + 1$

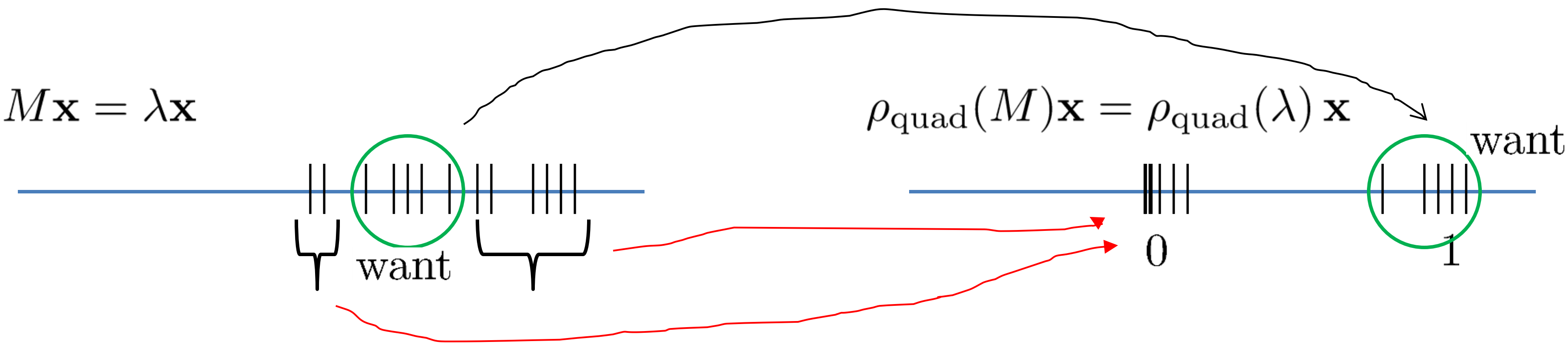
End Loop:

$Q_k^T M Q_k$ captures WANT, **FAST!**

FEAST as subspace Iteration

$M\mathbf{x} = \lambda\mathbf{x}$

$\rho_{\text{quad}}(M)\mathbf{x} = \rho_{\text{quad}}(\lambda)\,\mathbf{x}$



We know: $\mathcal{Q}_k$ captures the $\mathbf{x}_i$:
distance$(\mathbf{x}_i, \mathcal{Q}_k) \to 0$
for $i = 1, 2, \ldots, m$, $m_\lambda \le m \le p$

FEAST:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$

Loop $k = 1, 2, 3 \ldots$

$Y_k \leftarrow \rho_{\text{quad}}(M)\; Q_{k-1}$

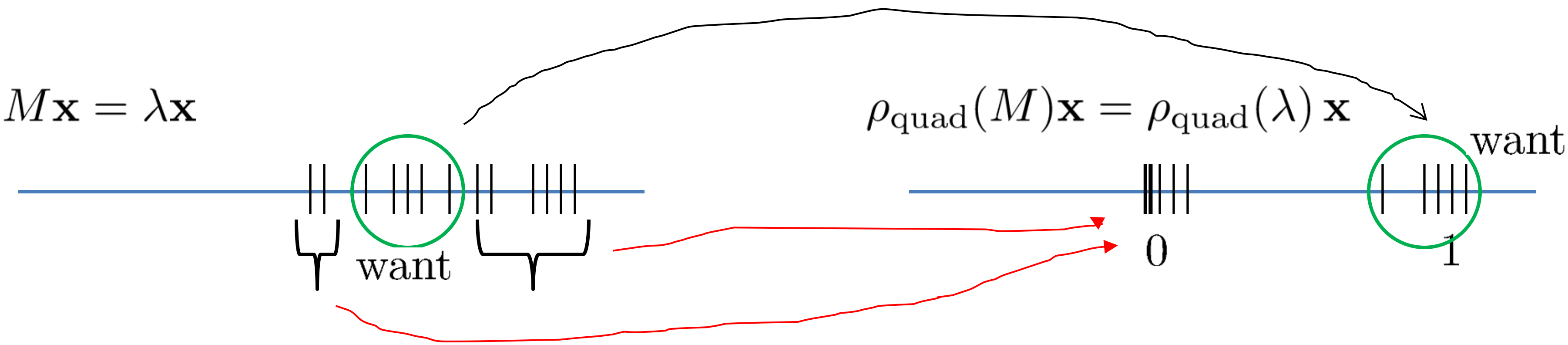
$Q_k \leftarrow \text{orthonormalize}(Y_k)$

