Inner-outer iterative methods for large sparse Eigenvalue computations

Melina Freitag

Department of Mathematical Sciences
University of Bath

Informal Postgraduate Seminar
University of Bath
21st March 2006

Outline

What is an eigenvalue?

- eigenvalue comes from the German word Eigenwert (like liverwurst, only half of it has been translated).
- arises after simplification/discretisation/linearisation of a problem
- can be meaningless intermediate values of a computation method in order to find the solution of a problem
- sometimes the values have a meaning for the problem (stability analysis)

What is an eigenvalue?

- eigenvalue comes from the German word Eigenwert (like liverwurst, only half of it has been translated).
- arises after simplification/discretisation/linearisation of a problem
- can be meaningless intermediate values of a computation method in order to find the solution of a problem
- sometimes the values have a meaning for the problem (stability analysis)

Millenium Footbridge

On its opening day, the Millenium footbridge in London started to wobble under the weight of 100s of people, who, in turn also struggled to keep their balance. The bridge had to be closed. After fitting of 37 fluid-viscous dampers and 1 year and £5m later the problem was fixed.

Millenium Footbridge and some more applications

What had happened?

Some of the frequencies of the bridge were similar to the components of the pedestrians footsteps, causing vibration amplification. Finding these natural frequencies amounts to solving an eigenvalue problem.

- Structural dynamics
- Quantum Chemistry/Chemical Reactions
- Markov chain techniques/Google
- Stability analysis of dynamical systems/solution of differential equations

Outline

A few definitions

- Eigenvalues of $A \in \mathbb{R}^{n,n}$ are roots of $\det(A - \lambda I)$
- Eigenvectors are $x \neq 0$ such that $Ax = \lambda x$ for some eigenvalue $\lambda \in \mathbb{C}$
- Schur Decomposition: There exists an orthogonal matrix $Q \in \mathbb{C}^{n,n}$ $(Q^T Q = I)$ s.t.

$$Q^T A Q = \begin{bmatrix} d_{11} & \cdots & & d_{1n} \\ 0 & d_{11} & \cdots & d_{2n} \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & d_{nn} \end{bmatrix}$$

A few definitions

- Eigenvalues of $A \in \mathbb{R}^{n,n}$ are roots of $\det(A - \lambda I)$
- Eigenvectors are $x \neq 0$ such that $Ax = \lambda x$ for some eigenvalue $\lambda \in \mathbb{C}$
- Schur Decomposition: There exists an orthogonal matrix $Q \in \mathbb{C}^{n,n}$ $(Q^T Q = I)$ s.t.

$$Q^T A Q = \begin{bmatrix} d_{11} & \cdots & & d_{1n} \\ 0 & d_{11} & \cdots & d_{2n} \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & d_{nn} \end{bmatrix}$$

Outline

Why not calculating the roots of the characteristic polynomial?

- $$\det(A) = \sum_{\sigma \in S_n} \left(\mathrm{sgn}(\sigma) \prod_{i=1}^{n} a_{i,\sigma(i)} \right)$$

  $\sigma \ldots$, permutation
- contains $n!$ summands, not very handy
- there exists no formula for calculation the roots of a polynomial of degree larger then 5
- calculating the roots numerically is unstable
- Back to matrices!

Why not calculating the roots of the characteristic polynomial?

- $$\det(A) = \sum_{\sigma \in S_n} \left(\operatorname{sgn}(\sigma) \prod_{i=1}^{n} a_{i,\sigma(i)} \right)$$

  $\sigma \ldots$, permutation
- contains $n!$ summands, not very handy
- there exists no formula for calculation the roots of a polynomial of degree larger then 5
- calculating the roots numerically is unstable
- Back to matrices!

Why not calculating the roots of the characteristic polynomial?

- $$\det(A) = \sum_{\sigma \in S_n} \left(\operatorname{sgn}(\sigma) \prod_{i=1}^{n} a_{i,\sigma(i)} \right)$$
  $\sigma \ldots,$ permutation
- contains $n!$ summands, not very handy
- there exists no formula for calculation the roots of a polynomial of degree larger then 5
- calculating the roots numerically is unstable
- Back to matrices!

Why not calculating the roots of the characteristic polynomial?

