Special Topics in Applied Mathematics I
Solution 1

XIA JIAHAN

January 21, 2026

Question 1

Report: Iterative Methods for Eigenvalue Computation

Overview

In many practical applications involving large sparse matrices, computing the entire spectrum is computationally expensive and often unnecessary. Instead, iterative methods such as **Power Iteration** are employed to find specific eigenvalues, typically the one with the largest magnitude (the dominant eigenvalue).

However, when the goal is to find an eigenvalue closest to a specific scalar σ (e.g., $\sigma = \pi$), standard Power Iteration is insufficient. For this purpose, we utilize a variant known as **Inverse Iteration with Shift**. This method transforms the spectrum of the matrix so that the desired eigenvalue becomes the dominant one in the transformed problem.

Methodology: Inverse Iteration with Shift

Let A be a square matrix. We seek the eigenvalue λ of A that minimizes $|\lambda - \sigma|$. The method relies on the spectral mapping theorem: if λ is an eigenvalue of A, then

$$\mu = \frac{1}{\lambda - \sigma}$$

is an eigenvalue of the matrix $(A - \sigma I)^{-1}$.

Observe that if λ is close to σ, the denominator $\lambda - \sigma$ is small, making the magnitude $|\mu|$ very large. Thus, the eigenvalue of A closest to σ corresponds to the dominant eigenvalue of $(A - \sigma I)^{-1}$. We can therefore apply Power Iteration to the matrix $M = (A - \sigma I)^{-1}$ to find this dominant μ, and subsequently recover λ.

Algorithm Implementation

To implement this efficiently, we **do not** compute the inverse matrix $(A - \sigma I)^{-1}$ explicitly, as this is numerically unstable and costly ($O(n^3)$). Instead, we solve a linear system at each iteration.

Example Goal: Demonstrate convergence to the eigenvalue closest to π.
Procedure:

1. **Initialization**:
   - Select the shift $\sigma = \pi$.
   - Choose an initial guess vector $x^{(0)}$ (randomly generated) with $\|x^{(0)}\|_2 = 1$.
   - Set a tolerance ϵ (e.g., 10^{-6}).

2. **Iterative Loop** (for $k = 1, 2, \ldots$):

   (a) **Solve Linear System**: Instead of multiplying by inverse, solve for $y^{(k)}$ in:

   $$(A - \sigma I)y^{(k)} = x^{(k-1)}$$

   Implementation Note: Pre-compute the LU decomposition of $(A - \sigma I)$ once outside the loop to speed up this step to $O(n^2)$.

   (b) **Normalize**: To prevent overflow/underflow, normalize the vector:

   $$x^{(k)} = \frac{y^{(k)}}{\|y^{(k)}\|_2}$$

   (c) **Eigenvalue Estimation**: Compute the Rayleigh Quotient approximation for the eigenvalue of A:

   $$\lambda^{(k)} = (x^{(k)})^T A x^{(k)}$$

3. **Termination**: Stop when the residual $\|Ax^{(k)} - \lambda^{(k)}x^{(k)}\|_2 < \epsilon$ or when $|\lambda^{(k)} - \lambda^{(k-1)}|$ is sufficiently small.

Convergence

The method converges linearly. The rate of convergence is governed by the ratio $|\mu_2|/|\mu_1|$, where μ_1 and μ_2 are the largest and second-largest eigenvalues of $(A - \sigma I)^{-1}$. In terms of the original matrix A, this is:

$$\text{Rate} \approx \left| \frac{\lambda_{\text{closest}} - \sigma}{\lambda_{\text{second closest}} - \sigma} \right|$$

Because we chose $\sigma = \pi$ close to a target eigenvalue, this ratio is typically small, ensuring rapid convergence.