CENG 371 - Scientific Computing
Fall 2022
Homework 3

Sezgin, Mustafa
e2380863@ceng.metu.edu.tr

December 24, 2022

Question 1

a)

Power method is implemented in the file `power_method.m`. The function usage is as follows.

- `[eigvec, eigval] = power_method(A)`
- `[eigvec, eigval] = power_method(A, v)`

If the starting vector `v` is not provided, the function uses a starting vector with all elements 1. The function iterates until the change in the elements of calculated vector is less than a threshold, which is set to be `eps`.

b)

Shifted inverse power method is implemented in the file `inverse_power.m`. The function usage is as follows.

- `[eigvec, eigval] = inverse_power(A, alpha)`

The function iterates until the change in the elements of calculated vector is less than a threshold, which is set to be `eps`.

c)

For the matrix A given in the question, the largest eigenvalue in magnitude and the corresponding eigenvalue is calculated using the `power_method(A)` function. The result is produced at the end of 61 iterations.

$$\lambda_1 = 3.7321 \quad v_1 = \begin{bmatrix} 0.5 \\ -0.86603 \\ 1 \\ -0.86603 \\ 0.5 \end{bmatrix}$$

For the same matrix, the smallest eigenvalue in magnitude and the corresponding eigenvalue is calculated using the `inverse_power(A, alpha)` function, with `alpha = 0`. The result is produced at the end of 19 iterations.

$$\lambda_5 = 0.26795 \quad v_5 = \begin{bmatrix} 0.5 \\ 0.86603 \\ 1 \\ 0.86603 \\ 0.5 \end{bmatrix}$$

d)

1. Using classical method: We can use the classical method to find eigenvalues and eigenvectors of the matrix B.

$$|B - \lambda I| = 0$$

$$\begin{vmatrix} 0.2-\lambda & 0.3 & -0.5 \\ 0.6 & -0.8-\lambda & 0.2 \\ -1.0 & 0.1 & 0.9-\lambda \end{vmatrix} = 0$$

$$-\lambda^3 + 0.3\lambda^2 + 1.4\lambda = 0$$

Eigenvalues are the roots of the characteristic equation.

$$\lambda_1 = \frac{3+\sqrt{569}}{20} \approx 1.3427$$

$$\lambda_2 = 0$$

$$\lambda_3 = \frac{3-\sqrt{569}}{20} \approx -1.0427$$

Eigenvectors can be found by solving the homogeneous system $(B - \lambda I)v = 0$.

$$v_1 = \begin{bmatrix} -0.4458 \\ -0.0315 \\ 1 \end{bmatrix} \quad v_2 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad v_3 = \begin{bmatrix} 1.4915 \\ -4.5116 \\ 1 \end{bmatrix}$$

The dominant eigenvalue is λ_1 and the corresponding eigenvector is v_1.

2. Using power method: We can use the power method to find the dominant eigenvalue and the corresponding eigenvector of the matrix B.

$$x_0 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

$$x_1 = Bsx_0 = \begin{bmatrix} 0.2 & 0.3 & -0.5 \\ 0.6 & -0.8 & 0.2 \\ -1.0 & 0.1 & 0.9 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

$$x_2 = x_3 = \cdots = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

Power method fails. This is because x_0 was not chosen appropriately. That is, the condition *"x_0 must have a component in the direction of v_1"* does not hold.

$$x_0 = \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3 \implies \alpha_1, \alpha_3 = 0,\ \alpha_2 = 1$$

However, because of rounding errors, `power_method` function successfully returns the result at the end of 156 iterations.

Question 2

a)

Let λ_1 be the dominant eigenvalue and v_1 be the corresponding eigenvector of a matrix A, such that $Av_1 = \lambda_1 v_1$. Then,

$$B = A - \lambda_1 \frac{v_1 v_1^T}{v_1^T v_1}$$

and λ_1 is not an eigenvalue of B.

$$\begin{aligned} Bv_1 &= \left(A - \lambda_1 \frac{v_1 v_1^T}{v_1^T v_1} \right) v_1 \\ &= Av_1 - \lambda_1 \frac{v_1 v_1^T v_1}{v_1^T v_1} \\ &= \lambda_1 v_1 - \lambda_1 v_1 \\ &= 0 \end{aligned}$$

b)

The algorithm for the mentioned idea is implemented in the file `power_k.m`. The function usage is as follows.

- `[eigvecs, eigvals] = power_k(A, k)`
- `[eigvecs, eigvals] = power_k(A, k, v)`

If the starting vector `v` is not provided, the function uses a starting vector with all elements 1. The function runs `power_method` function as subroutine.

c)

Subspace iterations method is implemented in the file `subspace_iteration.m`. The function usage is as follows.

- `[eigvecs, eigvals] = subspace_iteration(A, k)`

The function iterates until the change in the elements of calculated vectors is less than a threshold, which is set to be `eps`. The function runs the built-in `qr` function as subroutine.

d)

The implementations produce correct solutions for most of the matrices tested. The function power_k calls power_method as subroutine in an iterative fashion. The function subspace_iteration uses the built-in qr function as subroutine. However, both implementations does not appropriately produce a solution for can_229 matrix.