$k \leftarrow k + 1$

End Loop:

$Q_k^T M Q_k$ captures WANT, **FAST!**

FEAST as subspace Iteration



We know: $\mathcal{Q}_k$ captures the $\mathbf{x}_i$:
distance$(\mathbf{x}_i, \mathcal{Q}_k) \to 0$
for $i = 1, 2, \ldots, m$, $m_\lambda \leq m \leq p$

m_λ is the
exact # of eigenvalues
in $[\lambda_{\min}, \lambda_{\max}]$

But still need to know how to actually obtain $\mathbf{x}_i$ and λ_i

FEAST:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$

Loop $k = 1, 2, 3 \ldots$

$Y_k \leftarrow \rho_{\text{quad}}(M)\, Q_{k-1}$

$Q_k \leftarrow \text{orthonormalize}(Y_k)$

$k \leftarrow k + 1$

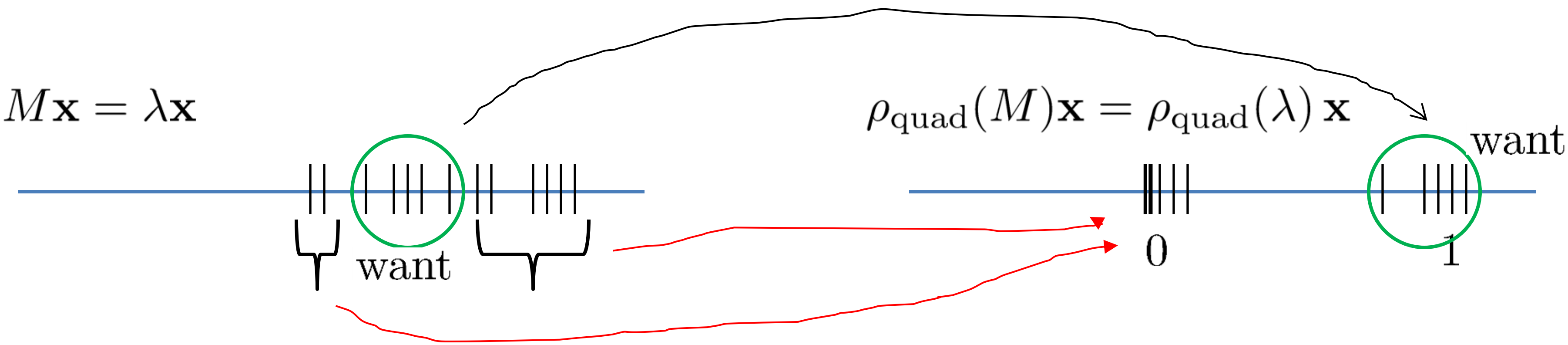
End Loop:

$Q_k^T M Q_k$ captures WANT, **FAST!**

FEAST for Hermitian Problems

- FEAST as filtered subspace iteration for standard Hermitian eigenproblem
- The approximate spectral projector filter
- Convergence analysis of filtered subspace iteration
- FEAST for generalized Hermitian problem

Convergence analysis



We know: $\mathcal{Q}_k$ captures the $\mathbf{x}_i$:
distance$(\mathbf{x}_i, \mathcal{Q}_k) \to 0$
for $i = 1, 2, \ldots, m$, $m_\lambda \le m \le p$

m_λ is the
exact # of eigenvalues
in $[\lambda_{\min}, \lambda_{\max}]$

But still need to know how to actually obtain x_i and λ_i

FEAST:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$

Loop $k = 1, 2, 3 \ldots$

$Y_k \leftarrow \rho_{\text{quad}}(M)\; Q_{k-1}$

$Q_k \leftarrow \text{orthonormalize}(Y_k)$

$k \leftarrow k + 1$

End Loop:

$Q_k^T M Q_k$ captures WANT, **FAST!**

Examine more closely

Structure of Subspace

Let $Q = [q_1, q_2, \ldots, q_m, q_{m+1}, \ldots, q_p]$ be orthonormal from iteration

