

## Tugas 4 Praktikum Metode Numerik Semester Genap Tahun Ajaran 2021/2022

### Petunjuk Umum:

1. Kerjakan secara individu
2. Kerjakan tugas ini dengan bahasa pemrograman python dengan format file python notebook (**file berbentuk .ipynb BUKAN .py**). Anda disarankan menggunakan jupyter untuk mengerjakan tugas ini.
3. **Sertakan penjelasan untuk setiap variable yang digunakan dan setiap proses secara singkat** di samping potongan kode (dengan '#'). **Sertakan juga penjelasan program secara lengkap** (idenya apa, bagaimana cara eksekusi dalam program atau algoritma program yang digunakan pada **cell dibawah program**. **Sertakan beberapa contoh input dan output**.

Contoh:

```
In [1]: a=input("Ini buat input: ") #untuk menyimpan yang akan diprint
        b=str(a) #paksa nilai dari variabel a menjadi str
        print(b)

Ini buat input: output
output

Program ini adalah program untuk print input dari user.
Idenya adalah menyimpan nilai input dari user kedalam suatu variabel lalu variabel yang disimpan akan diprint.
Algoritmanya:
1. Simpan input user dalam sebuah variabel a
2. Paksa variabel input menjadi sebuah string lalu simpen ke variabel baru b
3. Print variabel b
```

4. Format nama file adalah:  
**Nama\_NPM\_Kelas SIAK\_Tugas4PrakMetnum**  
Contoh penamaan yang benar:  
Baek Yi Jin\_2006512343\_Kelas A\_Tugas4PrakMetnum
5. Batas Pengumpulan tugas ini adalah **Minggu, 08 Mei 2022, pukul 23:00 WIB**.  
Tugas dikumpulkan sesuai dengan kelas SIAK anda:  
Kelas A: Kelas Metode Numerik A EMAS2  
Kelas B: Kelas Metode Numerik B EMAS2  
Kelas C: Kelas Metode Numerik C EMAS2 dan google form, link akan diberikan di dalam grup LINE kelas oleh aslab.  
Kelas D: Google form, link akan diberikan di dalam grup LINE kelas oleh aslab.
6. **Dilarang melakukan plagiarism** atau menduplikasi dalam mengerjakan tugas ini. Apabila terdapat kesamaan program atau penjelasan pada tugas yang dikumpulkan, **NILAI TUGAS PRAKTIKUM METODE NUMERIK ANDA LANGSUNG MENJADI 0 TANPA PERINGATAN** bagi semua pihak yang terlibat plagiarism dalam tugas ini.
7. Module yang boleh digunakan pada tugas ini hanya numpy. Penggunaan module selain numpy harap dikonfirmasi ke narahubung terlebih dahulu.
8. Apabila ada yang ingin ditanyakan, silakan mengontak salah satu kontak berikut:  
Richard Mulyadi (line: richardmulyadi29)  
Angelica Patricia D. S. (line: angelica.patricia)  
Rafi Alvanzah (line: rafi79)

## Soal Nomor 1

Baek Yi Jin mengaplikasikan Eliminasi Gauss dan Substitusi Balik pada suatu SPL yang memiliki elemen pivot yang cukup kecil dibandingkan dengan elemen di bawahnya. Namun, tiba-tiba muncul masalah ketika round-off error dilibatkan dalam perhitungan. Hal ini menyebabkan hasil dari substitusi balik jauh dari hasil solusi eksak.

Oleh karena itu, Baek Yi Jin meminta Beek Hee Do untuk melakukan strategi pivoting (Partial Pivoting dan Scaled Partial Pivoting). Strategi pivoting adalah memilih elemen terbesar yang entrinya berada di bawah elemen pivot untuk menjadi pivot baru, kemudian dilakukan penukaran baris elemen pivot lama dengan yang baru. Hal ini akan mampu mengatasi masalah sebelumnya. Ketika sudah dilakukan pivoting, baru dapat digunakan substitusi balik untuk memperoleh hasil solusi dari SPL yang ada.

### 1. Partial Pivoting

Pilih  $p \geq k$  terkecil sedemikian sehingga

$$|a_{pk}^{(k)}| = |a_{ik}^{(k)}|$$

Kemudian, lakukan operasi  $(E_k) \leftrightarrow (E_p)$

### 2. Scaled Partial Pivoting

Definisikan  $s_i = |a_{ij}|$  (scalar untuk tiap kolom)

Pilih  $p \geq k$  terkecil sedemikian sehingga

$$\frac{|a_{pk}^{(k)}|}{s_k} = \frac{|a_{ik}^{(k)}|}{s_i}$$

Kemudian, lakukan operasi  $(E_k) \leftrightarrow (E_p)$

Buatlah program yang dapat menyelesaikan SPL dengan input berupa matriks berukuran  $4 \times 5$  dengan ketentuan sebagai berikut.

