NE336 Quiz1 Oct 1st

1 Instructions

- The first line in each file should contain a comment with your complete name and student ID.
- All module import statements have been given but you may modify them however you wish.
- A minimal amount of comments (at least a few lines to explain your method of thought) is required and code with no comments will lose 0.5 marks in total.
- Any written answers can be provided in a word document or from a note taking application of your choice.

2 Conceptual questions

Please answer these in a word document and submit along with your code.

2.1 Question 1 (1.5 point)

In the `newton` function from `scipy`, the derivative of the function is an optional argument but to implement the NR method we need to follow

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$$

Please explain (in a few lines) what the algorithm does if we do not provide the derivative of the function and how it can still find the root.

2.2 Question 2 (1.5 points)

For solving systems of equations, we've mentioned both elimination and iterative methods. If we have 10000 linear equations, which approach would you recommend and why?

3 Programming questions

Submit the answers as python files.

3.1 Question 1 (3 points)

This question should be submitted as `finding_roots.py` and suitable methods can be found in `scipy.optimize`, linked here.

(a) Find the roots for

$$x^4 + 2xy^2 = 4$$
$$x^3y - 4y = 1$$

using a suitable built-in function (do not write or use your own functions).

(b) Include a suitable print statement to show the value for x and y of the root that you found.

(c) If this system has more than one root (hint: you may use Desmos), find one other root using the same method as part (a).

(d) Include another print statement to show the result for part (c).

Final question on next page

3.2 Question 2 (4 points)

There is a file called `systems_outline.py` attached to the dropbox. Please update this file with your answers and submit as `system.py` .

This file includes a function called `is_diag_dominant` to determine whether an array A from the system of equations $Ax = b$ is diagonally dominant.

Tasks to complete :

1. Please call the `is_diag_dominant` function with an example of A that would return None.

   Write your A and call the function under the `if __name__=="__main__"` section of the code and include a suitable print statement to show the returned value is None.

2. Consider the following system of equations and, write the corresponding A and b to determine whether it is diagonally dominant (by calling the function `is_diag_dominant(A)`)

$$\begin{aligned} 2x_1 - 6x_2 - x_3 &= -38 \\ -3x_1 - x_2 + 7x_3 &= -34 \\ -8x_1 + x_2 - 2x_3 &= -20 \end{aligned}$$

   Include suitable print statements to show the result of the function call.

3. If the function returns False, that is to say the system is not diagonally dominant, rearrange the rows such that it is diagonally dominant and write your new A and b arrays.

   Show that the function now returns True with a suitable print statement.

4. Would Gauss Seidel be guaranteed to converge if we did not make the system diagonally dominant? Please explain your answer as a comment in your code.

5. Please show *only two* iterations of the Gauss Seidel method for this diagonally dominant version of the system and print the values of x after each iteration. It can be written as a script similar to class. You do not need to apply relaxation.

6. **Optional Bonus** Continue the iterations for Gauss Seidel until convergence is achieved. Use a tol of $1e-5$.

7. Please find x using another method of your choice (builtin method is fine) and print the result. Comment on whether your two GS iterations were going in the right direction (towards the answer) and how you could potentially speed this up (get to the answer with fewer iterations).