Numerical Computing

Lecture 8: Eigenvalue Problems

Francisco Richter Mendoza

Università della Svizzera Italiana
Faculty of Informatics, Lugano, Switzerland

QR Algorithm: Off-diagonal Decay

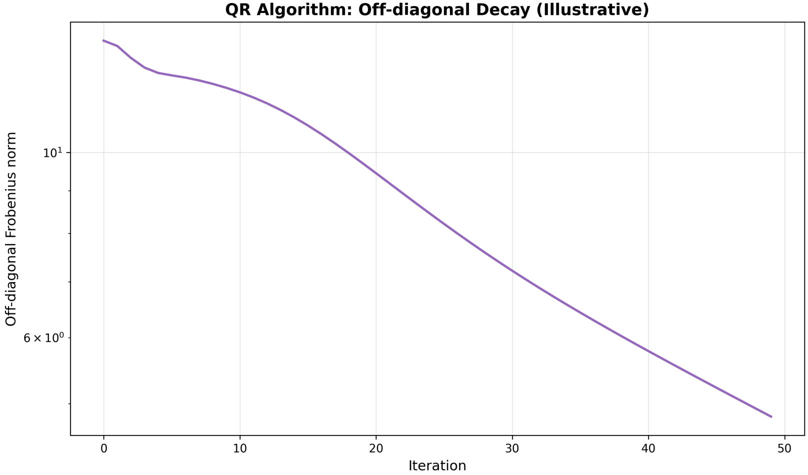

Figure: Illustrative off-diagonal Frobenius norm decay under unshifted QR iterations.

Eigenvalue Perturbation (Bauer–Fike)

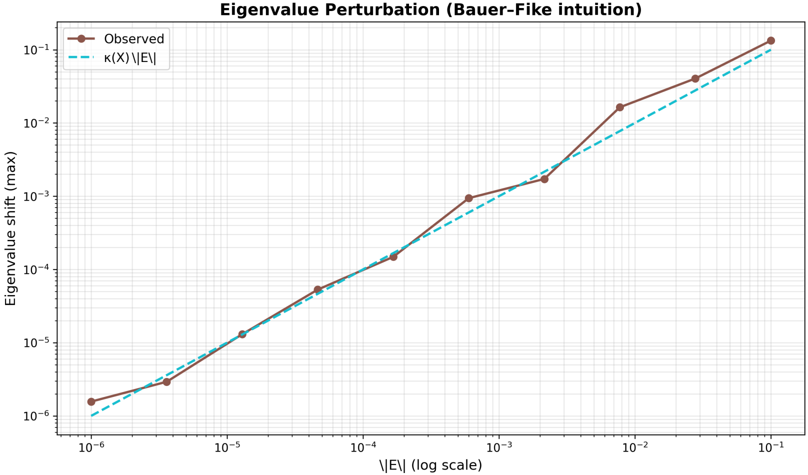

Figure: Observed eigenvalue shifts versus perturbation norm compared with $\kappa(X)\,\|E\|$ bound.

Lecture Overview

- **Power Iteration and Variants**: Power, Inverse, Rayleigh Quotient
- **QR Algorithm**: Shifts and Hessenberg reduction
- **Symmetric Problems**: Tridiagonal reduction, Wilkinson shift
- **Perturbation Theory**: Weyl, Bauer–Fike, eigenvector sensitivity
- **Large-Scale Methods**: Lanczos (symmetric), Arnoldi (nonsymmetric)
- **Practical Considerations**: Convergence, complexity, accuracy

Power Iteration

Basic Idea

For $Ax = \lambda x$, iterate $x^{(k+1)} = \dfrac{Ax^{(k)}}{\|Ax^{(k)}\|}$ to converge to the dominant eigenvector.

$$\text{Rate} \sim \left|\frac{\lambda_2}{\lambda_1}\right|^k, \quad |\lambda_1| > |\lambda_2| \geq \cdots \geq |\lambda_n|. \tag{1}$$

Variants: Inverse iteration, Rayleigh quotient iteration (cubic for symmetric matrices)

Classical Iterative Methods

Let $A = D - L - U$ where:

- D: diagonal part
- L: strictly lower triangular
- U: strictly upper triangular

Method Definitions

$$
\begin{align}
\text{Jacobi:} \quad x^{(k+1)} &= D^{-1}(L+U)x^{(k)} + D^{-1}b \tag{2}\\
\text{Gauss-Seidel:} \quad x^{(k+1)} &= (D-L)^{-1}Ux^{(k)} + (D-L)^{-1}b \tag{3}\\
\text{SOR:} \quad x^{(k+1)} &= (D-\omega L)^{-1}[(1-\omega)D + \omega U]x^{(k)} \tag{4}\\
&\quad + \omega(D-\omega L)^{-1}b \tag{5}
\end{align}
$$

Eigenvalue Methods: Convergence and Geometry

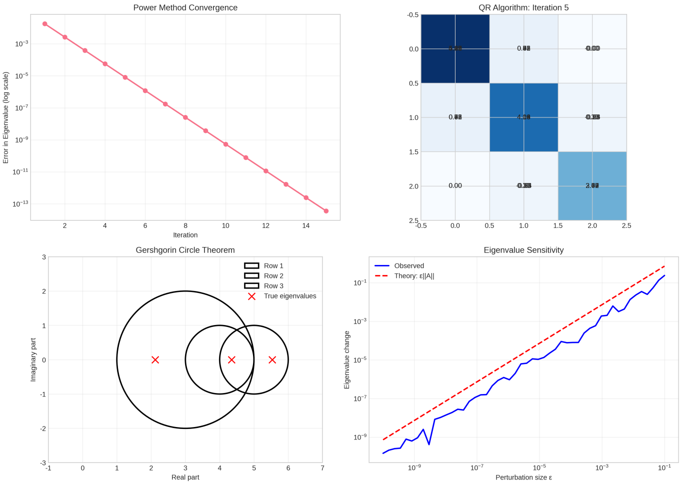

Figure: Comparison of eigenvalue algorithms. Left: convergence rates of power, inverse, and shifted methods. Right: geometric view of power method convergence to the dominant eigenvector.

QR Algorithm

Basic QR

Iterate $A^{(k)} = Q^{(k)}R^{(k)}$, $A^{(k+1)} = R^{(k)}Q^{(k)}$. With shifts and reduction to Hessenberg/tridiagonal forms, converges efficiently to Schur/diagonal form.

- **Shifts**: Wilkinson shift accelerates convergence
- **Complexity**: $O(n^3)$ setup, $O(n^2)$ per iteration on Hessenberg
- **Symmetric case**: Tridiagonal reduction and fast QR

Krylov Subspace Methods

Krylov Subspace

$\mathcal{K}_k(A, r_0) = \text{span}\{r_0, Ar_0, A^2 r_0, \ldots, A^{k-1} r_0\}$

- **Conjugate Gradient (CG)**: For SPD matrices

$$\|x_k - x^*\|_A \leq 2 \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1} \right)^k \|x_0 - x^*\|_A \tag{6}$$

- **GMRES**: For general matrices

$$\|r_k\| = \min_{p \in \mathcal{P}_k} \|p(A) r_0\| \tag{7}$$

Key advantage: Optimal approximation in Krylov subspace

Perturbation and Large-Scale Methods

- **Perturbation**: Weyl's theorem, Bauer–Fike bounds
- **Lanczos**: Symmetric matrices, tridiagonal projection, Ritz values
- **Arnoldi**: Nonsymmetric matrices, Hessenberg projection

Conjugate Gradient Algorithm

CG Algorithm for $Ax = b$ (A SPD)

1. $r_0 = b - Ax_0,\ p_0 = r_0$
2. For $k = 0, 1, 2, \ldots$ until convergence:

$$\alpha_k = \frac{r_k^T r_k}{p_k^T A p_k} \tag{8}$$

$$x_{k+1} = x_k + \alpha_k p_k \tag{9}$$

$$r_{k+1} = r_k - \alpha_k A p_k \tag{10}$$

$$\beta_k = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k} \tag{11}$$

$$p_{k+1} = r_{k+1} + \beta_k p_k \tag{12}$$

Memory: Only 4 vectors of length n

GMRES Algorithm

GMRES for General Matrices

- Build orthonormal basis $\{v_1, v_2, \ldots, v_k\}$ for $\mathcal{K}_k(A, r_0)$
- Solve least squares problem:

$$\min_{y \in \mathbb{R}^k} \|\beta e_1 - H_k y\|_2 \tag{13}$$

  where H_k is upper Hessenberg matrix
- Update: $x_k = x_0 + V_k y_k$

Advantages: Works for any matrix, minimizes residual norm
Disadvantages: Growing memory, restart needed

Preconditioning

Basic Idea

Solve $M^{-1}Ax = M^{-1}b$ instead of $Ax = b$

Goals:

- Reduce condition number: $\kappa(M^{-1}A) \ll \kappa(A)$
- M should be easy to invert

Common Preconditioners

- **Diagonal**: $M = \text{diag}(A)$
- **Incomplete LU**: $M \approx LU$ (sparse)
- **Multigrid**: Optimal for elliptic PDEs
- **Domain Decomposition**: Parallel-friendly

Practical Implementation Guidelines

Method Selection

- **SPD matrices**: Use CG with good preconditioner
- **General matrices**: Use GMRES with restart
- **Large sparse**: Classical methods with good ordering

Stopping Criteria

$$\frac{\|r_k\|}{\|r_0\|} < \text{tol} \quad \text{or} \quad \frac{\|r_k\|}{\|b\|} < \text{tol} \tag{14}$$

Typical tolerance: 10^{-6} to 10^{-12}

Computational Complexity

Cost per Iteration

- **Jacobi/Gauss-Seidel**: $O(n^2)$ for dense, $O(\text{nnz})$ for sparse
- **CG**: $O(n^2) + 1$ matrix-vector product
- **GMRES**: $O(kn) + 1$ matrix-vector product (k = iteration)

Total Complexity

- **CG**: $O(n^{3/2})$ for well-conditioned SPD
- **GMRES**: $O(n^2)$ to $O(n^3)$ depending on restart
- **Multigrid**: $O(n)$ optimal complexity

Key Takeaways

- **Power/Inverse/RQI**: foundational iterative eigenvalue solvers
- **QR with shifts**: robust all-eigenvalues method; structure reduction is key
- **Sensitivity**: eigenvalues/eigenvectors can be ill-conditioned
- **Lanczos/Arnoldi**: scalable approaches for large sparse problems

Next lecture: Singular Value Decomposition (SVD) and PCA