MTH 373/573:
Scientific Computing

Homework 3

IIIT-Delhi, Monsoon 2022
Kaushik Kalyanaraman

Due by: November 25, 2022 (Friday) by 23:59 hours IST

Total: 60 points

Guidelines

This assignment should be answered individually.

- You are not allowed to discuss specifics of your solutions, solution strategies, or coding strategies with any other student in this class.
- Please consult with and carefully read through IIIT-Delhi's Academic Dishonesty Policy.

Please don't be a cheater!

- You shall submit your solutions via Google Classroom – there are no printed/handwritten/-paper submissions. Please type, take a picture, or scan your solutions and upload them as a single PDF file and only a single PDF file. You can find many websites that will merge your scanned images into a PDF file or, for the nerdy types, you can find a command line script to do this.
- You may post queries relating to any HW question to `#discussions` on the class's Slack channel.
- Start working on your solutions early and don't wait until close to due date.
- Each problem should have a clear and concise write-up.
- Please clearly show all steps involved in your solution.
- You will need to write a separate code for each computational problem and sometimes for some subproblems too. You should name each such file as `problem_`*n*`.py` where *n* is the problem number. For example, your file names could be `problem_5.py`, `problem_6a.py`, `problem_6b.py`, and so on.
- *Python tip:* You can import `Python` modules as follows:

```
import numpy as np
import scipy as sp
import matplotlib.pyplot as plt
import numpy.linalg as npla
import scipy.linalg as spla
```

- We shall only accept `*.py` `Python` files – no Jupyter Notebook submissions are allowed. Please write your code for the machine problems in a text editor and generate appropriate output via Terminal using `Python` (directly) or `IPython` (recommended).

Problem 1: Triangular Matrix Eigenvalues — 5 points

Show that the eigenvalues of a triangular matrix $A \in \mathbb{R}^{n \times n}$ are exactly its diagonal elements.

Hint: Use that the eigenvalues are roots of the characteristic polynomial.

Problem 2: Rank of a Matrix and Zero Eigenvalues — 5 points

Show that the rank of a nondefective matrix $A \in \mathbb{R}^{n \times n}$ is equal to the number of nonzero eigenvalues of this matrix.

Hint: Use that the elements of kernel of a matrix (when it is not full rank) are linear combinations of its eigenvectors corresponding to the zero eigevnalue. Use this in conjunction with rank nullity theorem.

Problem 3: Outerproduct Eigenvalues — 5 points

Suppose $u, v \in \mathbb{R}^n$ such that $u^T v = 1$. Let $A = uv^T$. Whare are the eigenvalues of A? (*Hint:* Use the result of the previous problem.) How many iterations does power iteration take to converge to the dominant eigenvalue-eigenvector pair for this matrix?

Problem 4: Positive Definite Eigenvalues — 5 points

Show that all eigenvalues of a real symmetric positive definite matrix are all real and strictly greater than 0.

Problem 5: Matrix Exponential — 10 points

Suppose $A \in \mathbb{R}^{n \times n}$ is symmetric and positive definite. (That is, all its eigenvalues are real and greater than zero; furthermore, A is nondefective). We define the *exponential* of the matrix as:

$$e^A := \sum_{k=0}^{\infty} \frac{1}{k!} A^k,$$

and this sum is unconditionally convergent (you do not have to prove this!). Provide an alternative expression for e^A in terms of the eigenvalues of A.

Hint: Note that A in this question is nondefective. Consider any eigenvalue $\lambda_i \in \mathbb{R}$ of this matrix ($i \in \{1, 2, \ldots, n\}$) and its corresponding eigenvector $u_i \in \mathbb{R}^n$. What can you say about $e^A u_i$? How can you use this to compute $e^A v$ for any arbitrary $v \in \mathbb{R}^n$ and conclude what e^A therefore is,

Problem 6: Power Iteration — 5 points

Implement normalized power iteration to compute the largest magnitude eigenvalue (and corresponding normalized eigenvector) and inverse iteration to compute the smallest magnitude

eigenvalue (and corresponding normalized eigenvector) of the matrix:

$$A = \begin{bmatrix} 2 & 3 & 2 \\ 10 & 3 & 4 \\ 3 & 6 & 1 \end{bmatrix}$$

Use $x_0 = [0 \quad 0 \quad 1]^T$ as your starting vector for both methods. You may use a library routine like `numpy.linalg.solve` for solving the linear system in inverse iteration.

Use a general real eigensystem library routine (like `numpy.linalg.eig()` in NumPy) to compute all of the eigenvalues and eigenvectors of the matrix, and compare the results with those obtained by your functions.

Problem 7: Shifted Inverse Iteration — 5 points

Implement shifted inverse iteration to compute the eigenvalue nearest to 2 and corresponding normalized eigenvector for the matrix:

$$A = \begin{bmatrix} 6 & 2 & 1 \\ 2 & 3 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Use an arbitrary starting vector.

Use a real symmetric eigensystem library routine (like `numpy.linalg.eigh()` in NumPy) to compute all of the eigenvalues and eigenvectors of the matrix, and compare the results with those obtained by your function.

Problem 8: Rayleigh Quotient Iteration — 10 points

Write a function to implement Rayleigh quotient iteration for computing an eigenvalue and corresponding eigenvector of a matrix. Test your program on the matrix in the Problem 6 using an arbitrary starting vector. How does the result compare with the eigenvalues and eigenvector from the library routine you obtained earlier. Finally, assume that the largest magnitude eigenvalue provided by the library routine is the true eigenvalue. Compute the rate of convergence of the Rayleigh quotient iteration to this eigenvalue.

Notes: For computing the rate, you need to compute how quickly the error goes to zero. In order to do this, at the k^{th} iteration step, compute the error e_k in the estimate of the largest magnitude eigenvalue as the difference between the Rayleigh quotient and the true largest magnitude eigenvalue. We now assume that this error e_k is a simple polynomial function of the error in the previous step e_{k-1}, that is, $e_k = Ce_{k-1}^r$ for some $r \in \mathbb{Z}$ and a constant $C \in R$. Estimate this exponent r as your convergence rate. Note that the constant C is irrelevant and can be factored away between successive iteration steps.

Problem 9: Modified QR Iteration — 10 points

Write a function to implement the following version of QR iteration with shifts for computing the eigenvalues of a general real matrix A.

Repeat until convergence:

(a) $\sigma = a_{n,n}$ (use corner entry as shift)

(b) Compute the full QR factorization $QR = A - \sigma I$ using a library routine such as `scipy.linalg.qr`

(c) $A = RQ + \sigma I$

Test your function on the matrices in Problems 6 and 7.