111B Data Science and Python Programming

Homework Assignment #5

Due: 5/5 12:00:00

Welcome to the world of numerical linear algebra!!!

Problem #1: Partial Pivoting Method

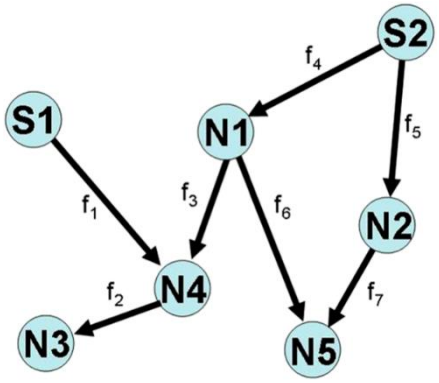
Please use partial pivoting method to rearrange the pivot position. Afterward, using either Jacobi or G-S iterative to solve the results.

$$\begin{bmatrix} 4 & -1 & 1 \\ 4 & -8 & 1 \\ -2 & 1 & 5 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 7 \\ -21 \\ 15 \end{bmatrix}$$

Since matrix A breaks the SDD condition, please solving $\mathbf{Ax} = \mathbf{b}$ by (1) Iterative Jacobi Method (2) Iterative G-S Method before pivoting.

Problem #2: Incidence Matrix and Graph Problem

Consider the following network consisting of two power supply stations denoted by S_1 and S_2 and five power recipient nodes denoted by N_1 to N_5. The nodes are connected by power lines, which are denoted by arrows, and power can flow between nodes along these lines in both directions.



Let d_i be a positive scalar denoting the power demands for node i, and assume that this demand must be met exactly. The capacity of the power supply stations is denoted by S. Power supply stations must run at their capacity. For each arrow, let f_j be the power flow along that arrow. Negative flow implies that power is running in the opposite direction of the arrow.

Please write a function "**my_flow_calculator(S, d)**", where S is a 1×2 vector representing the capacity of each power supply station, and d is a 1×5 row vector representing the demands at each node (i.e., $d[0]$ is the demand at node 1). The output argument, f, should be a 1×7 row vector denoting the flows in the network (i.e., $f[0] = f_1$ in the diagram). The flows contained in f should satisfy all constraints of the system, like power generation and demands. Note that there may be more than one solution to the system of equations. The total flow into a node must equal the total flow out of the node plus the demand; that is, for each node i, $f_{inflow} = f_{outflow} + d_i$. You may assume that $\sum S_j = \sum d_i$.

Here are some test cases:

```
s = np.array([[10, 10]])
d = np.array([[4, 4, 4, 4, 4]])
# f = [[10.0, 4.0, -2.0, 4.5, 5.5, 2.5, 1.5]]
f = my_flow_calculator(s, d)

s = np.array([[10, 10]])
d = np.array([[3, 4, 5, 4, 4]])
# f = [[10.0, 5.0, -1.0, 4.5, 5.5, 2.5, 1.5]]
f = my_flow_calculator(s, d)
```

Problem #3: LU Decomposition, $\mathbf{PA} = \mathbf{LU}$

Please use pivoting method to rearrange the pivot position, then solving the random generated linear equation 10-by-10 matrix. Compare the results with

$$\mathbf{P}, \mathbf{L}, \mathbf{U} = \text{scipy.linalg.lu}(\mathbf{A})$$

A. Random generated a 10-by-10 matrix ($\mathbf{A}$) and 10-by-1 vector ($\mathbf{y}$), check its SDD condition.
B. Use Gauss Elimination to solve the linear equation in $\mathbf{A}$.
C. Get the lower triangular matrix $\mathbf{L}$ and upper triangular matrix $\mathbf{U}$ from the Gauss Elimination.

Please accomplish this homework with an organized code (e.g., with main script and function script). For example, you can package your scripts that related to the class object in a module "**obj.py**", some useful functions in other module, and remain the main content in the main script "**main_hw5.py**" clear. In addition, you should use "**argparse**" to set all related parameters of this homework. Here is a template for your code structure:

```
111B_hw5_0123456789
    ├─ obj.py          # Objects
    ├─ ???.py          # ??? for hw5
    └─ main_hw5.py     # Main script of hw5
```

You don't need to follow this structure, just keep your main script clean.

Hand in procedure:

As we had mentioned in the lecture, you should list all your collaborators in your programs. Here is the template:

```
"""
Created on Sun Aug 7 01:23:45 2022

@author: Xi Winnie, student ID

@collaborators: Jane Doe, her student ID
                John Doe, his student ID
"""
```

Please save your code as a "**.zip**", "**.7z**", or "**.rar**" file, where the file name should follow this format:

111B_hw5_ID.zip

For example,

111B_hw5_0123456789.zip

Please be aware. **We are not going to accept any homework file with wrong file name or without signature**. Please double check the content of your files.

Once you have accomplished your works, you can upload your homework to the "E3@NYCU" system. There will be a section for uploading your homework.