High-Level Outline

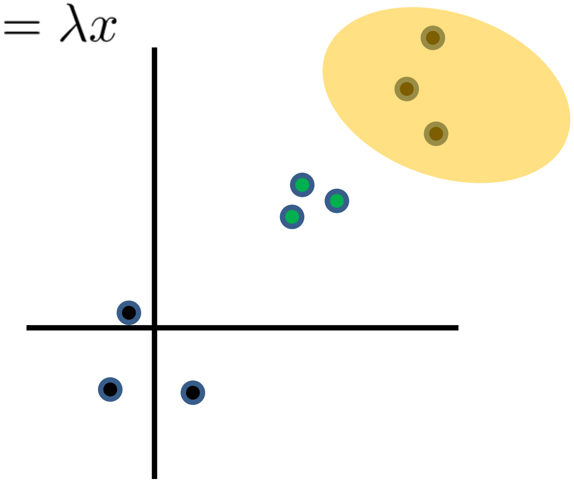
- FEAST as filtered subspace iteration, for Hermitian problems
- FEAST for non-Hermitian problems
- SS method as filtered Krylov subspace method
- Function approximation and computer arithmetic studies related to the filter

Review: Subspace Iteration

Matrix M, n-by-n

eigenvalues λ, eigenvectors x

$Mx = \lambda x$



Subspace Iteration:

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

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

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

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

End Loop:

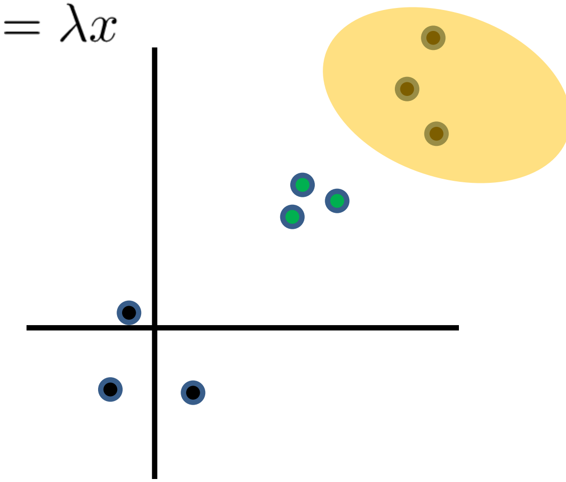
$Q_k^T M Q_k$ captures the "big"

Review: Subspace Iteration

Matrix M, n-by-n

eigenvalues λ, eigenvectors x

$Mx = \lambda x$



Informally: If $|\lambda_i| > |\lambda_{p+1}|$ then
distance$(x_i, \mathcal{Q}_k) = O(|\lambda_{p+1}/\lambda_i|^k)$

Subspace Iteration:

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

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

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

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

End Loop:

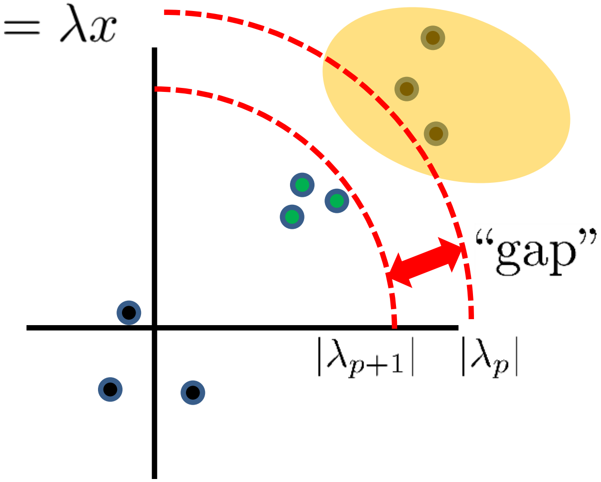
$Q_k^T M Q_k$ captures the "big"

Review: Subspace Iteration

Matrix M, n-by-n

eigenvalues λ, eigenvectors x

$Mx = \lambda x$



Informally: If $|\lambda_i| > |\lambda_{p+1}|$ then distance$(x_i, \mathcal{Q}_k) = O(|\lambda_{p+1}/\lambda_i|^k)$

Subspace Iteration:

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

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

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

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

End Loop:

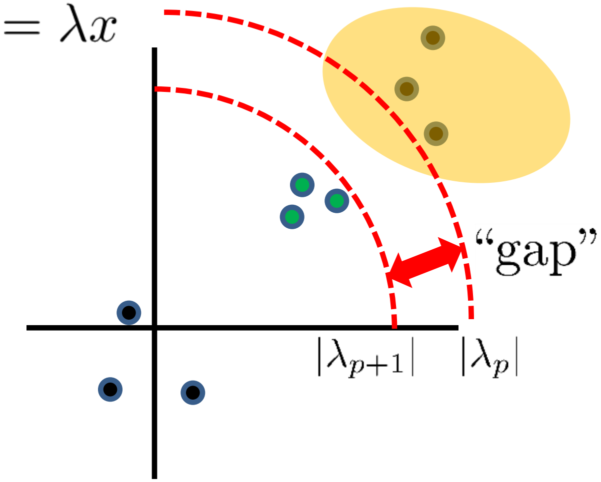
$Q_k^T M Q_k$ captures the "big"

Review: Subspace Iteration

Matrix M, n-by-n

eigenvalues λ, eigenvectors x

$Mx = \lambda x$



Informally: If $|\lambda_i| > |\lambda_{p+1}|$ then distance$(x_i, \mathcal{Q}_k) = O(|\lambda_{p+1}/\lambda_i|^k)$

Subspace Iteration:

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

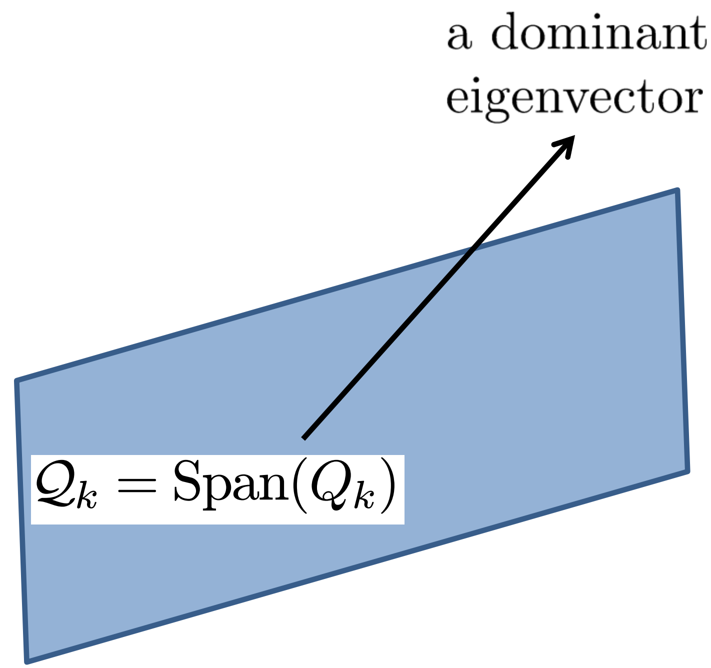
Loop $k = 1, 2, 3 \ldots$

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

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

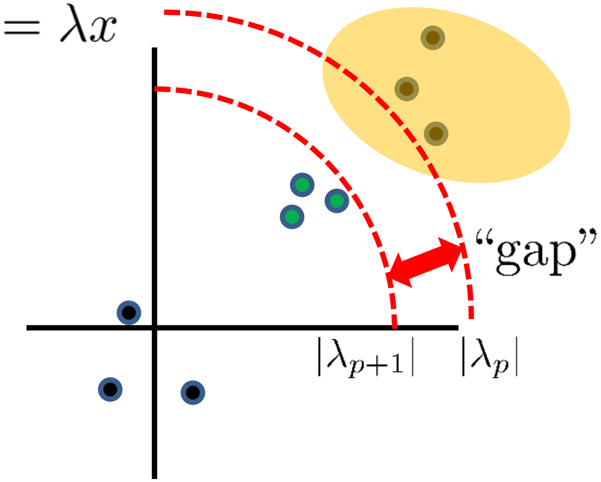
End Loop:

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



Review: Subspace Iteration

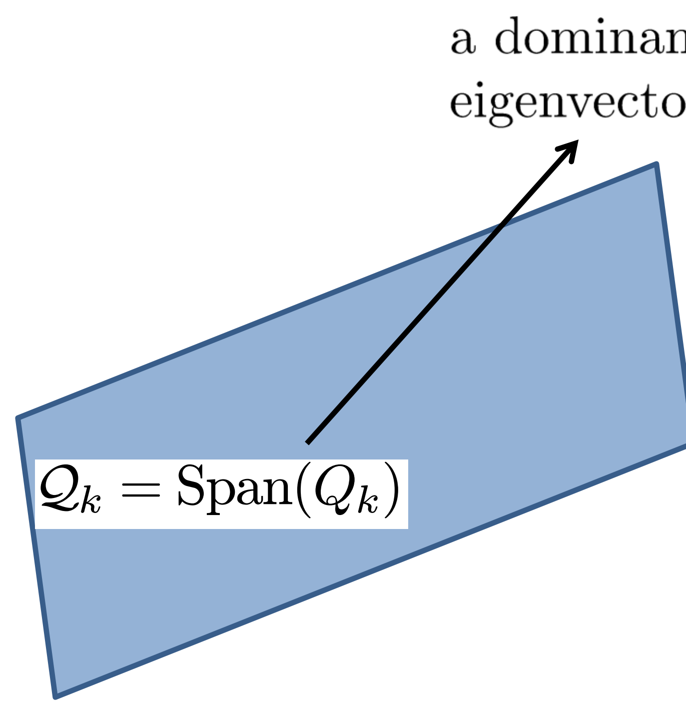
Matrix M, n-by-n
eigenvalues λ, eigenvectors x
$Mx = \lambda x$



Informally: If $|\lambda_i| > |\lambda_{p+1}|$ then
distance$(x_i, \mathcal{Q}_k) = O(|\lambda_{p+1}/\lambda_i|^k)$

