Computational Astrophysics HW3

Yi-Hsiang Kuo, 110022506

(Dated: Oct. 27, 2022)

I. EXERCISE 2

(1) The equation of **vector norm (L_p-norms)**:

$$||x||_p = (\sum_{i=1}^{n} |x_i|^p)^{\frac{1}{p}} \tag{1}$$

After calculation, we can get first and second norms 3.3 and 2.7 respectively for **infinite norm**, applying:

$$||x||_\infty = max|x_i|, \quad 1 \leq i \leq n \tag{2}$$

and get the answer: 2.6.

(2) For the matrix A, applying the **matrix norm** formula:

$$||A|| = max_{x \neq 0} \frac{||Ax||}{||x||} \tag{3}$$

$$||A||_1 = max_j \sum_{i=1}^{n} |a_{ij}| \tag{4}$$

$$||A||_\infty = max_i \sum_{j=1}^{n} |a_{ij}| \tag{5}$$

we can get the L1-norm (maximum absolute column sum):78, and the L_∞-norm (maximum absolute row sum):83

II. EXERCISE 3

To evaluate the real root(s) of:

$$x^3 + 1.5x^2 - 5.75x + 4.37 = 0 \tag{6}$$

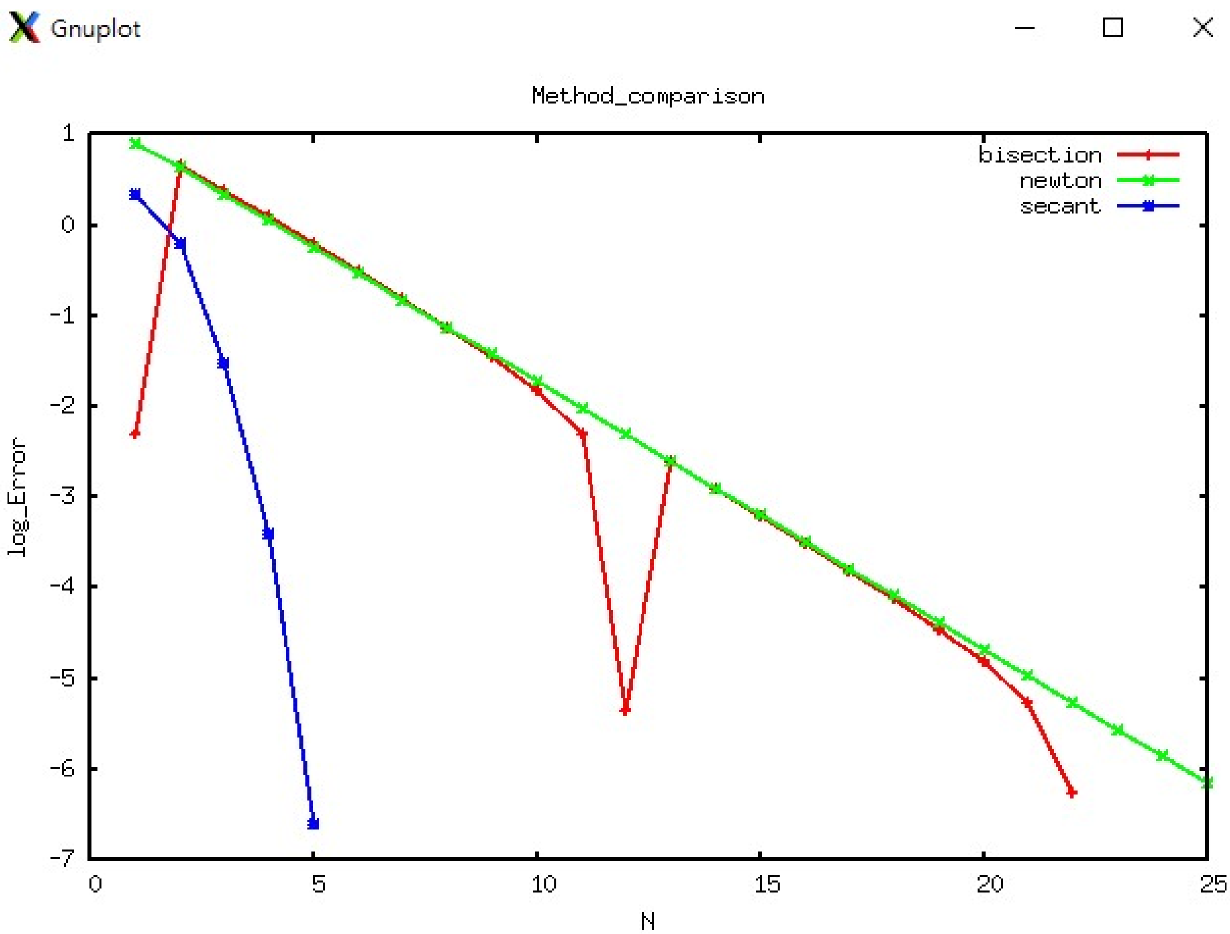

FIG. 1. Exercise3 Method comparison

using (a) bisection method with initial guess from -4.0 to -3.0; (b) Newton-Raphson's method with initial point $x = -3.0$ and (c) secant method with initial guess from -4.0 to -3.0, and FIG. 1. is the comparison result.

By the way, for Newton-Raphson's method, if our initial guess is $x = -3.5$, the iteration value N still need to approach $N = 14$, it seems that (c) secant method is the best way here.

III. EXERCISE 4

(1)(2)(3) The analytical calculations wrote on FIG. 2. :

(1) $$Ax = \begin{bmatrix} 1 & 2 & 2 \\ 4 & 6 & 8 \\ 4 & 8 & 10 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \\ 10 \end{bmatrix} = b$$

step1 $\hookrightarrow$ $M_1 A x = M_1 b$, where $M_1 = \begin{bmatrix} 1 & 0 & 0 \\ -4 & 1 & 0 \\ -4 & 0 & 1 \end{bmatrix}$

$$M_1 A = \begin{bmatrix} 1 & 0 & 0 \\ -4 & 1 & 0 \\ -4 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 2 & 2 \\ 4 & 6 & 8 \\ 4 & 8 & 10 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 2 \\ 0 & -2 & 0 \\ 0 & 0 & 2 \end{bmatrix}, \quad M_1 b = \begin{bmatrix} 1 & 0 & 0 \\ -4 & 1 & 0 \\ -4 & 0 & 1 \end{bmatrix}\begin{bmatrix} 4 \\ 6 \\ 10 \end{bmatrix} = \begin{bmatrix} 4 \\ -10 \\ -6 \end{bmatrix}$$

