**Tugas 5 Praktikum Metode Numerik**

**Semester Genap Tahun Ajaran 2022/2023**

**Petunjuk umum:**

1. Kerjakan secara individu.
2. Kerjakan tugas ini menggunakan bahasa pemrograman Python dengan *file format* berupa *interactive Python notebook* (yaitu *file* berbentuk **.ipynb** BUKAN .py), yang bisa dibuat misalnya menggunakan Jupyter Notebook atau Google Colaboratory.
3. **Harap sertakan penjelasan untuk setiap variabel yang digunakan dan setiap proses secara singkat** di samping potongan kode (dengan *comment*, ‘#'). Selain itu, **sertakan juga penjelasan program** (yang bisa mencakupi idenya apa, bagaimana cara eksekusi program, atau tentang algoritma program yang digunakan) **pada *cell* di sebelah (atas/bawah) program**.
4. Format nama *file* untuk Tugas 5 adalah:

**Nama Lengkap\_NPM\_Kelas SIAK\_Tugas5PrakMetnum.ipynb**

Contoh penamaan yang benar:

Johan Frederik Steffensen\_2234567890\_Kelas F\_Tugas5PrakMetnum.ipynb

1. Pengumpulan tugas dilakukan ke Google Forms berikut ini, sesuai dengan kelas Anda di SIAK NG (*link* akan selalu sama untuk semua tugas praktikum metode numerik):

* Kelas A: <https://forms.gle/Wgin1RdVyrNZ7xDbA>
* Kelas B: <https://forms.gle/cqurG75aqdTLq1LR8>
* Kelas C: <https://forms.gle/yPKk42n1mTLrJzFE6>
* Kelas D: <https://forms.gle/7TnSmCT9DcmhBvvo6>
* Kelas E: <https://forms.gle/SSQUksLtwsLkYTxT6>

1. Apabila ada yang ingin direvisi setelah pengumpulan, lakukan pengumpulan ulang di Google Forms yang sama, tambahkan keterangan bahwa ada revisi, dan tambahkan kata “revisi” pada bagian akhir nama *file*. Revisi boleh dilakukan berkali-kali.
2. Durasi pengerjaan Tugas 5 ini adalah **1 (satu) minggu**, dan tenggat waktu (*deadline*) pengumpulan Tugas 5 ini adalah **Minggu, 4 Juni 2023, pukul 13.00 WIB.**

Mohon manfaatkan waktu Anda dengan baik (seperti mencicil pengerjaan, bahkan sudah selesai dari jauh-jauh hari) agar mengumpulkan tugas sebelum *deadline*. Keterlambatan pengumpulan akan dikenakan pengurangan nilai (sudah dibicarakan dengan dosen). Lebih baik tidak terlambat daripada terlambat, dan lebih baik terlambat daripada tidak mengumpulkan sama sekali.

1. Sesuai standar Universitas Indonesia, **plagiarisme dilarang keras** dan bisa menyebabkan nilai tugas praktikum **menjadi nol untuk semua pihak yang terlibat, tanpa peringatan apapun.**
2. *Module* atau *package* Python yang boleh digunakan (di-*import*) untuk Tugas 5 ini hanyalah numpy. Apabila Anda berniat ingin menggunakan *module* lain, harap konfirmasikan ke narahubung terlebih dahulu (bisa saja diperbolehkan).
3. Narahubung untuk Tugas 5 Praktikum Metode Numerik adalah:

Bisma Rohpanca Joyosumarto (ID LINE: bisma\_joyosumarto)

Silakan hubungi narahubung di atas apabila ada yang ingin ditanyakan atau dikonfirmasikan.

**Soal Tugas 5 Metode Numerik**

Sebelum memulai pengerjaan tugas ini, harap hitung nilai k berikut ini berdasarkan NPM Anda:

**k = (NPM) mod 3**

Pada petunjuk tugas berikut ini, akan digunakan istilah **“metode langsung”**.

* Jika k=0, yang dimaksud dengan “metode langsung” adalah eliminasi Gauss (tanpa strategi pivoting).
* Jika k=1, yang dimaksud dengan “metode langsung” adalah *partial pivoting*.
* Jika k=2, yang dimaksud dengan “metode langsung” adalah *scaled partial pivoting*.

Diberikan sistem persamaan linier (SPL) berikut:

**Soal bagian wajib**

Buatlah suatu *file* .ipynb dan lakukan hal-hal berikut ini di dalamnya:

1. Gunakan **metode langsung** dan substitusi balik untuk menyelesaikan SPL di atas, tanpa *rounding* (pembulatan) sama sekali. Tampilkan (print) hasil **metode langsung** (yaitu hasil bentuk *triangular*) dan hasil solusinya.
2. Buatlah fungsi baru yang merupakan modifikasi dari fungsi **metode langsung** agar selalu melakukan pembulatan (hingga tiga digit di belakang koma) untuk tiap perhitungan, yaitu:

* untuk hasil perhitungan *multiplier* m; dan
* ketika melakukan operasi baris elementer (OBE), kecuali pertukaran baris.