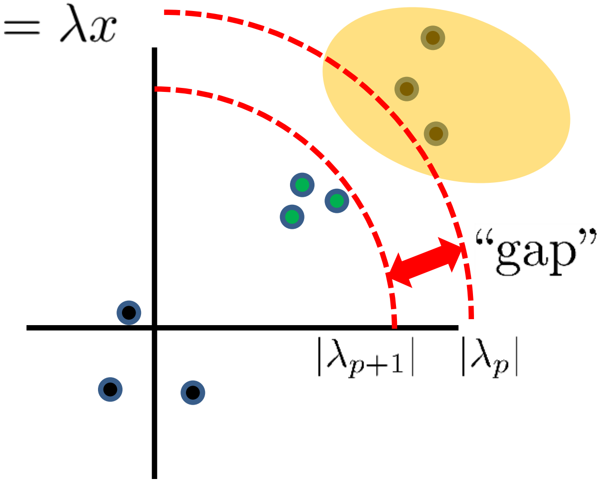
Subspace Iteration:

Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$
Loop $k = 1, 2, 3 \ldots$
  $Y_k \leftarrow M \; Q_{k-1}$
  $Q_k \leftarrow \text{orthonormalize}(Y_k)$
End Loop:
$Q_k^T M Q_k$ captures the "big"



Review: Subspace Iteration

Matrix M, n-by-n
eigenvalues λ, eigenvectors x
$Mx = \lambda x$



Informally: If $|\lambda_i| > |\lambda_{p+1}|$ then
distance$(x_i, \mathcal{Q}_k) = O(|\lambda_{p+1}/\lambda_i|^k)$

Subspace Iteration:

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

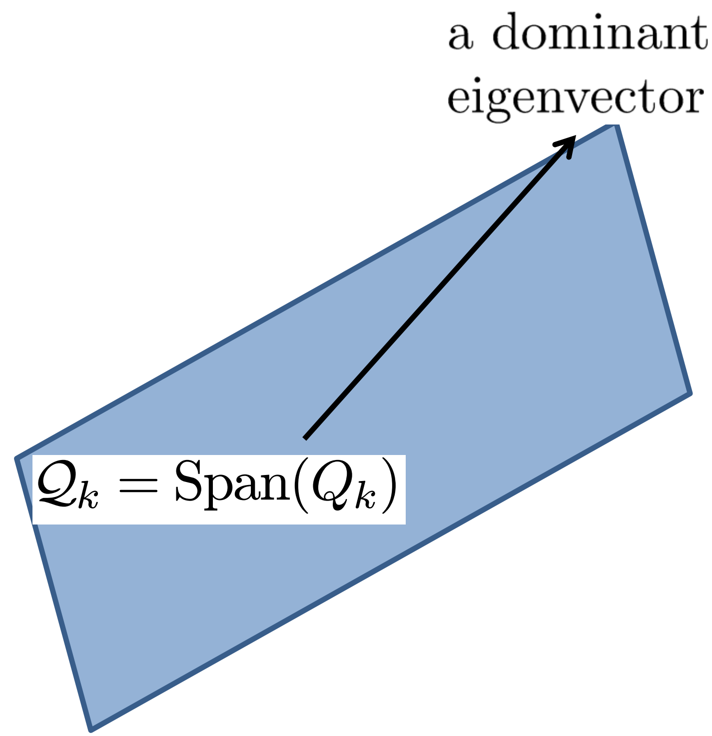
Loop $k = 1, 2, 3 \ldots$

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

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

End Loop:

$Q_k^T M Q_k$ captures the “big”

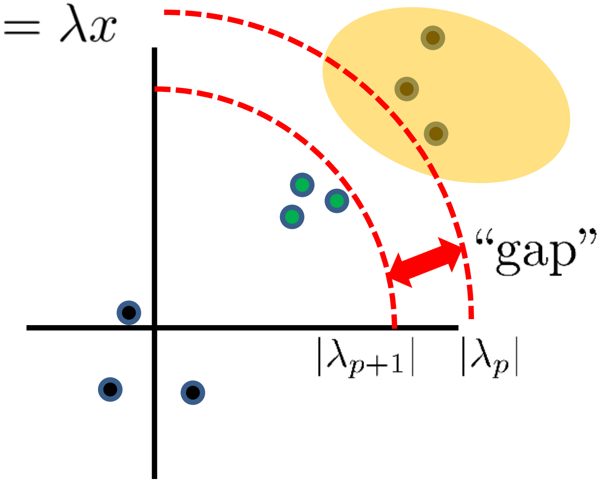


Review: Subspace Iteration

Matrix M, n-by-n

eigenvalues λ, eigenvectors x

$Mx = \lambda x$



Informally: If $|\lambda_i| > |\lambda_{p+1}|$ then
distance$(x_i, \mathcal{Q}_k) = O(|\lambda_{p+1}/\lambda_i|^k)$

Subspace Iteration:

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

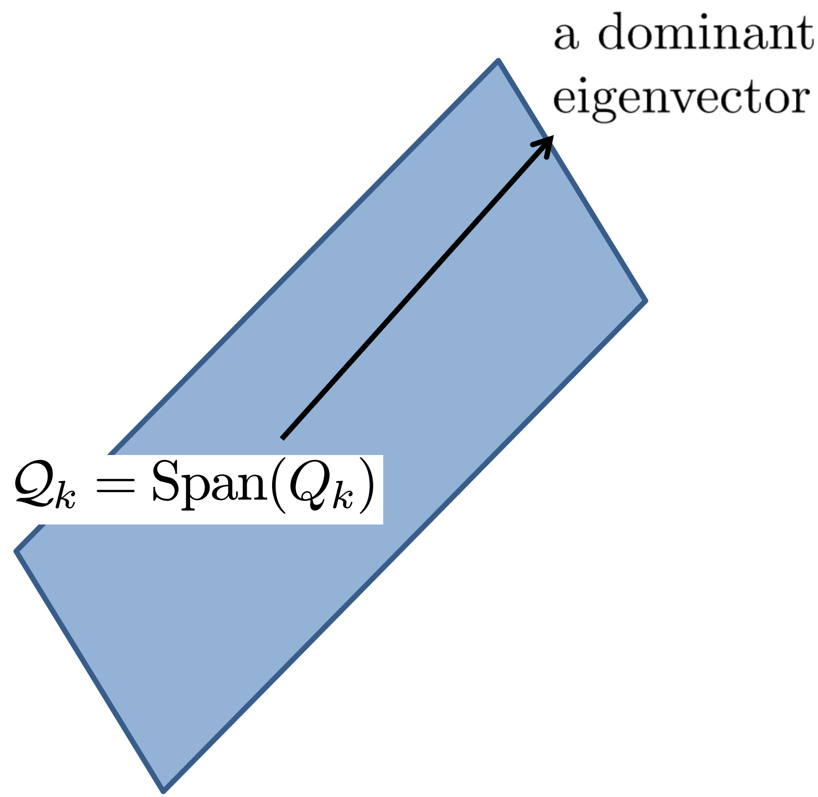
Loop $k = 1, 2, 3 \ldots$

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

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

End Loop:

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

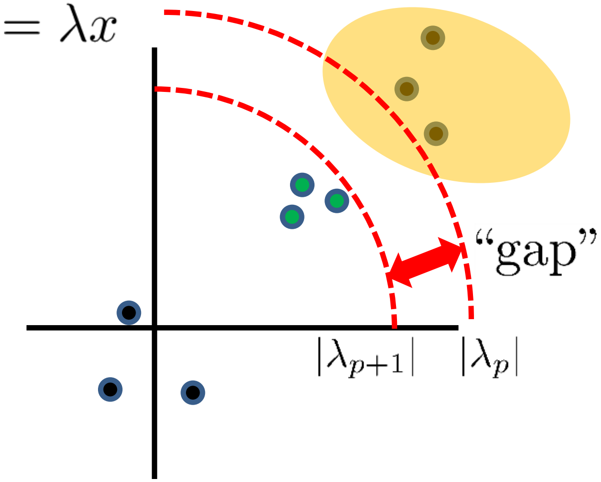


Review: Subspace Iteration

Matrix M, n-by-n

eigenvalues λ, eigenvectors x

$Mx = \lambda x$



Informally: If $|\lambda_i| > |\lambda_{p+1}|$ then distance$(x_i, \mathcal{Q}_k) = O(|\lambda_{p+1}/\lambda_i|^k)$

Subspace Iteration:

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

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

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

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

End Loop:

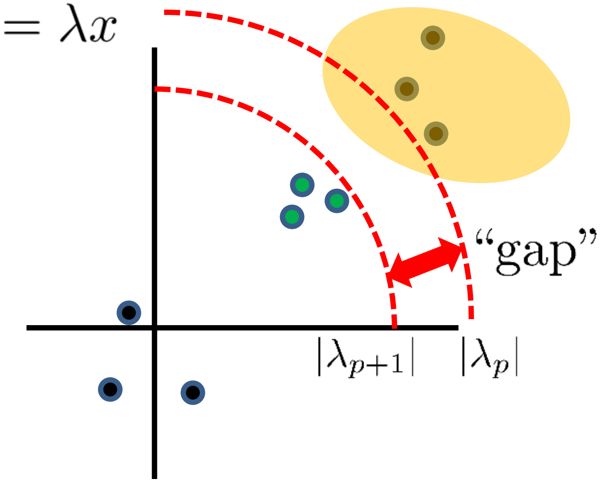
$Q_k^T M Q_k$ captures the “big”

Review: Subspace Iteration

Matrix M, n-by-n

eigenvalues λ, eigenvectors x

$Mx = \lambda x$



Informally: If $|\lambda_i| > |\lambda_{p+1}|$ then

distance$(x_i, \mathcal{Q}_k) = O(|\lambda_{p+1}/\lambda_i|^k)$

Subspace Iteration:

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

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

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

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

End Loop:

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

Only works for "big"