reduced to : $$Ux = \begin{bmatrix} 1 & 2 & 2 \\ 0 & -2 & 0 \\ 0 & 0 & 2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4 \\ -10 \\ -6 \end{bmatrix} \Rightarrow \begin{cases} x_1 = 0 \\ x_2 = 5 \\ x_3 = -3 \end{cases}$$ #.

(2) $$L = M_1^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 4 & 0 & 1 \end{bmatrix}, \quad U = \begin{bmatrix} 1 & 2 & 2 \\ 0 & -2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

(3) $$LU = \begin{bmatrix} 1 & 0 & 0 \\ 4 & 1 & 0 \\ 4 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 2 & 2 \\ 0 & -2 & 0 \\ 0 & 0 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 2 & 2 \\ 4 & 6 & 8 \\ 4 & 8 & 10 \end{bmatrix} = A$$ #

FIG. 2. Exercise4 Gaussian elimination

IV. EXERCISE 5

(1) To find the optimal h which minimize the total error (64-bit):

$$\epsilon_{tot} = \epsilon_{tr} + \epsilon_{ro} \approx \frac{f^{(2)}}{2}h + \frac{\epsilon_m}{h} \tag{7}$$

Since $f^{(2)} \approx 1$, so after differentiation we find that **when** $h \approx 4.47 \times 10^{-8}$

(2) To find the optimal h which minimize the total error of (64-bit) of **the central-difference algorithm**:

$$\epsilon_{tot} = \epsilon_{tr} + \epsilon_{ro} \approx \frac{f^{(3)}}{24}h^2 + \frac{\epsilon_m}{h} \tag{8}$$

Since $f^{(3)} \approx 1$, so after differentiation we find that **when** $h \approx 2.29 \times 10^{-5}$

(3)Put the optimal h back, we can get the minimal total error for both method, 4.47×10^{-8} and 6.55×10^{-11} for method 1 and method 2 respectively, so method 2 is better behaved, also the step size h is smaller for method 2.

V. EXERCISE 6

(1) I choose python to do the work and the step size $h = 0.1$ for this question, and the **git gub link is here** with the result FIG. 10 and FIG. 11, however I have not come up with the answer when $x = 100$

(2)