- Elemen baris pertama dan keempat adalah hasil random dengan

$$|a_{ij}| < 100, a_{ij} \in \mathbb{Z}$$

- Baris kedua berisi 5 digit pertama NPM
- Baris ketiga berisi 5 digit terakhir NPM

Output yang dihasilkan berupa solusi SPL yang ada dengan menggunakan kedua strategi pivoting.

Jangan lupa untuk menampilkan langkah-langkahnya, ya!

### **Bonus :**

1. Gunakan error handling jika SPL tidak memiliki solusi unik
2. Buatlah program yang dapat diulang

## Contoh Running Program

Selamat Datang di Program Penyelesaian SPL

### Iterasi 1

Masukkan NPM anda: 2736152947

Berikut adalah augmented matrix yang terbentuk.

```
[[-49. -17. 28. 78. 41.]
 [ 2. 7. 3. 6. 1.]
 [ 5. 2. 9. 4. 7.]
 [ 47. -74. -68. -26. -68.]]
```

=== PARTIAL PIVOTING ===

```
[[-4.90000000e+01 -1.70000000e+01 2.80000000e+01 7.80000000e+01
 4.10000000e+01]
 [ 2.22044605e-16 6.30612245e+00 4.14285714e+00 9.18367347e+00
 2.67346939e+00]
 [ 0.00000000e+00 2.65306122e-01 1.18571429e+01 1.19591837e+01
 1.11836735e+01]
 [ 0.00000000e+00 -9.03061224e+01 -4.11428571e+01 4.88163265e+01
 -2.86734694e+01]]
```

```
[[-4.90000000e+01 -1.70000000e+01 2.80000000e+01 7.80000000e+01
 4.10000000e+01]
 [ 0.00000000e+00 -9.03061224e+01 -4.11428571e+01 4.88163265e+01
 -2.86734694e+01]
 [ 0.00000000e+00 0.00000000e+00 1.17362712e+01 1.21025989e+01
 1.10994350e+01]
 [ 2.22044605e-16 0.00000000e+00 1.26983051e+00 1.25925424e+01
 6.71186441e-01]]
```

```
[[-4.90000000e+01 -1.70000000e+01 2.80000000e+01 7.80000000e+01
 4.10000000e+01]
 [ 0.00000000e+00 -9.03061224e+01 -4.11428571e+01 4.88163265e+01
 -2.86734694e+01]
 [ 0.00000000e+00 0.00000000e+00 1.17362712e+01 1.21025989e+01
 1.10994350e+01]
 [ 2.22044605e-16 0.00000000e+00 0.00000000e+00 1.12830763e+01
 -5.29740242e-01]]
```

Solusi SPL :

[-0.28759787 -0.16079486 0.99415322 -0.04694998]

=== SCALED PARTIAL PIVOTING ===

```
[ [ 47. -74. -68. -26. -68. ]
 [ 0. 10.14893617 5.89361702 7.10638298 3.89361702]
 [ 5. 2. 9. 4. 7. ]
 [-49. -17. 28. 78. 41. ]]
```

```
[ [ 47. -74. -68. -26. -68. ]
 [ 0. 10.14893617 5.89361702 7.10638298 3.89361702]
 [ 0. 9.87234043 16.23404255 6.76595745 14.23404255]
 [-49. -17. 28. 78. 41. ]]
```

```
[ [ 47. -74. -68. -26. -68. ]
 [ 0. 10.14893617 5.89361702 7.10638298 3.89361702]
 [ 0. 9.87234043 16.23404255 6.76595745 14.23404255]
 [ 0. -94.14893617 -42.89361702 50.89361702 -29.89361702]]
```

```
[ [ 47. -74. -68. -26. -68. ]
 [ 0. 10.14893617 5.89361702 7.10638298 3.89361702]
 [ 0. 0. 10.50104822 -0.14675052 10.44654088]
 [ 0. -94.14893617 -42.89361702 50.89361702 -29.89361702]]
```

```
[ [ 4.70000000e+01 -7.40000000e+01 -6.80000000e+01 -2.60000000e+01
 -6.80000000e+01]
 [ 0.00000000e+00 1.01489362e+01 5.89361702e+00 7.10638298e+00
 3.89361702e+00]
 [ 0.00000000e+00 0.00000000e+00 1.05010482e+01 -1.46750524e-01
 1.04465409e+01]
 [ 0.00000000e+00 1.42108547e-14 1.17798742e+01 1.16817610e+02
 6.22641509e+00]]
```

```
[ [ 4.70000000e+01 -7.40000000e+01 -6.80000000e+01 -2.60000000e+01
 -6.80000000e+01]
 [ 0.00000000e+00 1.01489362e+01 5.89361702e+00 7.10638298e+00
 3.89361702e+00]
 [ 0.00000000e+00 0.00000000e+00 1.05010482e+01 -1.46750524e-01
 1.04465409e+01]
 [ 0.00000000e+00 1.42108547e-14 0.00000000e+00 1.16982232e+02
 -5.49231384e+00]]
```

Solusi SPL :

[-0.28759787 -0.16079486 0.99415322 -0.04694998]

Apakah ingin menjalankan program lagi?