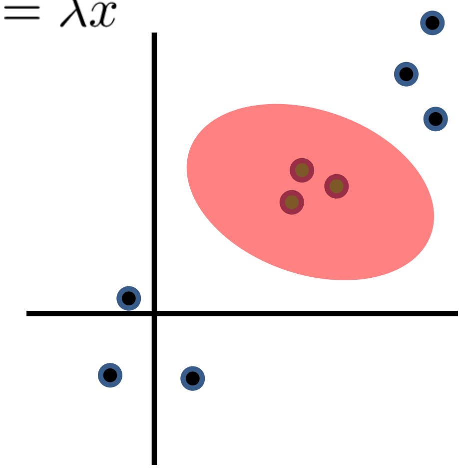
Speed at the mercy of "gap"

Can be VERY slow

Review: Subspace Iteration

Matrix M, n-by-n
eigenvalues λ, eigenvectors x

$Mx = \lambda x$



Subspace Iteration:

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

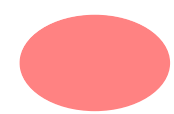
Loop $k = 1, 2, 3 \ldots$

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

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

End Loop:

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

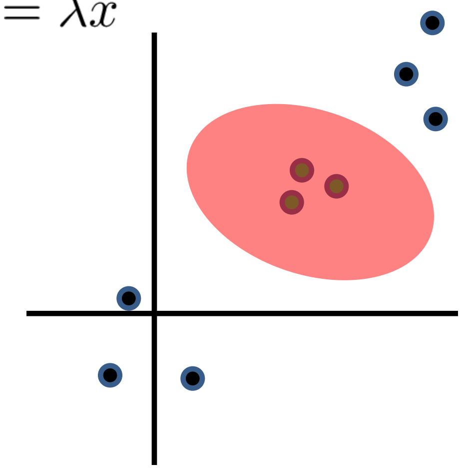
Will not work at all
if you want 

FEAST Algorithm at a Glance

Matrix M, n-by-n

eigenvalues λ, eigenvectors x

$Mx = \lambda x$



Subspace Iteration:

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

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

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

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

End Loop:

$Q_k^T M Q_k$ captures the “big”

FEAST Algorithm:

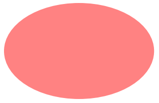
Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$

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

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

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

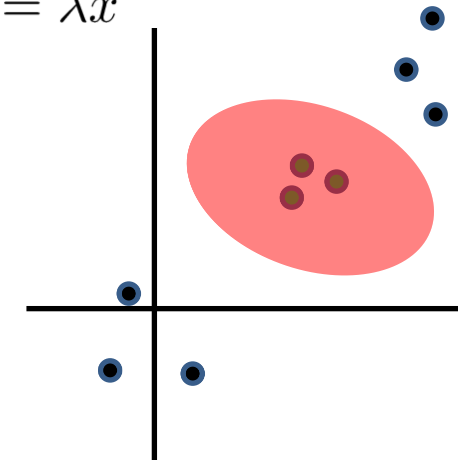
End Loop:

$Q_k^T M Q_k$ captures 

FEAST Algorithm at a Glance

Matrix M, n-by-n
eigenvalues λ, eigenvectors x

$Mx = \lambda x$



Subspace Iteration:

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

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

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

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

End Loop:

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

FEAST Algorithm:

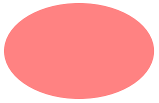
Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$

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

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

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

End Loop:

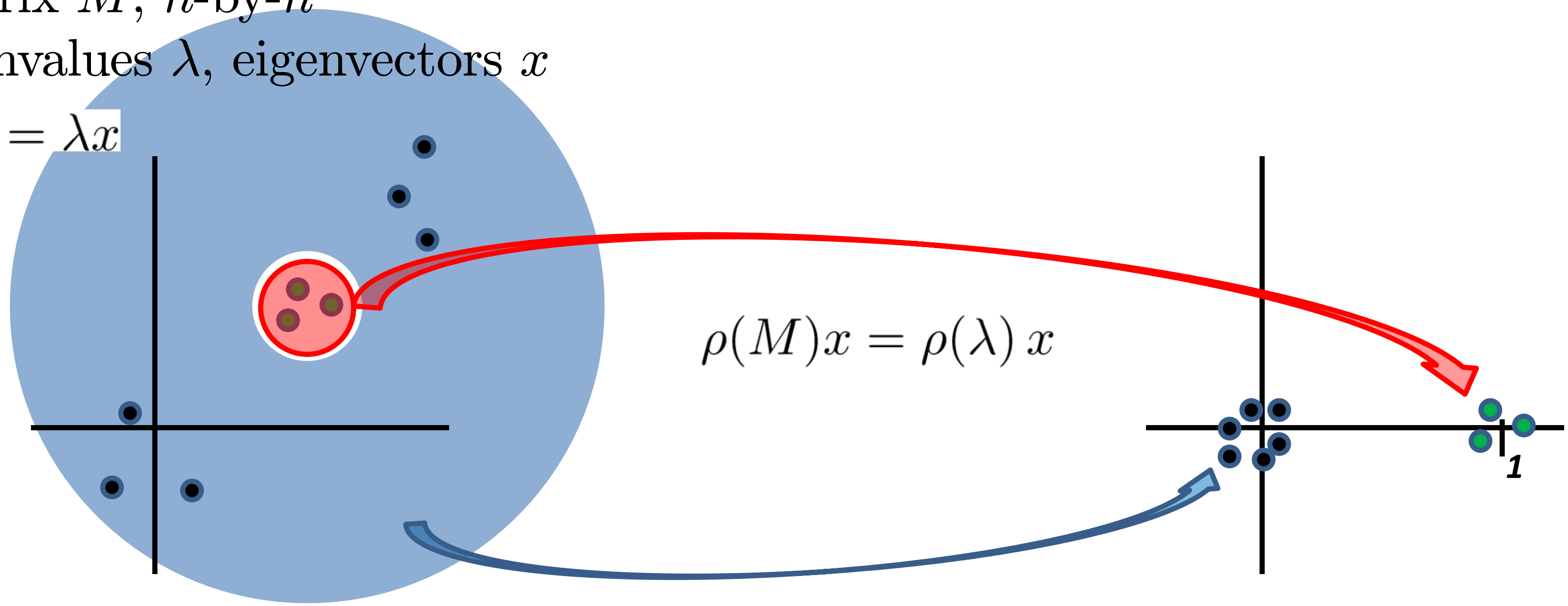
$Q_k^T M Q_k$ captures 

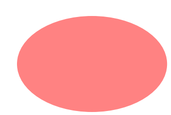
FEAST Algorithm at a Glance

Matrix M, n-by-n

eigenvalues λ, eigenvectors x

$Mx = \lambda x$



Subspace Iteration:	*FEAST Algorithm:*
Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$	Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$
Loop $k = 1, 2, 3 \ldots$	Loop $k = 1, 2, 3 \ldots$
$Y_k \leftarrow M \; Q_{k-1}$	$Y_k \leftarrow \rho(M)\, Q_{k-1}$ heart of FEAST
$Q_k \leftarrow \text{orthonormalize}(Y_k)$	$Q_k \leftarrow \text{orthonormalize}(Y_k)$
End Loop:	End Loop:
$Q_k^T M Q_k$ captures the “big”	$Q_k^T M Q_k$ captures 

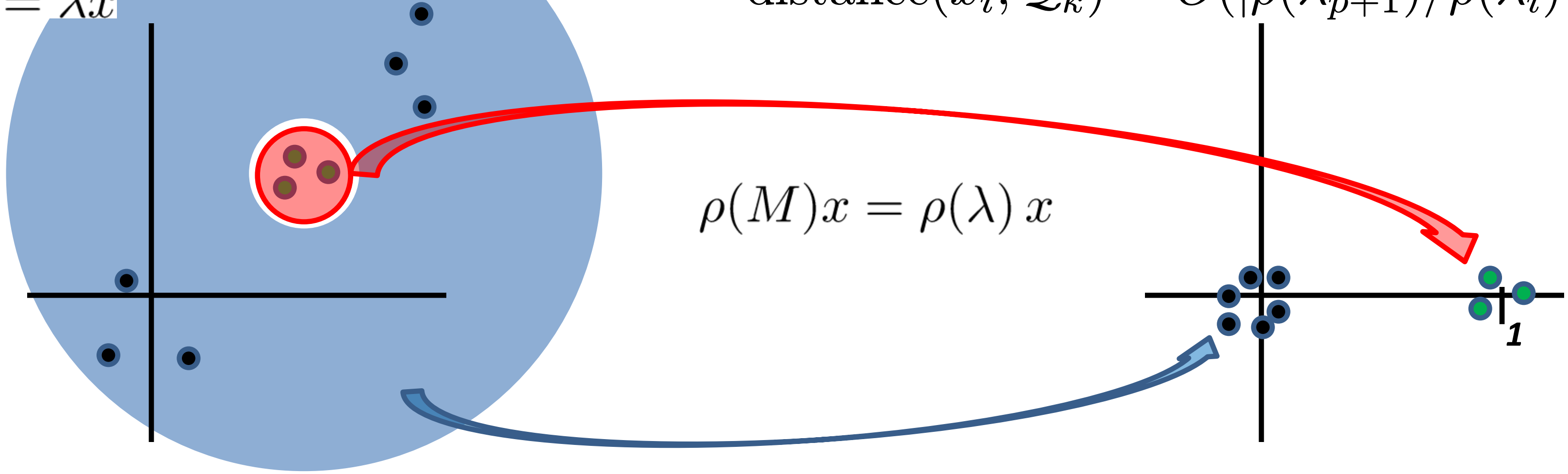
FEAST Algorithm at a Glance

Matrix M, n-by-n

eigenvalues λ, eigenvectors x

$Mx = \lambda x$

Informally: If $|\rho(\lambda_i)| > |\rho(\lambda_{p+1})|$ then distance$(x_i, \mathcal{Q}_k) = O(|\rho(\lambda_{p+1})/\rho(\lambda_i)|^k)$



$\rho(M)x = \rho(\lambda)\, x$

Subspace Iteration:

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

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

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

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

End Loop:

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

FEAST Algorithm:

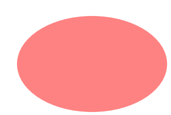
Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$

