EigenSolver

- Power Iteration
- Inverse Iteration
- QR Iteration
- Lanczos Iteration
- Arnoldi Iteration

- Assumption : real symmetric matrix
- ->real eigen values

  & invertible (Inverse Iteration Method)

  & symmetric (Lanczos)

Power Iteration Method

- - Goal: Compute the largest magnitude eigenvalue and its eigenvector.
- - Start with a random vector b_0.
- - Iterate: $b_{k+1} = A\ b_k$, then normalize.
- - Stop when $||b_{k+1} - b_k|| < \text{tolerance}$.
- - Estimate $\lambda \approx |Ab_k| / |b_k|$ using Rayleigh quotient.
- - Converges to the dominant eigenvalue (largest in magnitude).

Inverse Iteration Method

- \- Goal: Compute an eigenvalue close to a guess u (default: 0).
- \- Start with a random vector b_0.
- \- LU Decomposition: A = LU (precomputed).
- \- Iterate: Solve LU $b_{k+1} = b_k \rightarrow b_{k+1} \approx A^{-1} b_k$.

  if shift: $b_{k+1} \approx (A-u\,I)^{-1} b_k$
- \- Normalize x and check $|| b_{k+1} - b_k|| <$ tolerance.
- \- Estimate $\lambda \approx |Ab_k| / |b_k|$ using Rayleigh quotient.
- \- Finds eigenvalue closest to initial guess (or 0 if no shift).

QR Iteration Method

- Goal: compute all eigenvalues of a square matrix based on repeated QR decompositions and similarity transforms
- Given a square matrix $A_0 = A$:
- 1. Compute QR decomposition: $A_k = Q_k R_k$
- 2. Construct next iterate: $A_{k+1} = R_k Q_k$
- 3. Repeat until A_k becomes nearly diagonal or upper-triangular.

Why Does It Work?

- - $A_{k+1} = Q_k^T A_k Q_k$ is a similarity transform.
- - Similar matrices share eigenvalues.
- - As $k \to \infty$, $A_k \to$ diagonal form.
- - Diagonal elements ≈ eigenvalues.

Lanczos Iteration Method

- - Goal: approximate eigenvalues of symmetric matrix $A \in \mathbb{R}^{n\text{x}n}$
  - Builds orthonormal basis Q_k for Krylov subspace:
    $K_k(A, q_0) = \text{span}\{q_0, Aq_0, A^2q_0, \ldots, A^{k-1}q_0\}$
  - Projects A to low-dim tridiagonal matrix:
    $T_k = Q_k^T A Q_k \in \mathbb{R}^{k\text{x}k}$
  - $A \approx Q_k T_k Q_k^T$
  - $\text{Eig}(T_k) \approx$ partial eig(A)
- Efficient for large sparse symmetric matrices.

Lanczos Iteration Method

- \- $Q_k = [q_0, q_1, \ldots, q_{k-1}]$ ($n \times k$ matrix with orthonormal cols)
  \- T_k = tridiagonal matrix with α_j on diag, β_j on off-diag
  \- Then:
  $A \approx Q_k T_k Q_k^T \Rightarrow A Q_k = Q_k T_k$

$$T_k = \begin{bmatrix} \alpha_1 & \beta_1 & 0 & \cdots \\ \beta_1 & \alpha_2 & \beta_2 & \cdots \\ 0 & \beta_2 & \alpha_3 & \cdots \\ \vdots & \vdots & \ddots & \ddots \end{bmatrix}$$

- Main iteration:
  $A q_j = \beta_{j-1} q_{j-1} + \alpha_j q_j + \beta_j q_{j+1}$
  Implementation:
  $w = A q_j - \beta_{j-1} q_{j-1}$
  $\alpha_j = q_j^T w$
  $w \leftarrow w - \alpha_j q_j$
  $\beta_j = ||w||$, then $q_{j+1} = w / \beta_j$
  Yields: tridiagonal T with α on diag, β on off-diagonals
- Solve eigenvalue by QR Iteration on T

Arnoldi Iteration Method

- \- Goal: approximate eigenvalues of a general (non-symmetric) matrix $A \in \mathbb{R}^{nxn}$
  \- Builds orthonormal basis Q_k for Krylov subspace:
  $K_k(A, q_0) = \text{span}\{q_0, Aq_0, A^2q_0, \ldots, A^{k-1}q_0\}$
  \- Projects A to low-dim upper Hessenberg matrix:
  $H_k = Q_k^T A Q_k \in \mathbb{R}^{kxk}$
  \- $A \approx Q_k H_k Q_k^T$
  \- $\text{Eig}(H_k) \approx$ partial eig(A)

Arnoldi Iteration Method

- \- $Q_k = [q_0, q_1, ..., q_{k-1}]$ ($n \times k$ matrix with orthonormal cols)
  \- H_k = upper Hessenberg matrix (zero below subdiagonal)
  \- Then: $A \approx Q_k H_k Q_k^T \Rightarrow A Q_k = Q_k H_k$
- Main iteration:
  $w = A q_j$
  for $i = 0$ to j:
  $h_{ij} = q_i^T w$
  $w \leftarrow w - h_{ij} q_i$
  $h_{i+1,j} = ||w||$, $q_{i+1} = w / h_{i+1,j}$
- Solve eigenvalue by QR Iteration on H

Eigen Solver Summary

Method	Requirement	Return
Power Iteration	-	max magnitude eigen value
Inverse Iteration	invertible	eigen value close to a given value
QR Iteration	-	all eigen values
Lanczos Iteration	real symmetric	k eigen values
Arnoldi Iteration	-	k eigen values

Test case on Symmetric Positive Definite Matrix

Eigen Solver Test Case

- Generate a spd matrix of 50, max_iter = 50, num_eigenvalues=10

- [Testing Power Iteration]
- Finish iteration in iter 6
- Estimated dominant eigenvalue (Power Iteration): 792.356
- Time taken: 0 ms

- [Testing Inverse Iteration]
- Finish iteration in iter 3
- Estimated smallest eigenvalue (Inverse Iteration): 5.25056e-08
- Time taken: 145 ms

- [Testing QR Iteration]
- [QR finished]
- Time taken: 19478 ms
- [QR eigenvalues (first 10)]
- 792.356 12.8577 11.9656 10.7768 9.66794 8.73318 8.41568 7.88438 7.15266 7.14658

- [Testing Lanczos Iteration]
- [Lanczos finished]
- Time taken: 13 ms
- [Lanczos eigenvalues (first 10)]
- 792.356 792.355 12.778 11.4522 9.91066 7.30989 4.48347 2.7544 1.04209 0.133806

- [Testing Arnoldi Iteration]
- [Arnoldi finished]
- Time taken: 13 ms
- [Arnoldi eigenvalues (first 10)]
- 792.356 12.8503 11.6539 10.3237 8.81115 6.48019 3.72511 2.11372 0.924672 0.123859