- $$\det(A) = \sum_{\sigma \in S_n} \left(\mathrm{sgn}(\sigma) \prod_{i=1}^{n} a_{i,\sigma(i)} \right)$$

  $\sigma \ldots$, permutation
- contains $n!$ summands, not very handy
- there exists no formula for calculation the roots of a polynomial of degree larger then 5
- calculating the roots numerically is unstable
- Back to matrices!

Why not calculating the roots of the characteristic polynomial?

- $$\det(A) = \sum_{\sigma \in S_n} \left(\operatorname{sgn}(\sigma) \prod_{i=1}^{n} a_{i,\sigma(i)} \right)$$

  $\sigma \ldots$, permutation
- contains $n!$ summands, not very handy
- there exists no formula for calculation the roots of a polynomial of degree larger then 5
- calculating the roots numerically is unstable
- Back to matrices!

Outline

Algorithms based on the idea of calculating the Schur Form

- Schur Form:

$$Q^TAQ = \begin{bmatrix} d_{11} & \cdots & & d_{1n} \\ 0 & d_{11} & \cdots & d_{2n} \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & d_{nn} \end{bmatrix}$$

- Similarity transformation does not change the eigenvalues:

$$\begin{aligned} \det(Q^TAQ - \lambda I) &= \det(Q^TAQ - Q^T\lambda Q) \\ &= \det(Q^T)\det(A - \lambda I)\det(Q) \\ &= \det(A - \lambda I) \end{aligned}$$

- QR Algorithm

Algorithms based on the idea of calculating the Schur Form

- Schur Form:

$$Q^TAQ = \begin{bmatrix} d_{11} & \cdots & & d_{1n} \\ 0 & d_{11} & \cdots & d_{2n} \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & d_{nn} \end{bmatrix}$$

- Similarity transformation does not change the eigenvalues:

$$\begin{aligned} \det(Q^TAQ - \lambda I) &= \det(Q^TAQ - Q^T\lambda Q) \\ &= \det(Q^T)\det(A - \lambda I)\det(Q) \\ &= \det(A - \lambda I) \end{aligned}$$

- QR Algorithm

Algorithms based on the idea of calculating the Schur Form

- Schur Form:

$$Q^T AQ = \begin{bmatrix} d_{11} & \cdots & & d_{1n} \\ 0 & d_{11} & \cdots & d_{2n} \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & d_{nn} \end{bmatrix}$$

- Similarity transformation does not change the eigenvalues:

$$\begin{aligned} \det(Q^T AQ - \lambda I) &= \det(Q^T AQ - Q^T \lambda Q) \\ &= \det(Q^T)\det(A - \lambda I)\det(Q) \\ &= \det(A - \lambda I) \end{aligned}$$

- QR Algorithm

QR-Algorithm I

- Basic QR-iteration: given $A_0 := A$ compute

$$\begin{aligned} \text{Factor} \quad A_i &= Q_i R_i \quad \text{(QR decomposition)} \\ A_{i+1} &= R_i Q_i \end{aligned}$$

- A_i and A have the same eigenvalues and we hope that

$$A_i \to R = \begin{bmatrix} d_{11} & \cdots & & d_{1n} \\ 0 & d_{11} & \cdots & d_{2n} \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & d_{nn} \end{bmatrix}$$

- Work for calculating all the eigenvalues and eigenvectors of a matrix in $\mathcal{O}(n^3)$ operations

QR-Algorithm I

- Basic QR-iteration: given $A_0 := A$ compute

$$\begin{aligned} \text{Factor} \quad A_i &= Q_i R_i \quad \text{(QR decomposition)} \\ A_{i+1} &= R_i Q_i \end{aligned}$$

- A_i and A have the same eigenvalues and we hope that

$$A_i \to R = \begin{bmatrix} d_{11} & \cdots & & d_{1n} \\ 0 & d_{11} & \cdots & d_{2n} \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & d_{nn} \end{bmatrix}$$

- Work for calculating all the eigenvalues and eigenvectors of a matrix in $\mathcal{O}(n^3)$ operations

QR-Algorithm I

- Basic QR-iteration: given $A_0 := A$ compute

$$\begin{aligned} \text{Factor} \quad A_i &= Q_i R_i \quad \text{(QR decomposition)} \\ A_{i+1} &= R_i Q_i \end{aligned}$$

- A_i and A have the same eigenvalues and we hope that

$$A_i \rightarrow R = \begin{bmatrix} d_{11} & \cdots & & d_{1n} \\ 0 & d_{11} & \cdots & d_{2n} \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \cdots & d_{nn} \end{bmatrix}$$