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

$\quad Y_k \leftarrow \rho(M)\, Q_{k-1}$ heart of FEAST

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

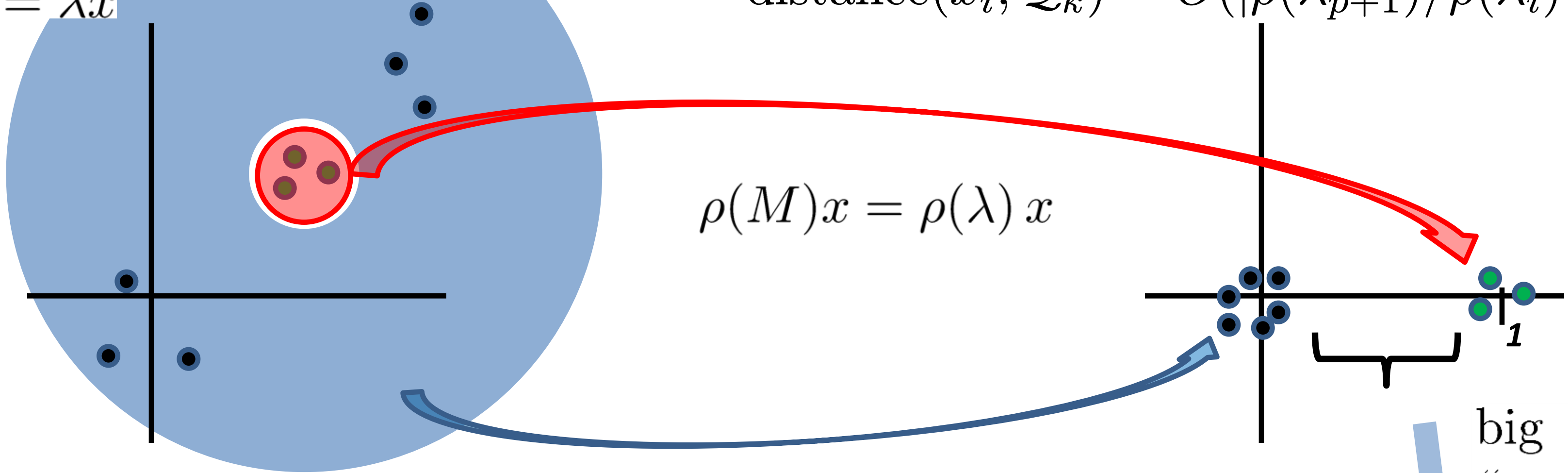
End Loop:

$Q_k^T M Q_k$ captures 

FEAST Algorithm at a Glance

Matrix M, n-by-n
eigenvalues λ, eigenvectors x
$Mx = \lambda x$

Informally: If $|\rho(\lambda_i)| > |\rho(\lambda_{p+1})|$ then
$\text{distance}(x_i, \mathcal{Q}_k) = O(|\rho(\lambda_{p+1})/\rho(\lambda_i)|^k)$



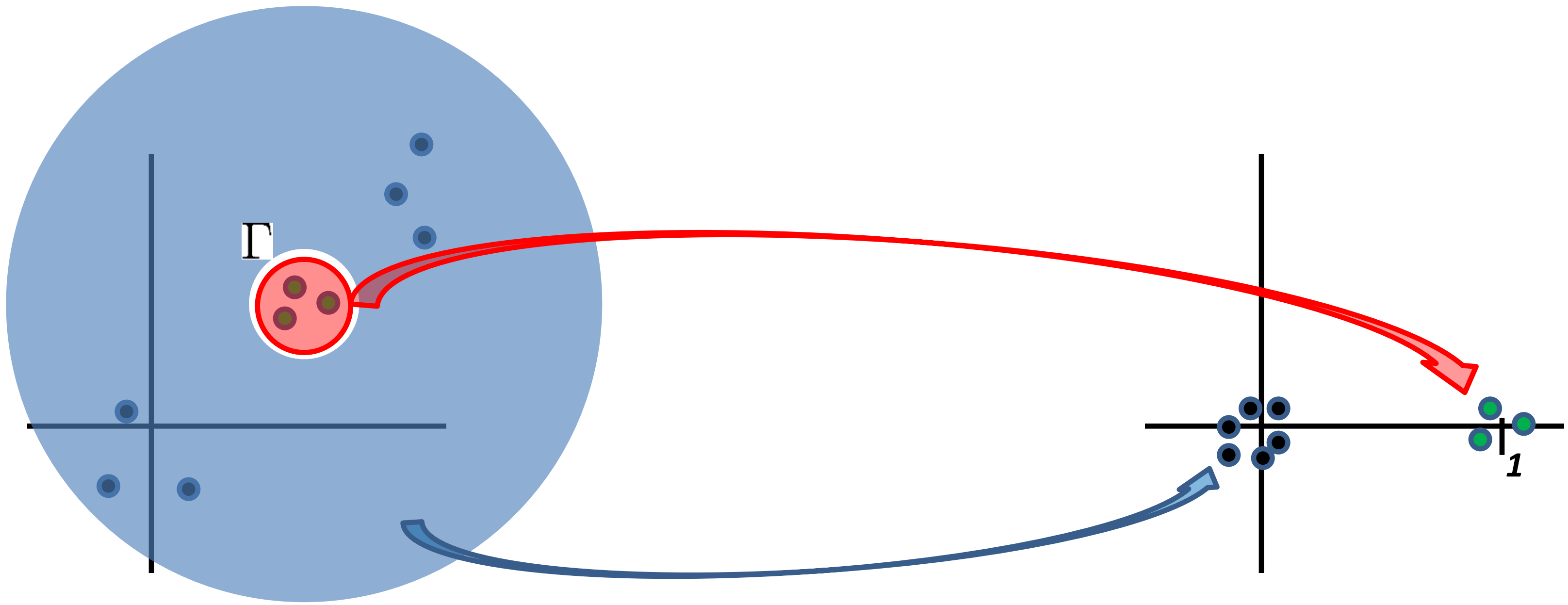
Subspace Iteration:

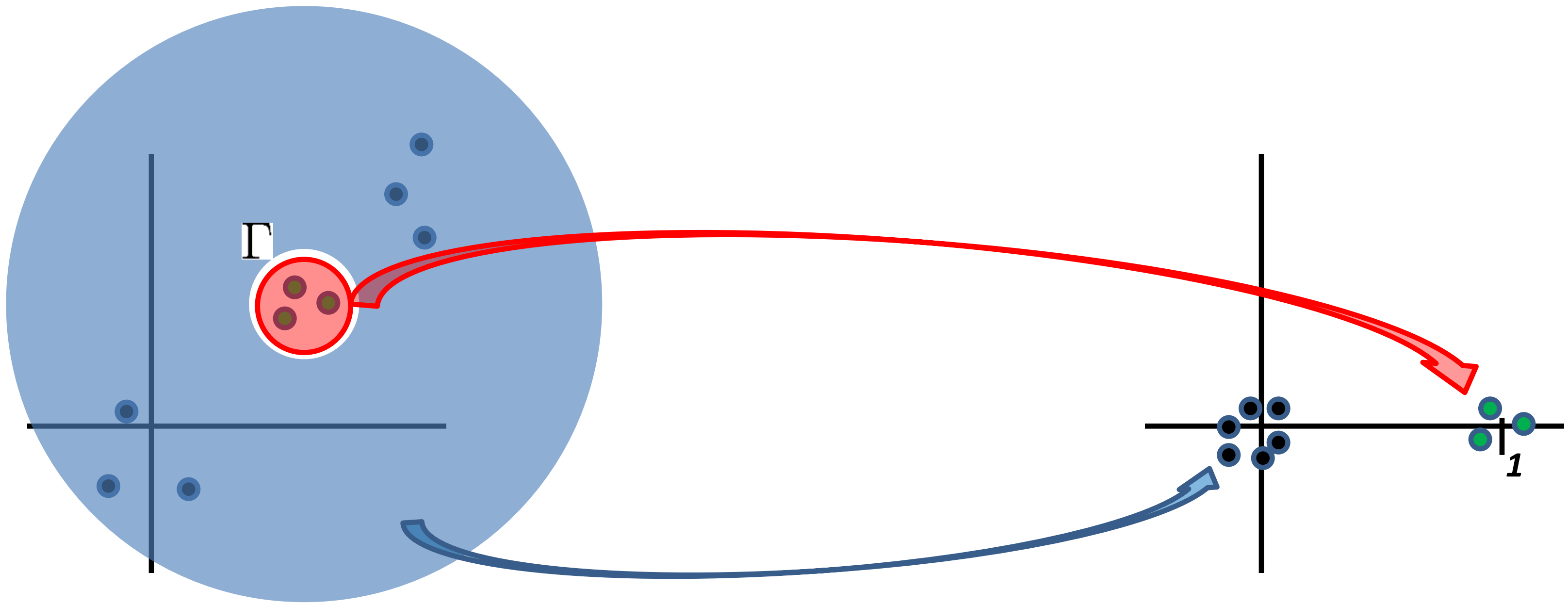
Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$
Loop $k = 1, 2, 3 \ldots$
$Y_k \leftarrow \;M\; Q_{k-1}$
$Q_k \leftarrow \text{orthonormalize}(Y_k)$
End Loop:
$Q_k^T M Q_k$ captures the "big"

FEAST Algorithm:

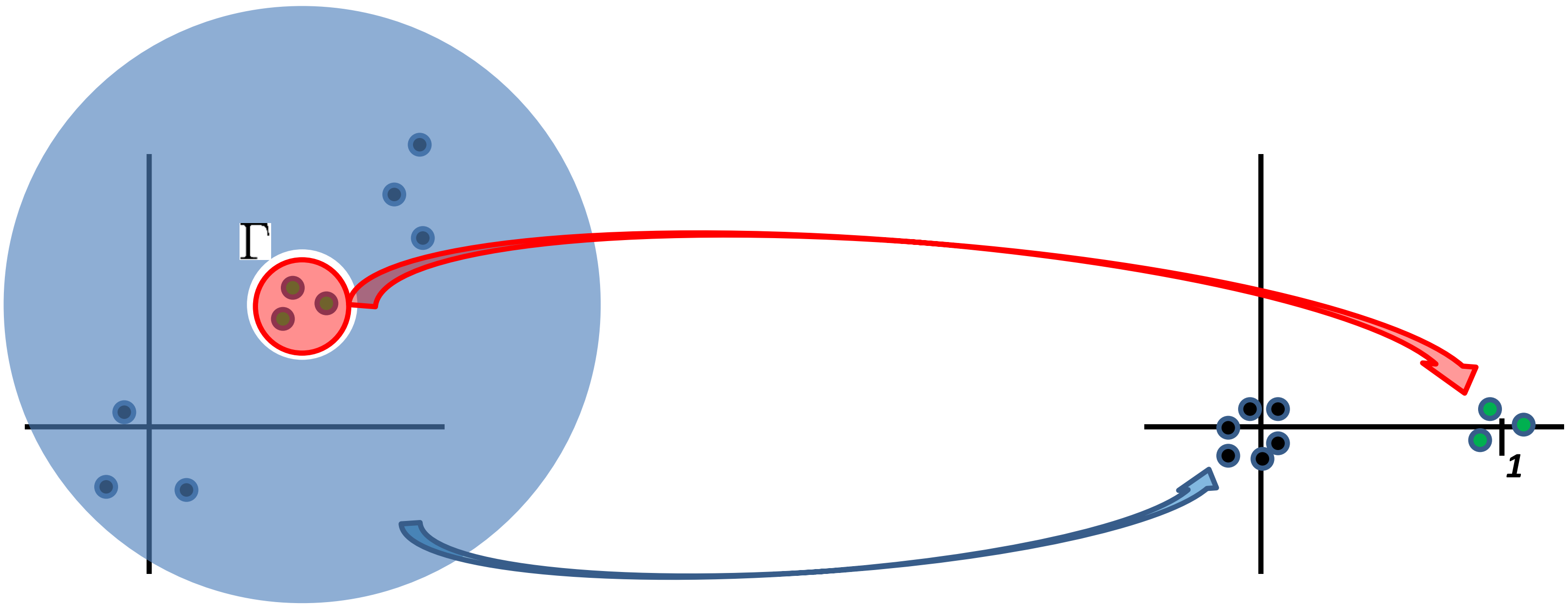
Random $Q_0 = [y_1, y_2, \ldots, y_p]$, $p \ll n$
Loop $k = 1, 2, 3 \ldots$
$Y_k \leftarrow \rho(M)\, Q_{k-1}$ heart of FEAST
$Q_k \leftarrow \text{orthonormalize}(Y_k)$
End Loop:
$Q_k^T M Q_k$ captures FAST!

Construction of the Accelerator

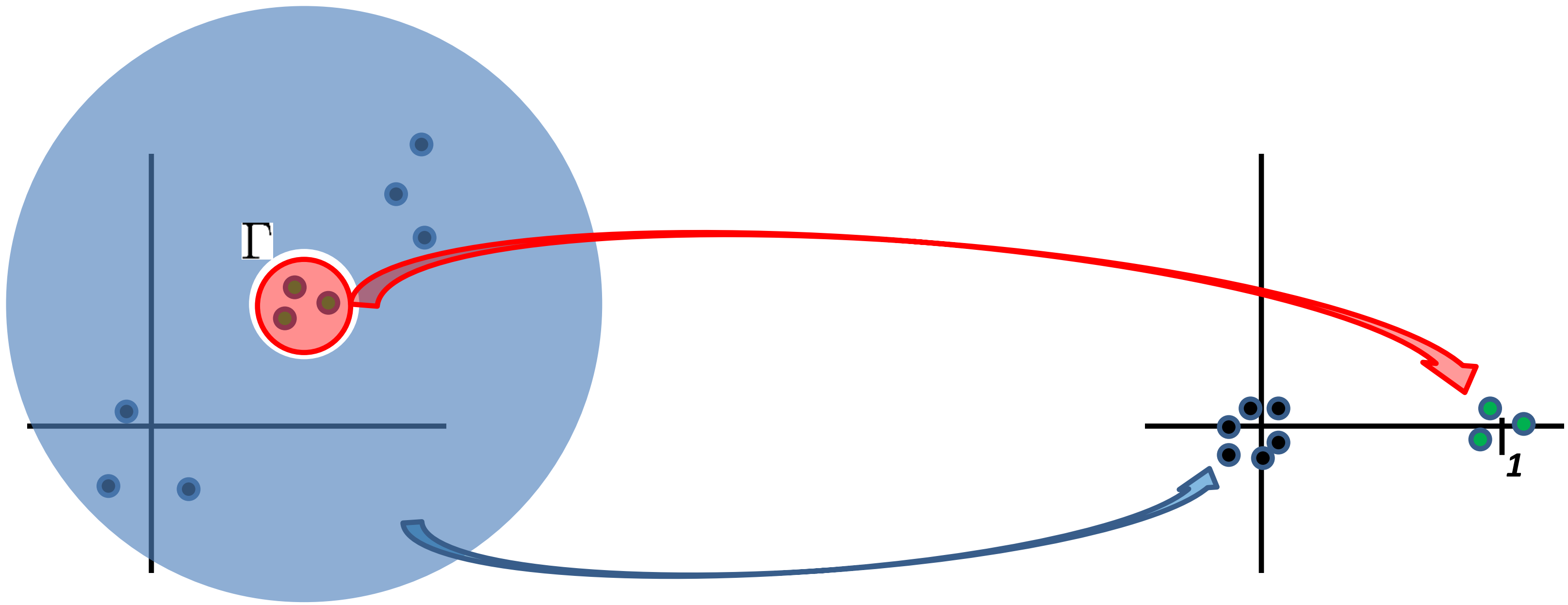

Γ
1



$$\pi(\lambda) = \frac{1}{2\pi\iota} \oint_\Gamma \frac{1}{z-\lambda} dz = \begin{cases} 1 & \lambda \in \\ 0 & \lambda \notin \end{cases}$$

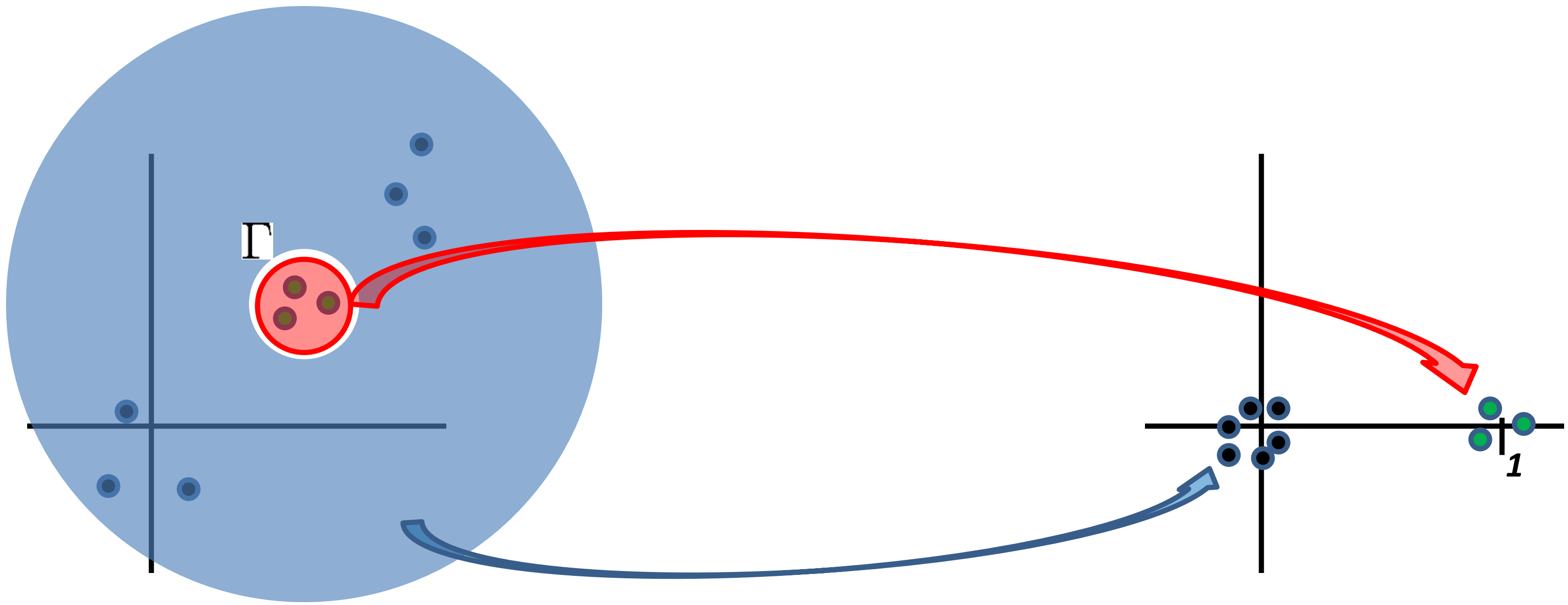


$$\sum_{k=1}^{q} w_k (z_k - \lambda)^{-1} = \rho(\lambda) \approx \pi(\lambda) = \frac{1}{2\pi\iota} \oint_{\Gamma} \frac{1}{z-\lambda} dz = \begin{cases} 1 & \lambda \in \\ 0 & \lambda \notin \end{cases}$$



$$\sum_{k=1}^{q} w_k (z_k - \lambda)^{-1} = \rho(\lambda) \approx \; \pi(\lambda) = \frac{1}{2\pi\iota} \oint_{\Gamma} \frac{1}{z-\lambda} dz = \begin{cases} 1 & \lambda \in \\ 0 & \lambda \notin \end{cases}$$

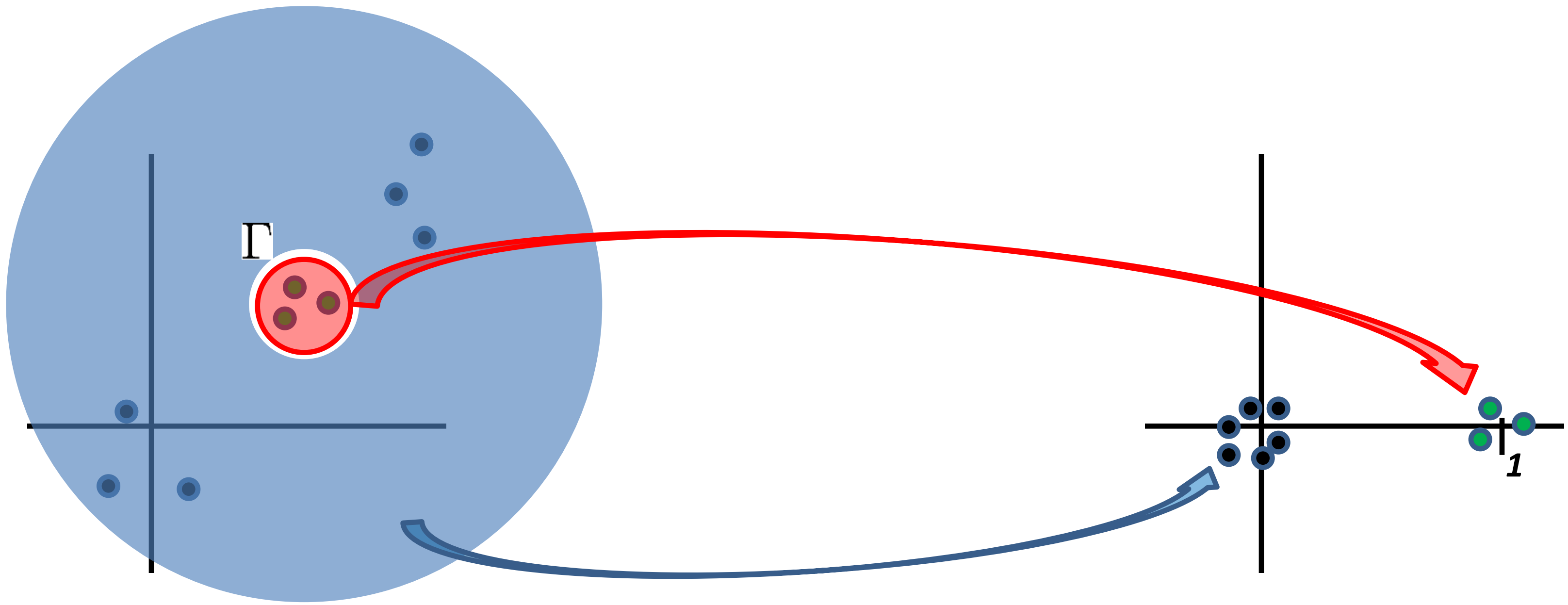
$$\rho(M) \cdot Y = \sum_{k=1}^{q} w_k (z_k I - M)^{-1} \cdot Y$$



$$\sum_{k=1}^{q} w_k (z_k - \lambda)^{-1} = \rho(\lambda) \approx \pi(\lambda) = \frac{1}{2\pi\iota} \oint_\Gamma \frac{1}{z-\lambda} dz = \begin{cases} 1 & \lambda \in \\ 0 & \lambda \notin \end{cases}$$

$$\rho(M) \cdot Y = \sum_{k=1}^{q} w_k (z_k I - M)^{-1} \cdot Y$$

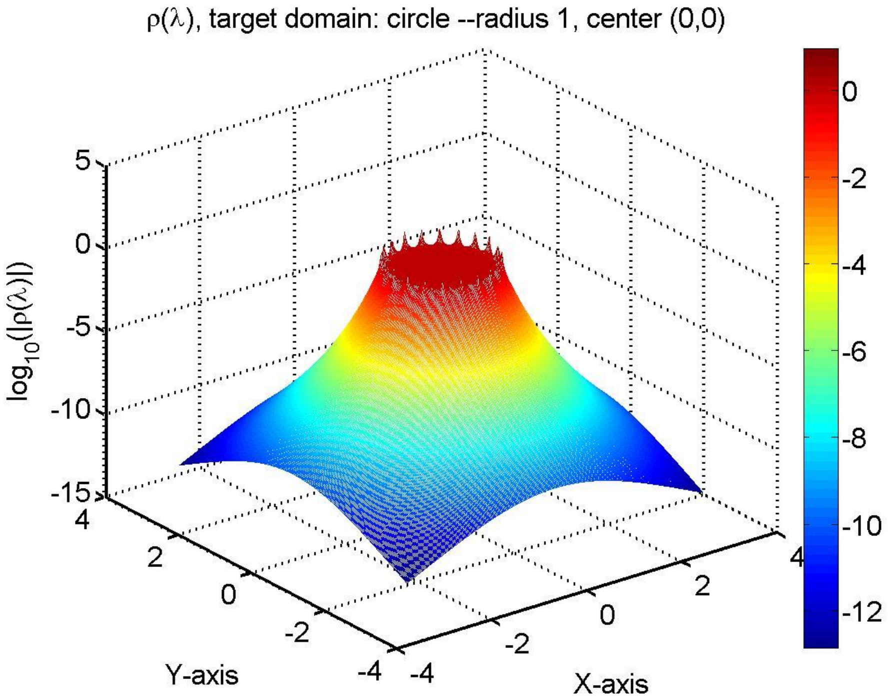
involves
solving q linear systems,
multiple RHS



$$\rho(M) \cdot Y = \sum_{k=1}^{q} w_k (z_k I - M)^{-1} \cdot Y$$

involves
solving q linear systems,
multiple RHS

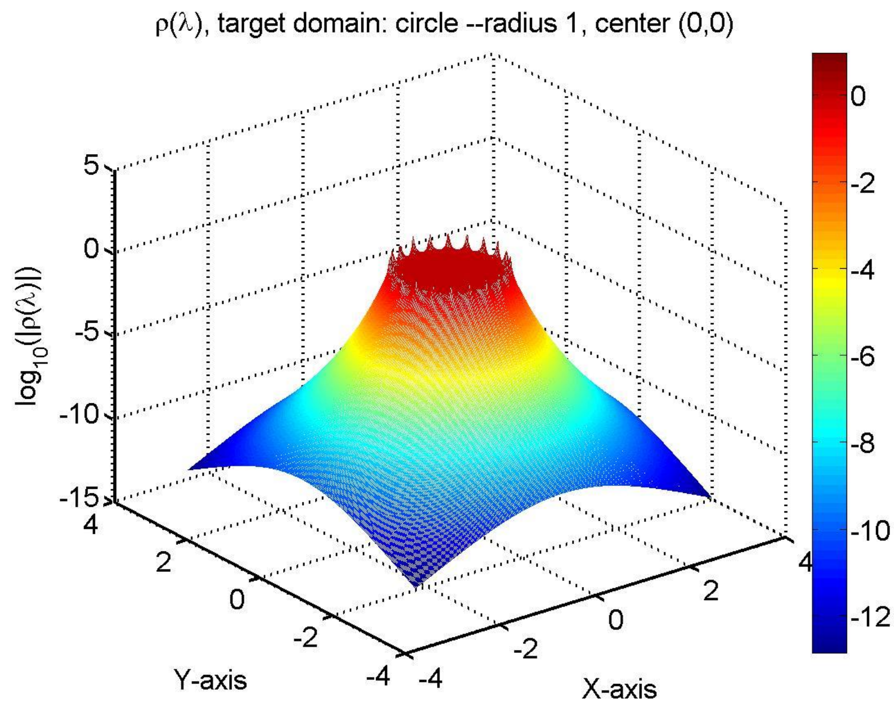
$$M = X \Lambda X^{-1} \Rightarrow \rho(M) = X \rho(\Lambda) X^{-1}$$

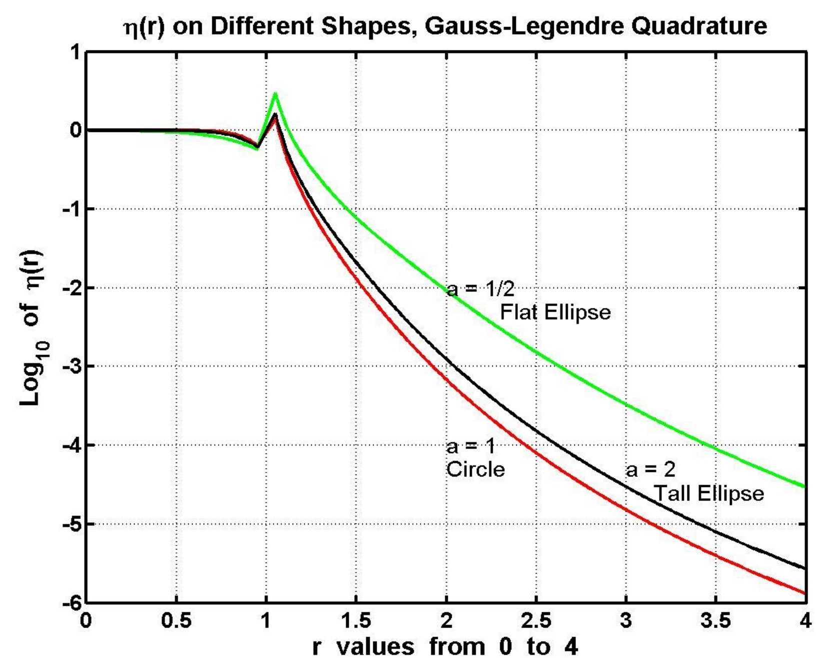


$$\rho(M) \cdot Y = \sum_{k=1}^{q} w_k (z_k I - M)^{-1} \cdot Y$$

involves
solving q linear systems,
multiple RHS

$$M = X\Lambda X^{-1} \Rightarrow \rho(M) = X\rho(\Lambda)X^{-1}$$

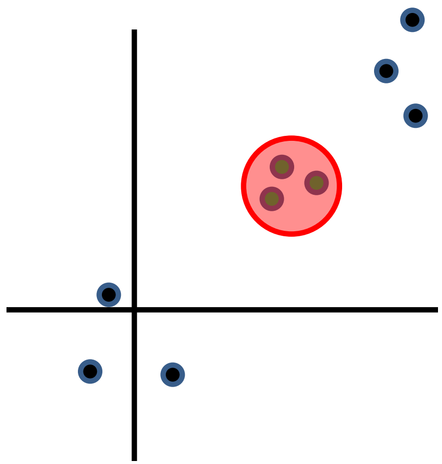




Another view of decay is to examine

$$\eta(r) = \begin{cases} \min_r |\rho(\Gamma \text{ at radius } r)| & r < 1 \\ \max_r |\rho(\Gamma \text{ at radius } r)| & r > 1 \end{cases}$$

Key Properties of FEAST



FEAST Algorithm:

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

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

$Y_k \leftarrow \rho(M)\, Q_{k-1}$

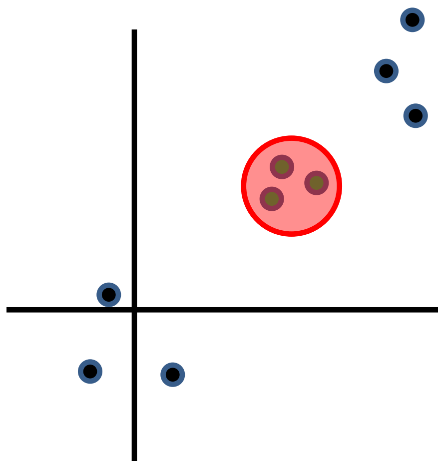
$Q_k \leftarrow \text{orthonormalize}(Y_k)$

$\hat{A} \leftarrow Q_k^T M Q_k$

Solve $\hat{A}\hat{W} = \hat{W}\hat{\Lambda}$ for $\hat{W}$ and $\hat{\Lambda}$

$Q_k \leftarrow Q_k \cdot \hat{W}$

End Loop:



FEAST Algorithm:

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

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

$Y_k \leftarrow \rho(M)\, Q_{k-1}$

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

$\hat{A} \leftarrow Q_k^T M Q_k$

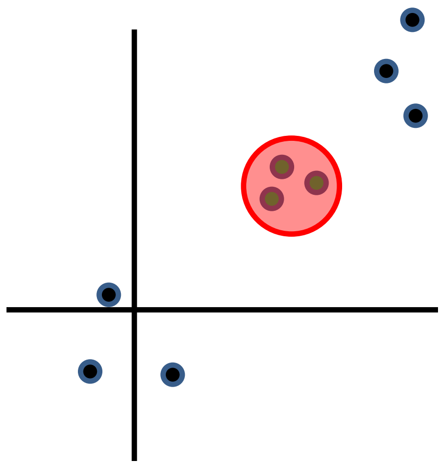
Solve $\hat{A}\hat{W} = \hat{W}\hat{\Lambda}$ for $\hat{W}$ and $\hat{\Lambda}$

$Q_k \leftarrow Q_k \cdot \hat{W}$

End Loop:

Under appropriate conditions, eigenpair (λ_j, x_j) converges at the rate of

$$|\rho(\lambda_{p+1})/\rho(\lambda_j)|^k$$



FEAST Algorithm:

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

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

$Y_k \leftarrow \rho(M)\, Q_{k-1}$

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

$\hat{A} \leftarrow Q_k^T M Q_k$

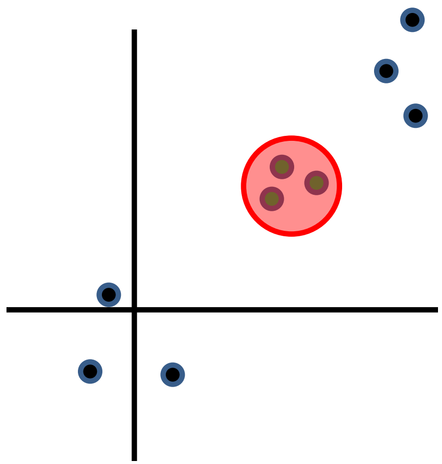
Solve $\hat{A}\hat{W} = \hat{W}\hat{\Lambda}$ for $\hat{W}$ and $\hat{\Lambda}$

$Q_k \leftarrow Q_k \cdot \hat{W}$

End Loop:

Recall that
$\rho(M) \cdot Q_k = \sum_{k=1}^{q} w_k (z_k I - M)^{-1} \cdot Q_k$

for eigenvalue problem
$Mx = \lambda x$



FEAST Algorithm: (works also for generalized problem)

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

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

$Y_k \leftarrow \rho(M)\, Q_{k-1}$

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

$\hat{A} \leftarrow Q_k^T M Q_k$

Solve $\hat{A}\hat{W} = \hat{W}\hat{\Lambda}$ for $\hat{W}$ and $\hat{\Lambda}$

$Q_k \leftarrow Q_k \cdot \hat{W}$

End Loop:

Recall that
$\rho(M) \cdot Q_k = \sum_{k=1}^{q} w_k (z_k I - M)^{-1} \cdot Q_k$

for eigenvalue problem
$Mx = \lambda x$

Note that
$$\rho(B^{-1} M) \cdot Q_k = \sum_{k=1}^{q} w_k (z_k I - B^{-1} M)^{-1} \cdot Q_k = \sum_{k=1}^{q} w_k (z_k B - M)^{-1} \cdot B \cdot Q_k$$

for generalized problem
$Mx = \lambda B x$

A Note on Projections (Rayleigh-Ritz)

For standard non-Hermitian problem $Ax = \lambda x$

- If we have Q_R, $\text{span}(Q_R) \approx \mathcal{K}_{\text{right}}$,
  then orthogonal projection $Q_R^T A\, Q_R$ is reasonable
  note $\text{span}(AQ_R) \approx \text{span}(Q_R)$
- If we also have Q_L, $\text{span}(Q_L) \approx \mathcal{K}_{\text{left}}$, $Q_L^T Q_R = I$,
  then oblique projection $Q_L^T A Q_R$ is resonable

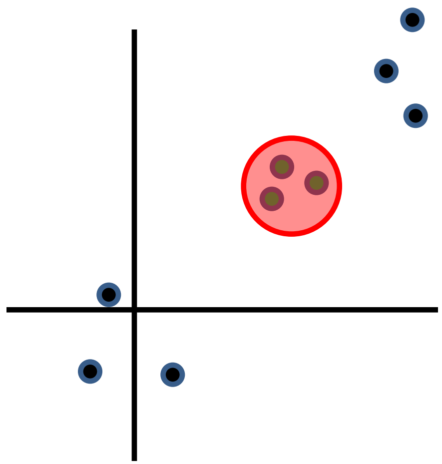
A Note on Projections (Rayleigh-Ritz)

For standard non-Hermitian generalized problem $Ax = \lambda Bx$

- If we have Q_R, span$(Q_R) \approx \mathcal{K}_{\text{right}}$,
  then orthogonal projection is $(Q_R^T A\, Q_R, Q_R^T B Q_R)$
  while reasonable but can fail in pathological cases

- Jin/Chan/Yeung propose using oblique projection
  $((BQ_R)^T AQ_R, (BQ_R)^T BQ_R)$
  note that span$(AQ_R) \approx$ span(BQ_R)

- If we have span$(Q_L) \approx \mathcal{K}_{\text{left}}$,
  then oblique projection $(Q_L^T AQ_R, Q_L^T BQ_R)$ is resonable



FEAST Algorithm:

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

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

$Y_k \leftarrow \rho(M)\, Q_{k-1}$

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

$\hat{A} \leftarrow Q_k^T M Q_k$

Solve $\hat{A}\hat{W} = \hat{W}\hat{\Lambda}$ for $\hat{W}$ and $\hat{\Lambda}$

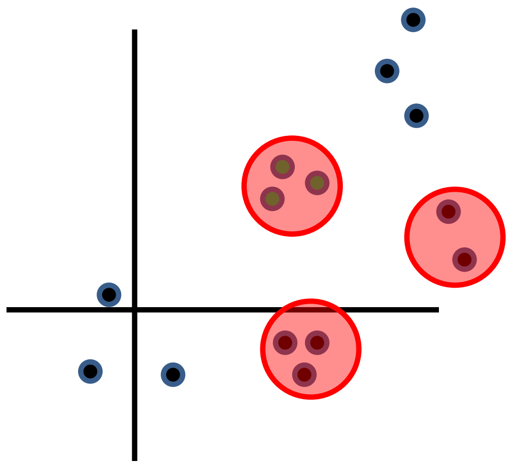
$Q_k \leftarrow Q_k \cdot \hat{W}$

End Loop:

Multiple Levels of Parallelism:

$$\rho(M) \cdot Q_k = \sum_{k=1}^{q} w_k (z_k I - M)^{-1} \cdot Q_k$$

q linear systems
each with p right-hand sides



FEAST Algorithm:

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

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

$Y_k \leftarrow \rho(M)\, Q_{k-1}$

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

$\hat{A} \leftarrow Q_k^T M Q_k$

Solve $\hat{A}\hat{W} = \hat{W}\hat{\Lambda}$ for $\hat{W}$ and $\hat{\Lambda}$

$Q_k \leftarrow Q_k \cdot \hat{W}$

End Loop:

Multiple Levels of Parallelism:

$\rho(M) \cdot Q_k = \sum_{k=1}^{q} w_k (z_k I - M)^{-1} \cdot Q_k$

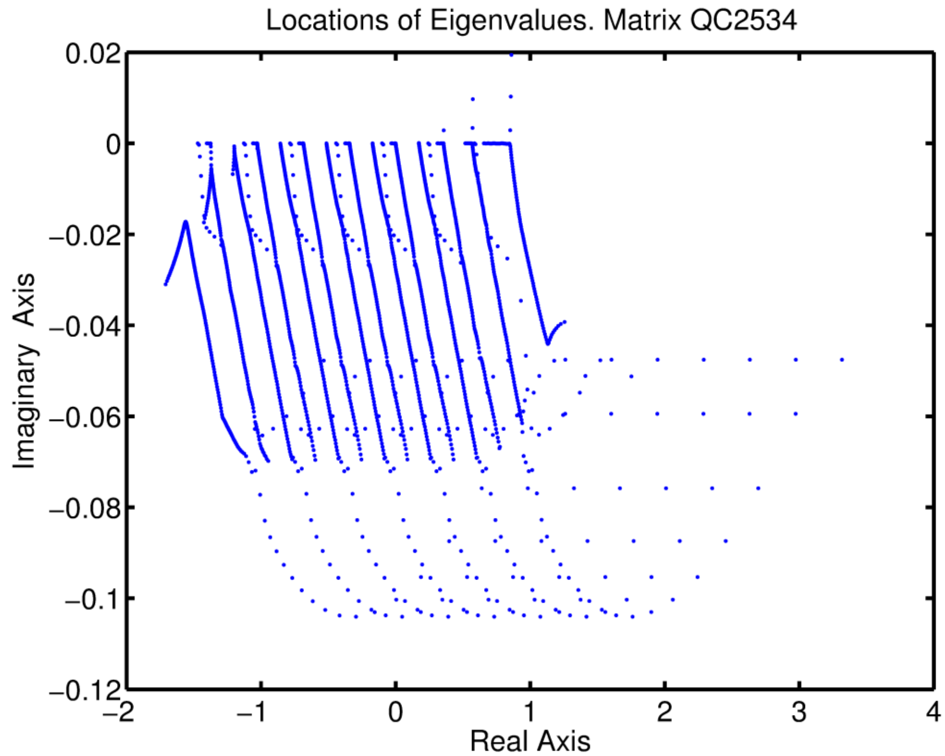
q linear systems
each with p right-hand sides

multiple independent regions

Illustrative Examples

Example 1

Circular region
center is $(0, -0.5)$
radius is 0.01
eigenvalues in region: 8
p set to 8
$\log_{10} |\rho(\lambda_{p+1}/\lambda_p|) = -0.93$

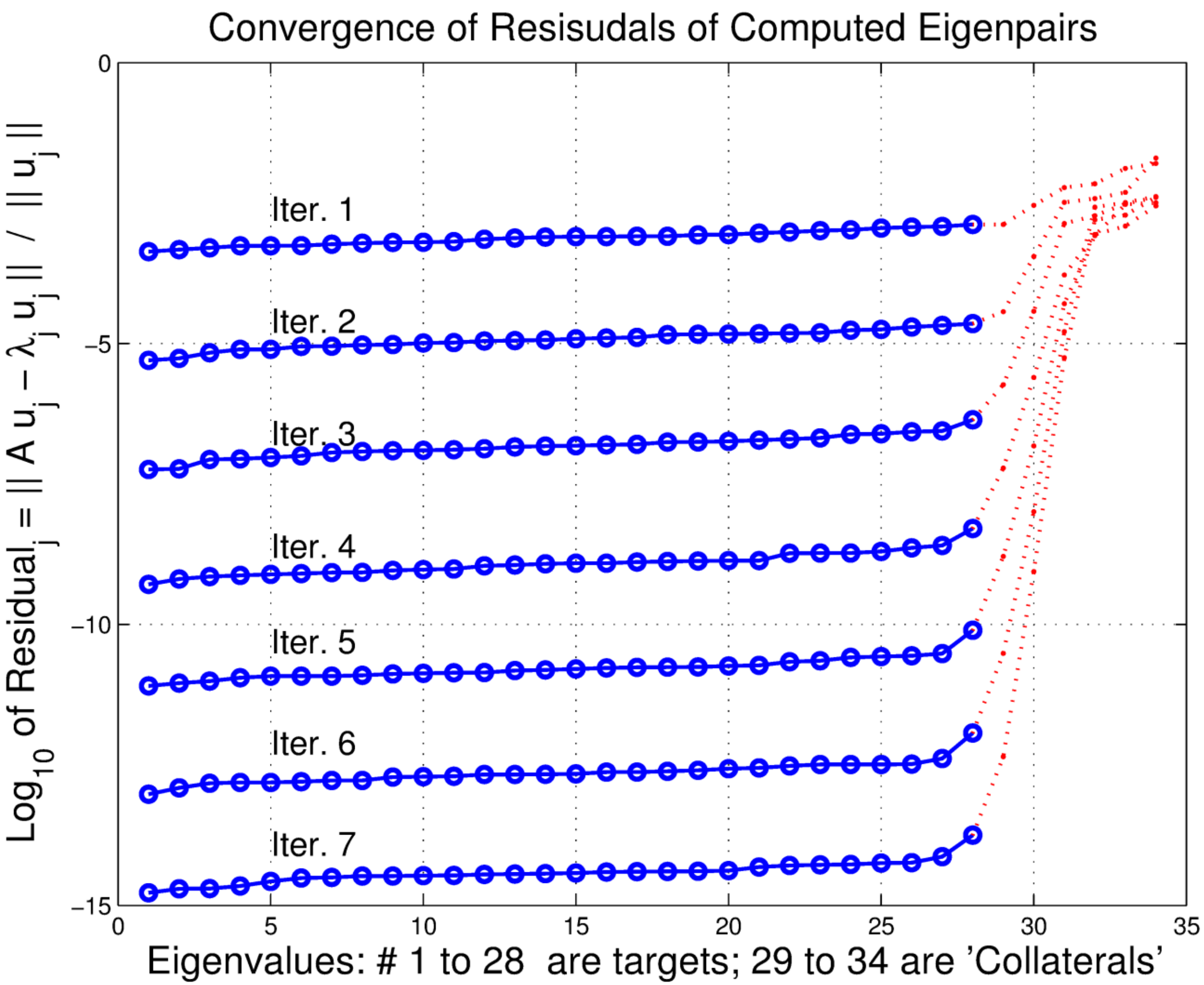


	Iterations						
	4	5	6	7	$\cdots$	13	14
$\max_j \log_{10}(\lvert\hat{\lambda}_j - \lambda_j\rvert)$	-5.4	-6.4	-7.3	-8.2	$\cdots$	-14.1	-14.7
$\max_j \log_{10}(\text{residual}_j)$	-5.3	-6.2	-7.1	-8.1	$\cdots$	-14.8	-14.5

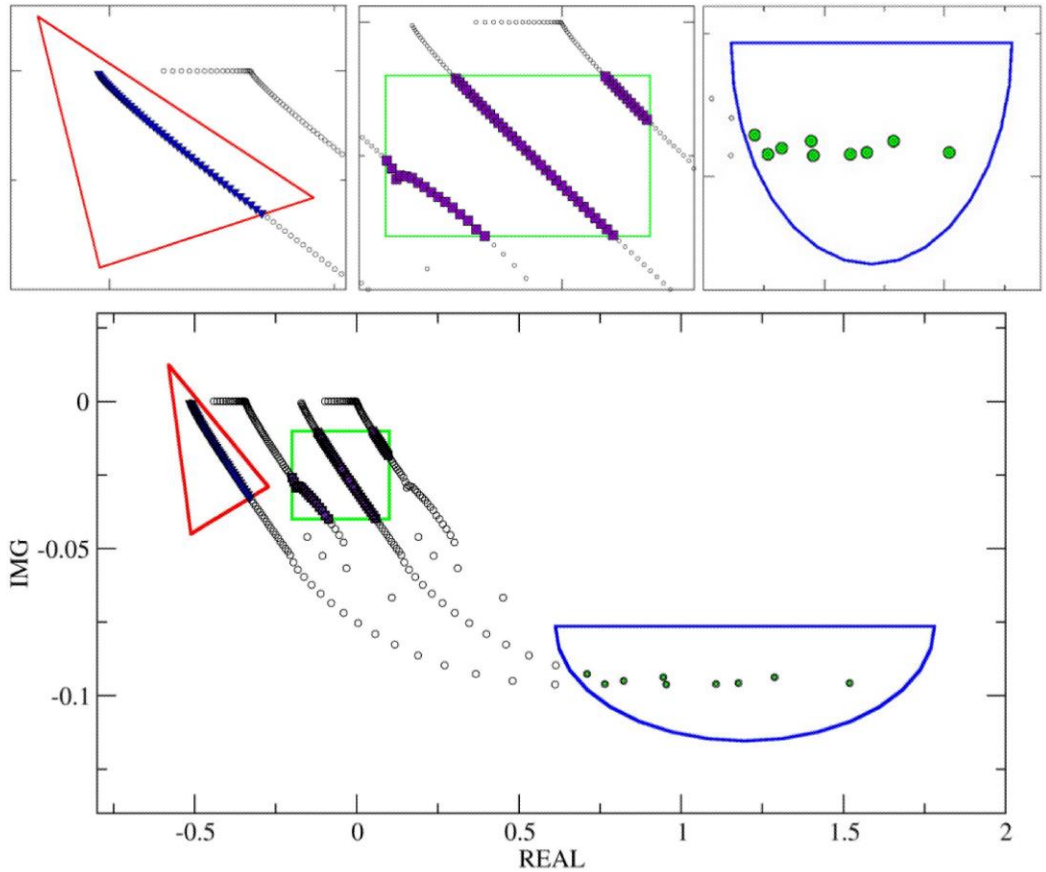
Example 2

Convergence of the 28 consistent with $|\rho(\lambda_{p+1})/\rho(\lambda_i)|$

Circular region
center is $(0, -0.17)$
radius is 0.02
28 eigenvalues in region
p set to $28 + 6 = 34$



Example 3



		Triangle	Square	Semicircle
# eigenvalues	m	45	64	9
subspace dim.	p	80	100	80
# quadrature nodes	K	24	32	16
convergence, measured in digits per iteration				
eigenvalues		4	2	5
residuals		2	1	2.5

An Application

An electronic structure application,
FEM discretization of the Kohn-Sham equation
using quadratic (FEM-Q) and cubic (FEM-C).

Size of matrix is n
Number of eigenvalues in region is m
Subspace dimension used is p

This example illustrate parallelism provided by
(1) sparse linear solver, and
(2) number of linear system solves, and

	FEM-Q $n = 149,192$ $m = 149$, $p = 300$				FEM-C $n = 1,165,485$ $m = 181$, $p = 300$			
Time (Ksec)	0.27	0.15	0.10	0.05	1.65	0.86	0.61	0.25
Eff. (%)	100	92	88	85	100	96	91	110
	1	2	3	6	1	2	3	6
	Number of Nodes, 2 MPI Process/Node							