Homework 3

Computational Astrophysics (ASTR660)

(Due at the start of class on October 27, 2022)

Exercise 1

1. Read the tutorial for Makefiles: `https://makefiletutorial.com`.

2. Read the root-finding documentation in `scipy`: `https://docs.scipy.org/doc/scipy/reference/optimize.html`.

3. Read the linear algebra documentation in `scipy`: `https://docs.scipy.org/doc/scipy/reference/linalg.html`.

Exercise 2

[Vector and matrix norms (0.5 pt)

(1) For the vector $\boldsymbol{x} = [2.6, -0.7]^T$, calculate the first, second, and infinity norms.

(2) For the matrix

$$\boldsymbol{A} = \begin{bmatrix} 7 & -3 & 2 \\ 1 & 1 & 5 \\ 2 & -2 & 1 \end{bmatrix}, \tag{2.1}$$

calculate the first and infinity norms.

Exercise 3

[Solving nonlinear systems (1 pt)]

Please modify the Fortran program we developed in the Lecture 5 to evaluate the real root(s) of

$$x^3 + 1.5x^2 - 5.75x + 4.37 = 0. \tag{3.1}$$

Please obtain the solution using (a) bisection method, (b) Newton-Raphson's method, and (c) secant method. Compare the performance of the methods by drawing a plot.

Exercise 4

[Analytical – Gaussian elimination and LU decomposition (1.5 pt)]

For this problem, please do the *analytical* calculations step by step and do not use a computer.

(1) Please solve the linear system using Gaussian elimination:

$$\boldsymbol{A}\boldsymbol{x} = \begin{bmatrix} 1 & 2 & 2 \\ 4 & 6 & 8 \\ 4 & 8 & 10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 6 \\ 10 \end{bmatrix} = \boldsymbol{b}. \tag{4.1}$$

(2) What is the LU decomposition of $\boldsymbol{A}$?

(3) Show that $\boldsymbol{A} = \boldsymbol{L}\boldsymbol{U}$.

Exercise 5

[Programming – Gaussian elimination and LU decomposition (2 pt)]

(1) Implement LU decomposition in the subroutine `LU_decomposition` in `linalg.f90` following the algorithm taught in Lecture 6. Call this subroutine from `matrix.f90` and verify your code using the example shown in the lecture.

(2) Apply the above subroutine to Exercise 4 and show that your code could successfully decompose $\boldsymbol{A}$ (as defined in Equation 4.1) into $\boldsymbol{L}$ and $\boldsymbol{U}$.

(3) Implement the subroutine `solve_lu` in `linalg.f90` to use LU decomposition to solve Equation 4.1. Recall that the solution vector $\boldsymbol{x}$ can be found by performing the following two steps:

$$\begin{aligned} \boldsymbol{L}\boldsymbol{y} &= \boldsymbol{b} \\ \boldsymbol{U}\boldsymbol{x} &= \boldsymbol{y}, \end{aligned} \tag{5.1}$$

where the first and second equations can be solved using the forward-substitution and back-substitution methods for triangular matrices, respectively.

Exercise 6

[Solving linear systems using scipy (2 pt)]

(1) Consider a linear system $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ where $\boldsymbol{A}$ is a banded $n \times n$ matrix

$$\begin{bmatrix} 9 & -4 & 1 & 0 & \dots & \dots & 0 \\ -4 & 6 & -4 & 1 & \dots & \dots & 0 \\ 1 & -4 & 6 & -4 & 1 & \dots & 0 \\ 0 & 1 & -4 & 6 & -4 & 1 & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & 1 & -4 & 6 & -4 & 1 \\ \dots & \dots & \dots & 1 & -4 & 5 & -2 \\ 0 & \dots & \dots & 0 & 1 & -2 & 1 \end{bmatrix} \tag{6.1}$$

and $b_i = 1$ for $i = 1, \dots, n$. Let $n = 100$, solve the linear system using LU decomposition in `scipy`.

(2) Solve the same linear system using the solver designed for banded matrix (`scipy.linalg.solve_banded`) and compare the performance with (1). You could use the Python interface (e.g., `timeit` or `datatime`) to measure the computing time.

(3) What is the condition number of $\boldsymbol{A}$?