- Work for calculating all the eigenvalues and eigenvectors of a matrix in $\mathcal{O}(n^3)$ operations

QR-Algorithm II

Problem Fill-in for large sparse matrices

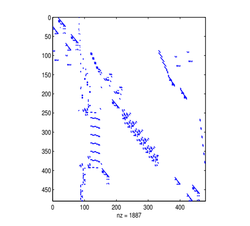


Figure: Sparse matrix

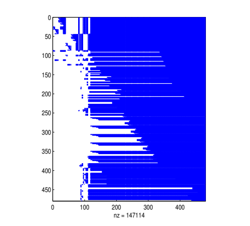


Figure: One step of QR

Outline

Large sparse matrices

- Only a few eigenvalues (largest, smallest) are required
- QR algorithm is too expensive in terms of storage and computation time
- need iterative methods!
- Examples: Power method, Inverse iteration, Rayleigh Quotient Iteration, Subspace iteration, Lanczos method, Arnoldi's method

Large sparse matrices

- Only a few eigenvalues (largest, smallest) are required
- QR algorithm is too expensive in terms of storage and computation time
- need iterative methods!
- Examples: Power method, Inverse iteration, Rayleigh Quotient Iteration, Subspace iteration, Lanczos method, Arnoldi's method

Large sparse matrices

- Only a few eigenvalues (largest, smallest) are required
- QR algorithm is too expensive in terms of storage and computation time
- need iterative methods!
- Examples: Power method, Inverse iteration, Rayleigh Quotient Iteration, Subspace iteration, Lanczos method, Arnoldi's method

Concepts of the Power method

- Given x_0

$$
\begin{aligned}
y_{i+1} &= Ax_i \\
x_{i+1} &= y_{i+1}/\|y_{i+1}\| \\
\lambda_{i+1} &= x_{i+1}^T A x_{i+1}
\end{aligned}
$$

- convergence to λ_1 with linear rate $\frac{|\lambda_2|}{|\lambda_1|} < 1$
- only needs one matrix-vector multiplication at each step
- Inverse Iteration is power method applied to $(A - \sigma I)^{-1}$
- Rayleigh Quotient iteration with special shift $\sigma_{i+1} = x_{i+1}^T A x_{i+1}$ (quadratic/cubic convergence)

Concepts of the Power method

- Given x_0

$$\begin{aligned} y_{i+1} &= Ax_i \\ x_{i+1} &= y_{i+1}/\|y_{i+1}\| \\ \lambda_{i+1} &= x_{i+1}^T A x_{i+1} \end{aligned}$$

- convergence to λ_1 with linear rate $\dfrac{|\lambda_2|}{|\lambda_1|} < 1$
- only needs one matrix-vector multiplication at each step
- Inverse Iteration is power method applied to $(A - \sigma I)^{-1}$
- Rayleigh Quotient iteration with special shift $\sigma_{i+1} = x_{i+1}^T A x_{i+1}$ (quadratic/cubic convergence)

Concepts of the Power method

- Given x_0

$$\begin{aligned} y_{i+1} &= Ax_i \\ x_{i+1} &= y_{i+1}/\|y_{i+1}\| \\ \lambda_{i+1} &= x_{i+1}^T A x_{i+1} \end{aligned}$$

- convergence to λ_1 with linear rate $\frac{|\lambda_2|}{|\lambda_1|} < 1$
- only needs one matrix-vector multiplication at each step
- Inverse Iteration is power method applied to $(A - \sigma I)^{-1}$
- Rayleigh Quotient iteration with special shift $\sigma_{i+1} = x_{i+1}^T A x_{i+1}$ (quadratic/cubic convergence)

Concepts of the Power method

- Given x_0

$$\begin{aligned} y_{i+1} &= Ax_i \\ x_{i+1} &= y_{i+1}/\|y_{i+1}\| \\ \lambda_{i+1} &= x_{i+1}^T A x_{i+1} \end{aligned}$$

- convergence to λ_1 with linear rate $\frac{|\lambda_2|}{|\lambda_1|} < 1$
- only needs one matrix-vector multiplication at each step
- Inverse Iteration is power method applied to $(A - \sigma I)^{-1}$
- Rayleigh Quotient iteration with special shift $\sigma_{i+1} = x_{i+1}^T A x_{i+1}$ (quadratic/cubic convergence)