1. Buatlah fungsi baru yang merupakan modifikasi dari fungsi substitusi balik agar selalu melakukan pembulatan (hingga tiga digit di belakang koma) untuk tiap perhitungan, yaitu:

* ketika menghitung sumasi; dan
* ketika membagi dengan elemen diagonal (termasuk untuk baris paling bawah yang paling awal dihitung).

1. Gunakan fungsi modifikasi **metode langsung** dan fungsi modifikasi substitusi balik (yaitu kedua fungsi yang telah Anda buat di soal no. 2 dan no. 3) untuk menyelesaikan SPL di atas. Tampilkan hasil **metode langsung** dan hasil solusinya.
2. Bandingkan hasil perhitungan dari soal no. 1 dan soal no. 4. Hitung selisihnya, kemudian dibuat nilai mutlak agar tidak negatif. Nilai ini disebut “round-off error”. Tampilkan *round-off error* tersebut.

**Catatan:**

* Pembulatan bisa dilakukan menggunakan **numpy.around(sesuatu, 3)**, untuk membulatkan hingga tiga digit di belakang koma.
* Perhitungan nilai mutlak bisa menggunakan **numpy.absolute(sesuatu)**.
* Notasi ilmiah: “e+k” artinya dikali , “e+0” artinya dikali 1, dan “e-k” artinya dikali , untuk apapun nilai k. Apabila *output* program Anda tidak menggunakan notasi ilmiah sedangkan contoh *output* di bawah ini menggunakan notasi ilmiah, atau apabila sebaliknya, itu tidak masalah selama nilainya sebenarnya sama. Contohnya, 1.530e+01 dan 15.3 itu sama saja, karena 1.530e+01 = .
* Elemen matriks diperbesar yang krusial untuk proses substitusi balik adalah elemen diagonal dan di atas diagonal. Apabila ada ketidakcocokkan hasil modifikasi **metode langsung** milik Anda dibandingkan dengan contoh *output* di bawah, tetapi ketidakcocokkan tersebut hanya untuk elemen di bawah diagonal, maka kemungkinan program Anda sudah benar. Elemen di bawah diagonal memang cenderung aneh (biasanya nol tetapi bisa jadi taknol) karena masalah *rounding*, tetapi pada akhirnya tidak mempengaruhi perhitungan solusi, sehingga tidak masalah.
* Indikator yang paling mudah untuk memastikan apakah program Anda sudah benar adalah langsung mencocokkan nilai *round-off error* dengan contoh *output* di bawah.

Pada soal bagian wajib, telah ditentukan bahwa pembulatan dilakukan hingga tiga digit di belakang koma. Ketentuan itu disebut “precision”. Pada soal di atas, precision=3 (tiga, karena diminta pembulatan hingga tiga digit di belakang koma).

**Soal bagian bonus**

Buatlah suatu program (suatu “fungsi utama”) yang bisa menerima input sembarang matriks diperbesar dan sembarang “*precision*”, kemudian melakukan perhitungan seperti soal no. 1, no. 4, dan no. 5 di atas.

**Contoh *output* program untuk wajib+bonus (jika k=0):**

| NPM saya 2106635580 sehingga k = 0.  Karena k = 0, akan digunakan eliminasi Gauss tanpa strategi pivoting.  Masukkan SPL yang ingin dicoba: [ [1.19, 2.11, -100, 1, 1.12], [14.2, -0.122, 12.2, -1, 3.44], [0, 100, -99.9, 1, 2.15], [15.3, 0.110, -13.1, -1, 4.16] ]  Akan diselesaikan SPL dengan matriks diperbesar sebagai berikut:  [[ 1.19 2.11 -100. 1. 1.12 ]  [ 14.2 -0.122 12.2 -1. 3.44 ]  [ 0. 100. -99.9 1. 2.15 ]  [ 15.3 0.11 -13.1 -1. 4.16 ]]  Akan dicoba tanpa dan dengan pembulatan.  Masukkan precision (sekian angka di belakang koma): 3  Tanpa pembulatan, eliminasi Gauss menghasilkan:  [[ 1.19000000e+00 2.11000000e+00 -1.00000000e+02 1.00000000e+00  1.12000000e+00]  [ 0.00000000e+00 -2.53001513e+01 1.20547731e+03 -1.29327731e+01  -9.92470588e+00]  [ 0.00000000e+00 0.00000000e+00 4.66480397e+03 -5.01173747e+01  -3.70778520e+01]  [ 0.00000000e+00 0.00000000e+00 0.00000000e+00 -2.04328072e-01  2.41640155e-01]]  Kemudian, substitusi balik menghasilkan solusi:  [ 0.1768253 0.01269269 -0.02065405 -1.1826087 ]  Dengan pembulatan, eliminasi Gauss menghasilkan:  [[ 1.190000e+00 2.110000e+00 -1.000000e+02 1.000000e+00 1.120000e+00]  [-0.000000e+00 -2.530100e+01 1.205500e+03 -1.293300e+01 -9.925000e+00]  [ 0.000000e+00 1.000000e-02 4.664236e+03 -5.011100e+01 -3.707400e+01]  [ 0.000000e+00 3.000000e-03 -8.810000e-01 -1.950000e-01 2.490000e-01]]  Kemudian, substitusi balik dengan pembulatan menghasilkan solusi:  [ 0.171 -0.003 -0.022 -1.277]  Round-off error adalah mutlak selisih di antara kedua hasil solusi, yaitu:  [0.0058253 0.01569269 0.00134595 0.0943913 ] |
| --- |

**Contoh *output* program untuk wajib+bonus (jika k=1):**

| NPM saya 2106635581 sehingga k = 1.  Karena k = 1, akan digunakan partial pivoting.  Masukkan SPL yang ingin dicoba: [ [1.19, 2.11, -100, 1, 1.12], [14.2, -0.122, 12.2, -1, 3.44], [0, 100, -99.9, 1, 2.15], [15.3, 0.110, -13.1, -1, 4.16] ]  Akan diselesaikan SPL dengan matriks diperbesar sebagai berikut:  [[ 1.19 2.11 -100. 1. 1.12 ]  [ 14.2 -0.122 12.2 -1. 3.44 ]  [ 0. 100. -99.9 1. 2.15 ]  [ 15.3 0.11 -13.1 -1. 4.16 ]]  Akan dicoba tanpa dan dengan pembulatan.  Masukkan precision (sekian angka di belakang koma): 3  Tanpa pembulatan, partial pivoting menghasilkan:  [[ 15.3 0.11 -13.1 -1. 4.16 ]  [ 0. 100. -99.9 1. 2.15 ]  [ 0. 0. -96.88176811 1.05676333 0.75126339]  [ 0. 0. 0. 0.19359673 -0.22894918]]  Kemudian, substitusi balik menghasilkan solusi:  [ 0.1768253 0.01269269 -0.02065405 -1.1826087 ]  Dengan pembulatan, partial pivoting menghasilkan:  [[ 1.530e+01 1.100e-01 -1.310e+01 -1.000e+00 4.160e+00]  [ 0.000e+00 1.000e+02 -9.990e+01 1.000e+00 2.150e+00]  [-3.000e-03 1.000e-03 -9.688e+01 1.057e+00 7.510e-01]  [ 1.000e-03 -2.400e-02 3.400e-02 1.930e-01 -2.290e-01]]  Kemudian, substitusi balik dengan pembulatan menghasilkan solusi:  [ 0.176 0.012 -0.021 -1.187]  Round-off error adalah mutlak selisih di antara kedua hasil solusi, yaitu:  [0.0008253 0.00069269 0.00034595 0.0043913 ] |
| --- |

**Contoh *output* program untuk wajib+bonus (jika k=2):**

| NPM saya 2106635582 sehingga k = 2.  Karena k = 2, akan digunakan scaled partial pivoting.  Masukkan SPL yang ingin dicoba: [ [1.19, 2.11, -100, 1, 1.12], [14.2, -0.122, 12.2, -1, 3.44], [0, 100, -99.9, 1, 2.15], [15.3, 0.110, -13.1, -1, 4.16] ]  Akan diselesaikan SPL dengan matriks diperbesar sebagai berikut:  [[ 1.19 2.11 -100. 1. 1.12 ]  [ 14.2 -0.122 12.2 -1. 3.44 ]  [ 0. 100. -99.9 1. 2.15 ]  [ 15.3 0.11 -13.1 -1. 4.16 ]]  Akan dicoba tanpa dan dengan pembulatan.  Masukkan precision (sekian angka di belakang koma): 3  Tanpa pembulatan, scaled partial pivoting menghasilkan:  [[ 1.42000000e+01 -1.22000000e-01 1.22000000e+01 -1.00000000e+00  3.44000000e+00]  [ 0.00000000e+00 1.00000000e+02 -9.99000000e+01 1.00000000e+00  2.15000000e+00]  [ 0.00000000e+00 -2.77555756e-17 -2.60038612e+01 7.50502817e-02  4.48329937e-01]  [ 0.00000000e+00 1.05566843e-16 0.00000000e+00 7.77150858e-01  -9.19065362e-01]]  Kemudian, substitusi balik menghasilkan solusi:  [ 0.1768253 0.01269269 -0.02065405 -1.1826087 ]  Dengan pembulatan, scaled partial pivoting menghasilkan:  [[ 1.4200e+01 -1.2200e-01 1.2200e+01 -1.0000e+00 3.4400e+00]  [ 0.0000e+00 1.0000e+02 -9.9900e+01 1.0000e+00 2.1500e+00]  [ 7.0000e-03 4.1000e-02 -2.6039e+01 7.5000e-02 4.5100e-01]  [-3.0000e-02 -1.3600e-01 -5.0000e-03 7.7800e-01 -9.2700e-01]]  Kemudian, substitusi balik dengan pembulatan menghasilkan solusi:  [ 0.176 0.012 -0.021 -1.192]  Round-off error adalah mutlak selisih di antara kedua hasil solusi, yaitu:  [0.0008253 0.00069269 0.00034595 0.0093913 ] |
| --- |