## Soal Nomor 2

Kamisato Ayato sedang berjalan-jalan santai dan secara tidak sengaja menemukan sebuah peti harta karun. Peti ini ternyata dikunci password dengan 9 kotak yang harus ia isi dengan angka. Tidak jauh dari peti itu terdapat sebuah gulungan tua yang berisi cara bagaimana Ayato dapat menemukan semua angka-angka yang diperlukan untuk membuka petinya. Ayato berniat melakukan *brute force* untuk mencari semua passwordnya, akan tetapi peti itu terlihat sangat rapuh sehingga hanya dapat dilakukan 1x percobaan.

Bantu Ayato menemukan semua angka-angka yang diperlukan untuk membuka peti dalam 1x percobaan dengan mengikuti petunjuk pada gulungan tua. Gulungan itu bertuliskan:

1. Definisikan matriks awal  $A$  sebagai matriks  $3 \times 3$ :

$$A = \begin{bmatrix} 7 & 3 & 6 \\ 1 & 5 & 2 \\ 6 & 2 & 7 \end{bmatrix}$$

2. Gunakan LU Factorization untuk mencari matriks segitiga bawah ( $L$ ) dan segitiga atas ( $U$ ) dari matriks  $A$
3. Definisikan matriks  $A_1$  sebagai  $A_1 = L + U$
4. Ulangi langkah 2 dan 3 sebanyak 10 kali sampai diperoleh matriks  $A_{10}$
5. Catatan: Matriks  $A_n = L_{n-1} + U_{n-1}$ , contoh:

Matriks  $A_1 = L + U$ , di mana  $L$  dan  $U$  adalah hasil Faktorisasi LU matriks  $A$

Matriks  $A_2 = L + U$ , di mana  $L$  dan  $U$  adalah hasil Faktorisasi LU matriks  $A_1$

Matriks  $A_3 = L + U$ , di mana  $L$  dan  $U$  adalah hasil Faktorisasi LU matriks  $A_2$

...

Matriks  $A_{10} = L + U$ , di mana  $L$  dan  $U$  adalah hasil Faktorisasi LU matriks  $A_9$

Hint: gunakan `round()` pada setiap elemen dari matriks  $L$  dan matriks  $U$  guna menghindari angka yang terlalu rumit untuk dilakukan faktorisasi LU berikutnya

## Contoh Running Program

↳ Masukkan matrix A Anda:  
[[7,3,6],[1,5,2],[6,2,7]]

=====

Faktorisasi LU matrix A 0

Matrix L  
[[1. 0. 0.]  
[0. 1. 0.]  
[0. 0. 1.]]

Matrix U

[[7 3 6]  
[0 4 1]  
[0 0 1]]

Matrix A 1

[[8. 3. 6.]  
[0. 5. 1.]  
[0. 0. 2.]]

=====

Faktorisasi LU matrix A 1

Matrix L  
[[1. 0. 0.]  
[0. 1. 0.]  
[0. 0. 1.]]

Matrix U

[[8. 3. 6.]  
[0. 5. 1.]  
[0. 0. 2.]]

Matrix A 2

[[9. 3. 6.]  
[0. 6. 1.]  
[0. 0. 3.]]

=====

Faktorisasi LU matrix A 2

Matrix L  
[[1. 0. 0.]  
[0. 1. 0.]  
[0. 0. 1.]]

Matrix U

[[9. 3. 6.]  
[0. 6. 1.]  
[0. 0. 3.]]

Matrix A 3

[[10. 3. 6.]  
[ 0. 7. 1.]  
[ 0. 0. 4.]]

=====

dst...

=====

Faktorisasi LU matrix A 8

Matrix L  
[[1. 0. 0.]  
[0. 1. 0.]  
[0. 0. 1.]]

Matrix U

[[15. 3. 6.]  
[ 0. 12. 1.]  
[ 0. 0. 9.]]

Matrix A 9

[[16. 3. 6.]  
[ 0. 13. 1.]  
[ 0. 0. 10.]]

=====

Faktorisasi LU matrix A 9

Matrix L  
[[1. 0. 0.]  
[0. 1. 0.]  
[0. 0. 1.]]

Matrix U

[[16. 3. 6.]  
[ 0. 13. 1.]  
[ 0. 0. 10.]]

Matrix A 10

[[17. 3. 6.]  
[ 0. 14. 1.]  
[ 0. 0. 11.]]

\*\*\*\*\*

Diperoleh matrix A10:

[[17. 3. 6.]  
[ 0. 14. 1.]  
[ 0. 0. 11.]]