Concepts of the Power method

- Given x_0

$$\begin{aligned} y_{i+1} &= Ax_i \\ x_{i+1} &= y_{i+1}/\|y_{i+1}\| \\ \lambda_{i+1} &= x_{i+1}^T A x_{i+1} \end{aligned}$$

- convergence to λ_1 with linear rate $\frac{|\lambda_2|}{|\lambda_1|} < 1$
- only needs one matrix-vector multiplication at each step
- Inverse Iteration is power method applied to $(A - \sigma I)^{-1}$
- Rayleigh Quotient iteration with special shift $\sigma_{i+1} = x_{i+1}^T A x_{i+1}$ (quadratic/cubic convergence)

Concepts of Lanczos/ Arnoldi's method I

- Power method with initial vector q computes $q, Aq, \dots, A^k q$
- idea of Arnoldi's method: retain past information: after k steps we have $k+1$ vectors $q, Aq, \dots, A^k q$
- Given q_1, $\|q_1\|_2 = 1$ On subsequent steps $k = 1, 2, \dots, m$ take

$$\tilde{q}_{k+1} = Aq_k - \sum_{j=1}^{k} q_j h_{jk}$$

where h_{jk} is the Gram-Schmidt coefficient $h_{jk} = < Aq_k, q_j >$. Normalise

$$q_{k+1} = \frac{\tilde{q}_{k+1}}{h_{k+1,k}} \quad \text{where} \quad h_{k+1,k} = \|\tilde{q}_{k+1}\|_2$$

Concepts of Lanczos/ Arnoldi's method I

- Power method with initial vector q computes $q, Aq, \ldots, A^k q$
- idea of Arnoldi's method: retain past information: after k steps we have $k+1$ vectors $q, Aq, \ldots, A^k q$
- Given q_1, $\|q_1\|_2 = 1$ On subsequent steps $k = 1, 2, \ldots, m$ take

$$\tilde{q}_{k+1} = Aq_k - \sum_{j=1}^{k} q_j h_{jk}$$

where h_{jk} is the Gram-Schmidt coefficient $h_{jk} =< Aq_k, q_j >$. Normalise

$$q_{k+1} = \frac{\tilde{q}_{k+1}}{h_{k+1,k}} \quad \text{where} \quad h_{k+1,k} = \|\tilde{q}_{k+1}\|_2$$

Concepts of Lanczos/ Arnoldi's method II

Definition

For any j the space span$\{q, Aq, \ldots, A^{j-1}q\}$ is called the jth Krylov subspace associated with A and q and is denoted by $\mathcal{K}_j(A, q)$.

Matrix representation

The Arnoldi process can be written in the form

$$AQ_m = Q_m H_m + q_{m+1} h_{m+1,m} e_m^T$$

where H_m is square upper Hessenberg.

Concepts of Lanczos/ Arnoldi's method III

- For the Lanczos process: H_m is tridiagonal (three term recurrence)
- Approximate eigenvalues and eigenvectors of A can be found from eigenvalues and eigenvectors of much smaller matrix H_m:

Let Q_m, H_m and $h_{m+1,m}$ be generated by the Arnoldi process. Let μ be an eigenvalue of H_m with associated eigenvector x normalised so that $\|x\|_2=1$. Let $y = Q_m x \in \mathbb{C}^n$. Then

$$\|Ay - \mu y\|_2 = |h_{m+1,m}||x_m|,$$

where x_m denotes the mth (and last) component of x.

Concepts of Lanczos/ Arnoldi's method III

- For the Lanczos process: H_m is tridiagonal (three term recurrence)
- Approximate eigenvalues and eigenvectors of A can be found from eigenvalues and eigenvectors of much smaller matrix H_m:

Let Q_m, H_m and $h_{m+1,m}$ be generated by the Arnoldi process. Let μ be an eigenvalue of H_m with associated eigenvector x normalised so that $\|x\|_2=1$. Let $y = Q_m x \in \mathbb{C}^n$. Then

$$\|Ay - \mu y\|_2 = |h_{m+1,m}||x_m|,$$

where x_m denotes the mth (and last) component of x.

Concepts of Lanczos/ Arnoldi's method III

- For the Lanczos process: H_m is tridiagonal (three term recurrence)
- Approximate eigenvalues and eigenvectors of A can be found from eigenvalues and eigenvectors of much smaller matrix H_m:

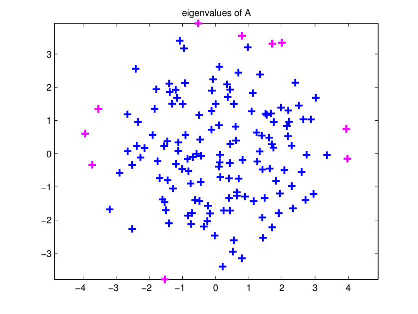
Let Q_m, H_m and $h_{m+1,m}$ be generated by the Arnoldi process. Let μ be an eigenvalue of H_m with associated eigenvector x normalised so that $\|x\|_2=1$. Let $y = Q_m x \in \mathbb{C}^n$. Then

$$\|Ay - \mu y\|_2 = |h_{m+1,m}||x_m|,$$

where x_m denotes the mth (and last) component of x.

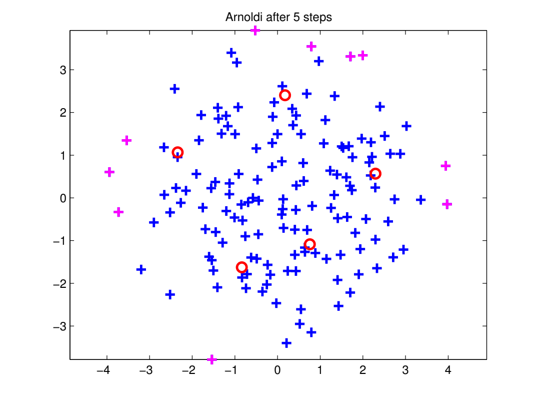
Example

random complex matrix of dimension $n = 144$ generated in MATLAB:

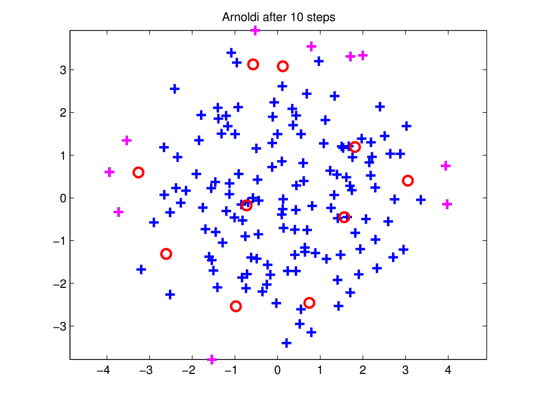


approximation of outer eigenvalues first!