$\Downarrow$ orthogonal change of basis

$W = [w_1, w_2, \ldots, w_m, w_{m+1}, \ldots, w_p]$ orthonormal

very close

$X = [x_1, x_2, \ldots, x_m, x_{m+1}, \ldots, x_p, x_{p+1}, \ldots, x_n]$ orthonormal

Structure of Subspace

Let $Q = [q_1, q_2, \ldots, q_m, q_{m+1}, \ldots, q_p]$ be orthonormal from iteration

orthogonal change of basis

$$W = [w_1, w_2, \ldots, w_m, w_{m+1}, \ldots, w_p] \text{ orthonormal}$$

very close

$$X = [x_1, x_2, \ldots, x_m, x_{m+1}, \ldots, x_p, x_{p+1}, \ldots, x_n] \text{ orthonormal}$$

$$X^H W = \quad n \underbrace{\left[\begin{array}{c|c} I_m & 0 \\ \hline 0 & G \end{array}\right]}_{p} + \Delta$$

$$G^H G = I_{p-m}, \ \|\Delta\| = O(\epsilon)$$

Structure of Subspace

Let $Q = [q_1, q_2, \ldots, q_m, q_{m+1}, \ldots, q_p]$ be orthonormal from iteration

$\Downarrow$ orthogonal change of basis

$W = [w_1, w_2, \ldots, w_m, w_{m+1}, \ldots, w_p]$ orthonormal

$$\mathrm{eig}(Q^H M Q) = \mathrm{eig}(W^H M W)$$

$$M = X \Lambda X^H$$

$$X^H W = n \Big\updownarrow \underbrace{\left[\begin{array}{c:c} \boxed{I_m} & 0 \\ \hdashline 0 & \boxed{G} \end{array}\right]}_{p} + \Delta$$

$$W^H M W = (W^H X) \Lambda (X^H W)$$

$$= p \Big\updownarrow \underbrace{\left[\begin{array}{c:c} \boxed{\Lambda_m} & \mathbf{O} \\ \hdashline \mathbf{O} & \boxed{S} \end{array}\right]}_{p} + \Delta'$$

$$S = G^H \Lambda_{n-m} G \qquad \|\Delta'\| = O(\epsilon)$$

$$G^H G = I_{p-m}, \ \|\Delta\| = O(\epsilon)$$

Structure of Subspace

Let $Q = [q_1, q_2, \ldots, q_m, q_{m+1}, \ldots, q_p]$ be orthonormal from iteration

⇓ orthogonal change of basis

$W = [w_1, w_2, \ldots, w_m, w_{m+1}, \ldots, w_p]$ orthonormal

$$W^H M W = (W^H X)\Lambda(X^H W)$$

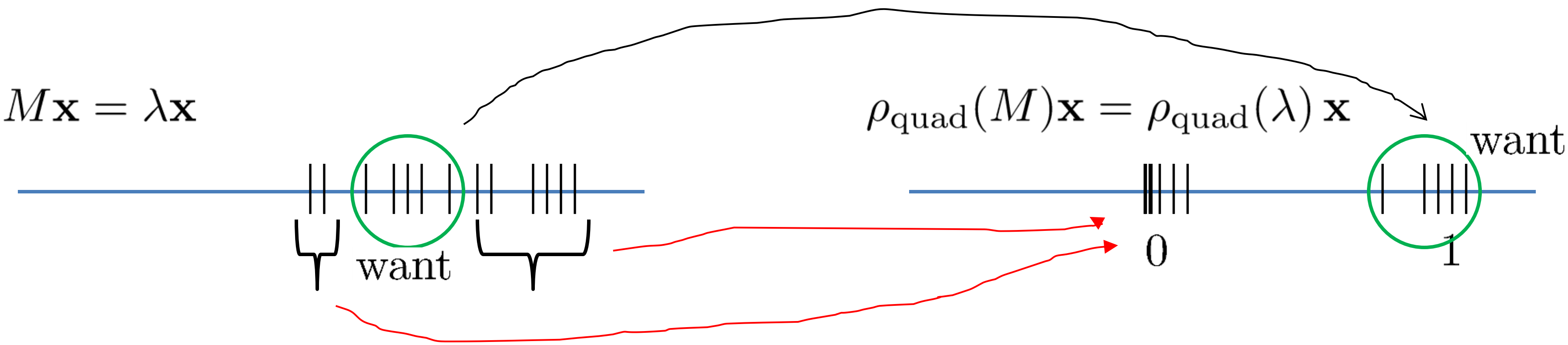
$$= \underbrace{\begin{bmatrix} \Lambda_m & \mathbf{O} \\ \mathbf{O} & S \end{bmatrix}}_{p \times p} + \Delta'$$

