BLG 202E Homework - 2

Due 24.04.2020 23:59

Abdullah Ekrem Okur
okurabd@itu.edu.tr

Policy:

- Cheating is highly discouraged. It will be punished by a negative grade. Also disciplinary actions will be taken. Please do your homework on your own. Team work is not allowed. Pattern of your solutions must belong to only you.
- Upload your solutions through Ninova. Homeworks sent via e-mail and late submissions will not be accepted.
- You should write all your codes in Python language using Jupyter notebook. You can install Jupyter Notebook by following these steps on this documentation. If you are not familiar with Jupyter Notebook, you can check this tutorial.
- Prepare a report including all your solutions, codes and their results.
- You do not have to use Latex for the report but if you use Latex, you will get 10% more points. You can use this Latex template for the report.
- If you do not use Latex, the handwritten parts of the solutions must be presented on white paper legibly and scanned clearly. 10% penalty will be applied for illegible reports.

1. **[50 points]**

   There are many iterative methods to find solutions to the scalar, nonlinear equations in given interval. Thus, we need to choose a method for each application depending on the its criteria. Since each method has its own strengths, deciding which method to use can be difficult. In this question, you are expected to suggest a rough road map on which method should be chosen for a specific problem to overcome this issue. As a result of this study, you should compare all iterative methods given below for several possibilities and analyze their weaknesses and strengths.

   Methods we will cover:

   - Bisection method

- Fixed point iteration
- Newton's method
- Secant Method

In our textbook [1], desirable qualities of a root finding algorithm are defined as follows:

- Efficient: requires a small number of function evaluations.
- Robust: fails rarely, if ever. Announces failure if it does fail.
- Requires a minimal amount of additional data such as the function's derivative.
- Requires f to satisfy only minimal smoothness properties.
- Generalizes easily and naturally to many equations in many unknowns.

No algorithm we are aware of satisfies all of these criteria.

Consequently, you are expected to investigate the qualities of each algorithm by designing an experiment for each method with different equations, different initial points/intervals and different stopping criteria. You need to determine different experiment setup to present weaknesses and strengths of each method. For example, if you think a method is not robust, you should prove that it fails in some cases. All the experiment parameters are entirely your choices. You can use different graphics, tables, visualization techniques to strengthen your hypothesis. Steps given below are created to give you an idea of what to do and to make your job easier. If you are confident enough you may skip the below step-by-step suggestions and do your own thing but if you do that, do read these instructions anyway to ensure you have not missed something.

The minimum requirements to achieve the targeted solution are to complete the following tasks.

- You should implement a Python script for each method. Please write reusable codes to use them easily in the experiment phase.
- Select at least two nonlinear equations and find their solutions using Bisection method on an interval [a,b] with an absolute tolerance. Then, you should repeat the experiment with different intervals and tolerances. At the end of the experiment, you should present meaningful experimental setups in the report. These results should support your some hypothesis. For example, there should be some cases that the algorithm do not converge despite many iterations if you think Bisection method is not robust.
- You should present your experiment results with at least 3 different intervals and 2 different stopping criteria for each equation in your report.

- Produce different graphics showing your experiment results and put them on your report. (You can use Matplotlib library or any Python library.) Also, you can review our textbook or any numerical methods books/papers for inspiration on how to produce better graphics.
- Create tables showing your experiment results and put them on your report.
- Give comments about advantages and disadvantages of Bisection method.
- Apply all rules and steps to other 3 methods. You may need to update the suggestions according to each method's own requirements. For example, you should use different initial guesses instead of intervals for Newton's method.
- You can use different equations for the methods to examine their behaviour but please use at least one common equation for all methods to compare them better.
- Compare all methods according to the speed of convergence/rate of convergence by providing graphs.
- Prove theoretically what is the order of convergence (linear, quadratic, or superlinear) for each method and show that the theoretical results are also compatible with experimental data using different visualization techniques.
- Design and run an experiment that investigates the possibility of overflow.
- You can compare the methods theoretically if you think it is not possible to compare them experimentally according to a criterion.
- Summarize the findings and list pros/cons of each method.
- Come up with a rough road map on which method should be chosen for a specific problem

This question does not have a fixed solution. You are expected to explore the topic with experiments and to prepare a detailed report.
Note. You can use any sources to analyze and interpret the experiments but please give the references.
Highly recommended. After reading this question once, examine chapter 3 from our textbook by thinking how to solve this question.

Bonus [20 points].

Bonus criteria:

- Additional experiments
- Effective and impressive visualization techniques
- Good report design
- Creative solutions and comments

2. **[25 points]**

$$\begin{pmatrix} 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ 0 & -1 & 2 & -1 \\ -1 & 2 & -1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

(a) Decompose the matrix using partial pivoting as

$$PA = LU$$

where U is upper triangular, L is unit lower triangular and P is a permutation matrix. Record all the elementary matrices, $P^{(i)}$, and $M^{(i)}$ along the way.

(b) Find x that satisfies Ax = b.

3. **[25 points]**

The quality or impact of a scientific paper is generally evaluated using the total number of citations it have received in other papers. Since the approach does not provide sufficient and accurate information, various measurement metrics are developed such as h-index. In this question, you are expected to come up with an algorithm like h-index to gauge influence of papers. We will use "High-energy physics theory citation network" dataset[2] to evaluate your algorithms. This dataset consists of cross citations of 27770 papers from HEP-TH (High Energy Physics - Theory) section of arXiv. The file "cit-HepTh.txt" includes citations between the papers in the dataset. You can use the citation dataset as a graph network and convert it to adjacency matrix. Please carefully examine the dataset from the link above.

- Develop an algorithm based on the methods we have seen in the course so far to find the most influential 10 papers. (The algorithms included in the course should be sufficient.)
- Write your algorithm step by step.
- Implement your algorithm using Python. (You can use built-in Numpy methods.)
- Report results of your algorithm and name and id of the top 10 paper from the High-energy physics theory citation network dataset. (You can find the name of the papers from the arXiv using paper id and the section code 'HEP-TH')

Note. You can use genfromtxt method of Numpy to read citation file easily. Also, because the dataset is very large, use Numpy methods for all operations like sum,

division, multiplication, etc as much as possible. If you are not familiar with Python or Numpy, you can check this tutorial.

Bonus [10 points].

- Propose a basic real-life problem that can be solved by the algorithm you have developed question above.
- Collect data for this problem and report your sources.
- Turn your collected data into a 4x4 matrix to easily solve. (Use subset of your dataset if it is larger than 4x4. For example, for the question above, you would consider only random 4 papers and their citations between each other.)
- Guess results of your problem intuitively without coding or formulating.
- Solve problem by hand using the algorithm you have developed in the above. (without coding)
- Compare your mathematical solution with your guess.

References

[1] *Chapter 3: Nonlinear Equations in One Variable*, pp. 39–64. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/9780898719987.ch3

[2] J. Gehrke, P. Ginsparg, and J. Kleinberg, "Overview of the 2003 kdd cup," *SIGKDD Explor. Newsl.*, vol. 5, no. 2, pp. 149–151, Dec. 2003. [Online]. Available: http://doi.acm.org/10.1145/980972.980992