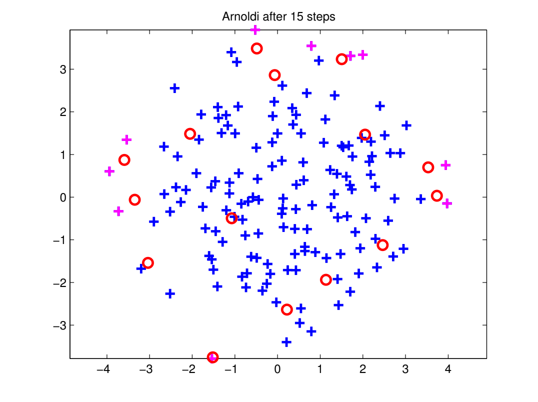
after 5 Arnoldi steps



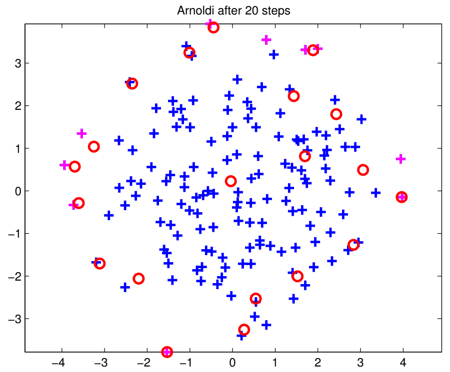
after 10 Arnoldi steps



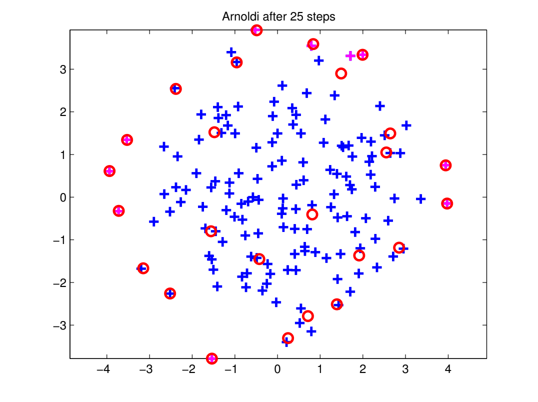
after 15 Arnoldi steps



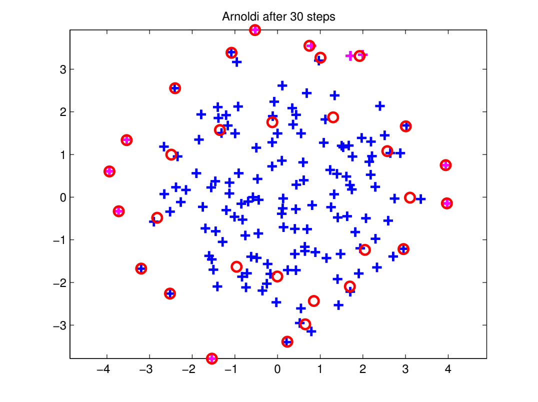
after 20 Arnoldi steps



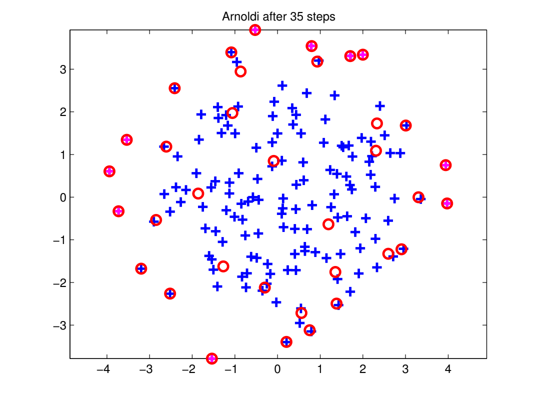
after 25 Arnoldi steps



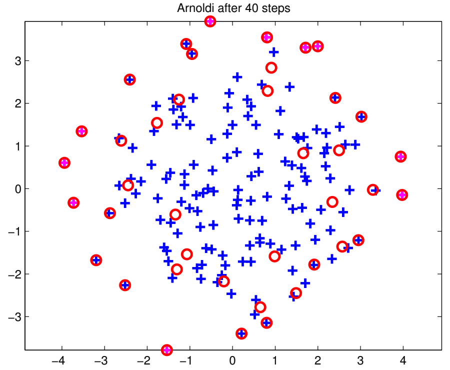
after 30 Arnoldi steps



after 35 Arnoldi steps



after 40 Arnoldi steps



Outline

Large sparse matrices

- interior eigenvalues: Shift-Invert Transformation
- $$\begin{aligned} Ax &= \lambda x \\ (A - \sigma I)^{-1} x &= \frac{1}{\lambda - \sigma} x \end{aligned}$$
- "outer" eigenvalues are the one closest to σ
- Problem: requires a solution of a linear system at each step:
  $$(A - \sigma I)u = b$$
- Solve is done iteratively (since A is large and sparse) using MINRES, GMRES, other iterative methods

Large sparse matrices

- interior eigenvalues: Shift-Invert Transformation
- $$\begin{aligned} Ax &= \lambda x \\ (A-\sigma I)^{-1}x &= \frac{1}{\lambda-\sigma}x \end{aligned}$$
- "outer" eigenvalues are the one closest to σ
- Problem: requires a solution of a linear system at each step:
$$(A-\sigma I)u = b$$
- Solve is done iteratively (since A is large and sparse) using MINRES, GMRES, other iterative methods

Large sparse matrices

- The solve will be inexact
- How does the inexact inner solve influence the convergence property of the outer solve?
- For inexact inverse iteration:

$$(A - \sigma I)y_i = x_i + \text{res}_i$$

If residual res_i is chosen to decrease in the same manner as the eigenvalue residual decreases the convergence rate from exact solves is recovered

- Ongoing research: Situation for Krylov methods

Large sparse matrices

- The solve will be inexact
- How does the inexact inner solve influence the convergence property of the outer solve?
- For inexact inverse iteration:

$$(A - \sigma I)y_i = x_i + \mathrm{res}_i$$

  If residual res_i is chosen to decrease in the same manner as the eigenvalue residual decreases the convergence rate from exact solves is recovered
- Ongoing research: Situation for Krylov methods