So m eigenvalues of $Q^H A Q$ close to targets

And the remaining $p - m$ between min/max of remaining $n - m$ eigenvalues

$$S = G^H \Lambda_{n-m} G \qquad \|\Delta'\| = O(\epsilon)$$

FEAST as subspace Iteration



FEAST:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$

Loop $k = 1, 2, 3 \ldots$

$Y_k \leftarrow \rho_{\text{quad}}(M)\, Q_{k-1}$

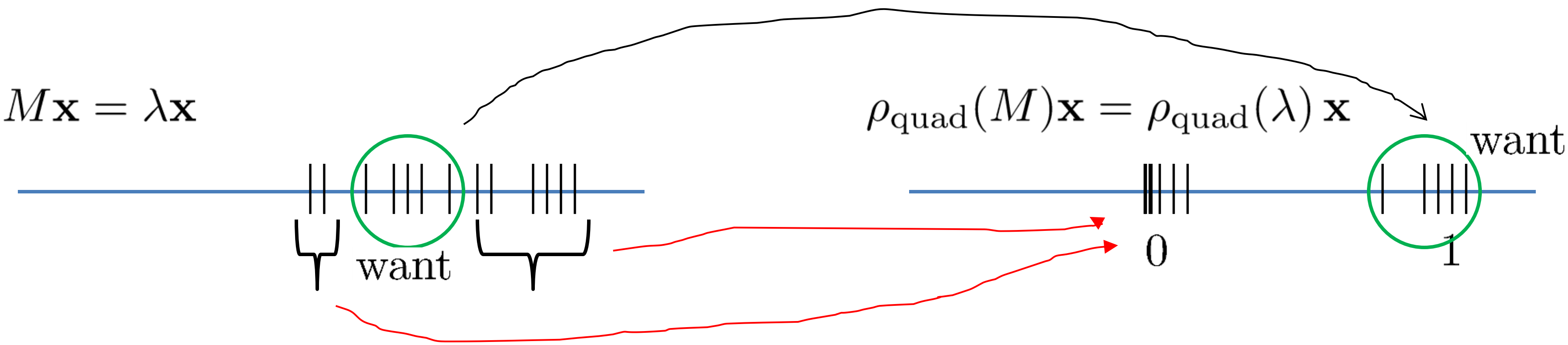
$Q_k \leftarrow \text{orthonormalize}(Y_k)$

$k \leftarrow k + 1$

End Loop:

$Q_k^T M Q_k$ captures WANT, **FAST!**

FEAST as subspace Iteration



FEAST:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$

Loop $k = 1, 2, 3 \ldots$

$Y_k \leftarrow \rho_{\text{quad}}(M)\; Q_{k-1}$

$Q_k \leftarrow \text{orthonormalize}(Y_k)$

$k \leftarrow k + 1$

End Loop:

$Q_k^T M Q_k$ captures WANT, **FAST!**

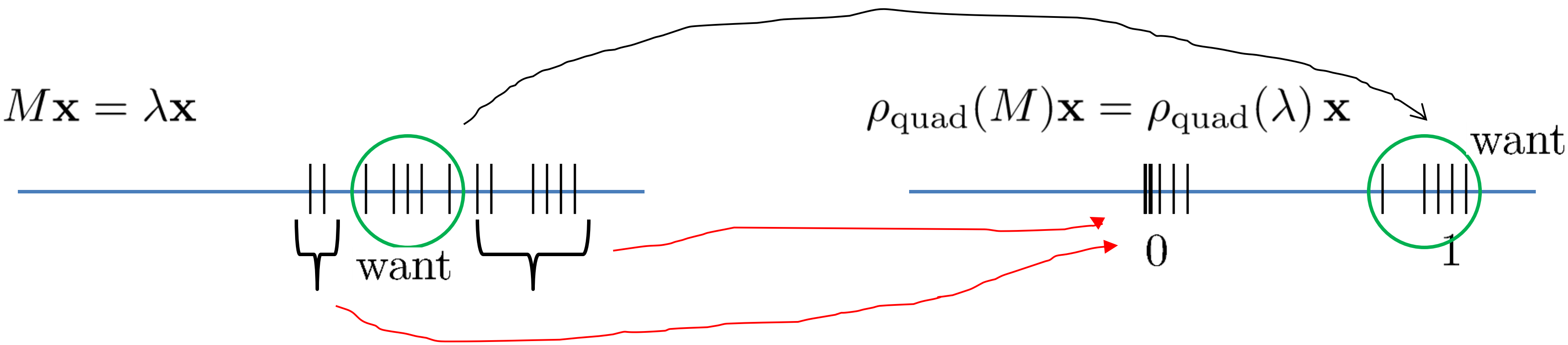
To orthonormalize, one can

1. Form $A_p \leftarrow Y_k^H M Y_k$, $B_p \leftarrow Y_k^H Y_k$
2. Solve GHEP $A_p V = B_p V \Lambda'$
3. $Q_k \leftarrow Y_k V$

FEAST for Hermitian Problems

- FEAST as filtered subspace iteration for standard Hermitian eigenproblem
- The approximate spectral projector filter
- Convergence analysis of filtered subspace iteration
- FEAST for generalized Hermitian problem

FEAST as subspace Iteration



FEAST:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$

Loop $k = 1, 2, 3 \ldots$

$Y_k \leftarrow \rho_{\text{quad}}(M)\, Q_{k-1}$

$A_p \leftarrow Y_k^H M Y_k,\ B_p \leftarrow Y_k^H Y_k$

Solve GHEP $A_p V = B_p V \Lambda'$

$Q_k \leftarrow Y_k V$

$k \leftarrow k + 1$

End Loop:

FEAST as subspace Iteration

Standard eigenvalue problem

$A\mathbf{x} = \lambda\mathbf{x},\ A^H = A$

$A = X\Lambda X^H,\ X^H X = I$

FEAST:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$

Loop $k = 1, 2, 3 \ldots$

$Y_k \leftarrow \rho_{\text{quad}}(A)\, Q_{k-1}$

$A_p \leftarrow Y_k^H A Y_k,\ B_p \leftarrow Y_k^H Y_k$

Solve GHEP $A_p V = B_p V \Lambda'$

$Q_k \leftarrow Y_k V$

$k \leftarrow k + 1$

End Loop:

FEAST as subspace Iteration

Standard eigenvalue problem
$A\mathbf{x} = \lambda\mathbf{x},\ A^H = A$

$A = X\Lambda X^H,\ X^H X = I$

Generalized eigenvalue problem
$A\mathbf{x} = \lambda B\,\mathbf{x},\ \ A, B$ Hermitian
B positive definite
$B^{-1}A = X\Lambda X^{-1}, \quad X^H BX = I$

FEAST:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$

Loop $k = 1, 2, 3\ldots$

$Y_k \leftarrow \rho_{\text{quad}}(A)\, Q_{k-1}$

$A_p \leftarrow Y_k^H A Y_k,\ B_p \leftarrow Y_k^H Y_k$

Solve GHEP $A_p V = B_p V \Lambda'$

$Q_k \leftarrow Y_k V$

$k \leftarrow k + 1$

End Loop:

FEAST as subspace Iteration

Standard eigenvalue problem
$A\mathbf{x} = \lambda\mathbf{x},\ A^H = A$

$A = X\Lambda X^H,\ X^H X = I$

FEAST:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$

Loop $k = 1, 2, 3 \ldots$

$Y_k \leftarrow \rho_{\text{quad}}(A)\, Q_{k-1}$

$A_p \leftarrow Y_k^H A Y_k,\ B_p \leftarrow Y_k^H Y_k$

Solve GHEP $A_p V = B_p V \Lambda'$

$Q_k \leftarrow Y_k V$

$k \leftarrow k + 1$

End Loop:

Generalized eigenvalue problem
$A\mathbf{x} = \lambda B\,\mathbf{x},\ \ A, B$ Hermitian
B positive definite
$B^{-1}A = X\Lambda X^{-1}, \quad X^H B X = I$

$\rho(B^{-1}A) = X\rho(\Lambda)X^{-1}$

FEAST as subspace Iteration

Standard eigenvalue problem
$A\mathbf{x} = \lambda\mathbf{x},\ A^H = A$

$A = X\Lambda X^H,\ X^H X = I$

FEAST:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$

Loop $k = 1, 2, 3 \ldots$

$Y_k \leftarrow \rho_{\text{quad}}(A)\ Q_{k-1}$

$A_p \leftarrow Y_k^H A Y_k,\ B_p \leftarrow Y_k^H Y_k$

Solve GHEP $A_p V = B_p V \Lambda'$

$Q_k \leftarrow Y_k V$
$k \leftarrow k + 1$

End Loop:

Generalized eigenvalue problem
$A\mathbf{x} = \lambda B\,\mathbf{x},\ \ A, B$ Hermitian
B positive definite
$B^{-1}A = X\Lambda X^{-1}, \quad X^H B X = I$

$\rho(B^{-1}A) = X\rho(\Lambda)X^{-1}$

$\rho_{\text{quad}}(B^{-1}A)$

$= \sum_{k=1}^{q} w_k (z_k I - B^{-1}A)^{-1}$

FEAST as subspace Iteration

Standard eigenvalue problem
$A\mathbf{x} = \lambda\mathbf{x},\ A^H = A$

$A = X\Lambda X^H,\ X^H X = I$

FEAST:

Random $Q_0 = [y_1, y_2, \ldots, y_p],\ p \ll n$

Loop $k = 1, 2, 3 \ldots$

$Y_k \leftarrow \rho_{\text{quad}}(A)\, Q_{k-1}$

$A_p \leftarrow Y_k^H A Y_k,\ B_p \leftarrow Y_k^H Y_k$

Solve GHEP $A_p V = B_p V \Lambda'$

$Q_k \leftarrow Y_k V$

$k \leftarrow k + 1$

End Loop:

Generalized eigenvalue problem
$A\mathbf{x} = \lambda B\,\mathbf{x},\ \ A, B$ Hermitian
B positive definite
$B^{-1}A = X\Lambda X^{-1}, \quad X^H B X = I$

$$\rho(B^{-1}A) = X\rho(\Lambda)X^{-1}$$

$$\rho_{\text{quad}}(B^{-1}A)$$

$$= \sum_{k=1}^{q} w_k (z_k I - B^{-1}A)^{-1}$$

$$= \sum_{k=1}^{q} w_k (z_k B - A)^{-1}\, B$$

FEAST as subspace Iteration

Standard eigenvalue problem
$A\mathbf{x} = \lambda \mathbf{x},\ A^H = A$

$A = X \Lambda X^H,\ X^H X = I$

FEAST:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$
Loop $k = 1, 2, 3 \ldots$
$Y_k \leftarrow \rho_{\text{quad}}(A)\, Q_{k-1}$
$A_p \leftarrow Y_k^H A Y_k,\ B_p \leftarrow Y_k^H Y_k$
Solve GHEP $A_p V = B_p V \Lambda'$
$Q_k \leftarrow Y_k V$
$k \leftarrow k + 1$
End Loop:

Generalized eigenvalue problem
$A\mathbf{x} = \lambda B\,\mathbf{x},\ \ A, B$ Hermitian
B positive definite
$B^{-1} A = X \Lambda X^{-1}, \quad X^H B X = I$

FEAST:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$
Loop $k = 1, 2, 3 \ldots$
$Y_k \leftarrow \rho_{\text{quad}}(B^{-1} A)\, Q_{k-1}$
$A_p \leftarrow Y_k^H A Y_k,\ B_p \leftarrow Y_k^H B Y_k$
Solve GHEP $A_p V = B_p V \Lambda'$
$Q_k \leftarrow Y_k V$
$k \leftarrow k + 1$
End Loop: