**Tugas 4 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 4 adalah:

**Nama Lengkap\_NPM\_Kelas SIAK\_Tugas4PrakMetnum.ipynb**

Contoh penamaan yang benar:

Johan Frederik Steffensen\_2234567890\_Kelas F\_Tugas4PrakMetnum.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 4 ini adalah **2 (dua) minggu**, dan tenggat waktu (*deadline*) pengumpulan Tugas 4 ini adalah **Sabtu, 27 Mei 2023, pukul 23.59 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 4 ini hanyalah numpy, tabulate, dan sympy. Apabila Anda berniat ingin menggunakan *module* lain, harap konfirmasikan ke narahubung terlebih dahulu (bisa saja diperbolehkan).
3. Narahubung untuk Tugas 4 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 4 Metode Numerik**

Diberikan fungsi berikut, yang menerima dua nilai a, b. Fungsi I(a,b) ini merupakan suatu integral tentu, di mana pengintegralan dilakukan pada interval :

Sayangnya, **integral tersebut tidak bisa dihitung secara analitik/kalkulus\***. Istilahnya, integral tersebut merupakan *non-elementary integral* atau *non-elementary antiderivative* (fungsi yang diintegralkan tidak memiliki *antiderivative* yang eksplisit).

\*kebetulan ada pengecualian untuk beberapa interval, yaitu ada solusi analitik khusus pengintegralan pada interval , interval , dan interval saja.

**Anda perlu menghitung I(-2/5, k/5)**, yaitu Anda perlu menghitung nilai integral berikut:

di mana **k = (NPM) mod 10.**

**Soal bagian wajib**

Buatlah program untuk menghitung nilai integral tersebut secara numerik, menggunakan metode integrasi numerik berikut ini.

1. Gunakan metode Newton-Cotes, baik *closed* (dengan n=1,2,3,4) maupun *open* (dengan n=0,1,2,3). Jabarkan hasilnya untuk tiap variasi nilai n, serta tuliskan nama metodenya kalau ada (misalnya, untuk *closed* Newton-Cotes dengan n=1, tuliskan “trapezoidal rule”). Kemudian, ringkas semua hasilnya dalam satu tabel.
2. Gunakan metode integrasi komposit berikut ini, dengan banyaknya subinterval besar :

* metode Simpson komposit (banyaknya subinterval kecil );
* metode Trapezoidal komposit (banyaknya subinterval kecil , sehingga “subinterval kecil” dan “subinterval besar” adalah sama); dan
* metode Midpoint komposit (banyaknya subinterval kecil ).

(Anda boleh memilih antara menggunakan rumus umum komposit atau rumus khusus.)

1. Gunakan metode kuadratur adaptif dengan metode Simpson (metode Simpson adaptif), dengan toleransi 10-4 dan faktor/pengkali toleransi = 10.
2. Gunakan metode kuadratur Gauss dengan bilangan bulat positif n=4.

Selain itu, ikuti juga ketentuan berikut ini:

1. Program bisa menerima sembarang fungsi, sembarang NPM, dan sembarang interval untuk melakukan integrasi numerik pada interval tersebut dengan semua metode di atas, serta:

* sembarang banyaknya subinterval besar untuk metode komposit,
* sembarang toleransi dan pengkali toleransi untuk metode Simpson adaptif, dan
* sembarang bilangan bulat positif n untuk kuadratur Gauss.

1. Keseluruhan program dikemas di dalam satu *function* utama yang bisa di-*run* di *cell* terpisah.

**Soal bagian bonus** (tidak wajib; hanya untuk menaikkan nilai apabila jawaban pada soal bagian wajib belum sempurna)

1. Diberikan fungsi baru bernama , di mana , sehingga

Fungsi ini disebut ***error function***. Pada *cell* terpisah, gunakan program yang telah Anda buat untuk mengaproksimasi nilai secara numerik. (Anda boleh mengubah batasan integral, memperbanyak partisi/subinterval untuk metode komposit, mengubah toleransi maupun pengkali toleransi untuk kuadratur adaptif, memperbesar nilai n untuk kuadratur Gauss, dan sebagainya.) Menurut Anda, dari hasil aproksimasi beberapa metode, hasil yang mana saja yang paling akurat? Kira-kira berapa nilai sesungguhnya? Ketik penjelasan Anda (jelaskan secara singkat) di *file* .ipynb yang sama, pada *cell* di bawahnya lagi.

1. Gunakan sympy.erf(numpy.inf) untuk mendapatkan solusi analitik dari . Bandingkan dengan hasil aproksimasi yang telah Anda dapatkan: apakah ada hasil aproksimasi yang sama dengan solusi analitik, atau setidaknya mendekati? Jelaskan secara singkat di *file* .ipynb yang sama.
2. Menggunakan solusi analitik tersebut, apakah Anda bisa menentukan solusi analitik untuk integral tentu berikut ini (tidak lagi dikalikan )? Kalau bisa, tentukan. Jelaskan secara singkat di *file* .ipynb yang sama.

(*Fun fact*: integral ini berkaitan dengan suatu integral istimewa yang disebut *Gaussian integral*.)

**Catatan**:

* Dibandingkan dengan metode komposit, metode kuadratur adaptif, maupun metode kuadratur Gauss, ada kecenderungan bahwa semua metode Newton-Cotes (baik *closed* maupun *open*) tidak begitu akurat untuk interval yang besar.
* Mencoba nilai bilangan bulat positif n yang sangat besar untuk orde kuadratur Gauss bisa saja mengakibatkan program berjalan dengan sangat lama apabila akar-akar dan koefisien polinom Legendre diperoleh dengan cara manual.
* Untuk soal bagian bonus, Anda diperbolehkan menggunakan *software* untuk plot fungsi, seperti Desmos atau Wolfram Mathematica, untuk melihat bentuk fungsi, agar Anda terbantu dalam menentukan nilai yang sesuai untuk batasan integral, banyaknya partisi, dll. Kemudian, ketika menulis penjelasan, Anda boleh menjelaskan bagaimana Anda menentukan nilai-nilai yang sesuai berdasarkan plot yang Anda lihat.

**Contoh *output* program: bagian wajib**

| >>> tugas4metnum()  Program ini akan mengaproksimasi integral f(t) pada interval yang ditentukan,  dengan beberapa metode Newton-Cotes, metode komposit, Simpson adaptif, dan kuadratur Gauss.  Masukkan NPM Anda: 2106635589  Maka, k = 9  Berikut ini, akan diminta beberapa input. Semua boleh bergantung nilai k.  Masukkan rumus f(t): 2/sqrt(pi) \* exp(-t\*\*2)  Masukkan batas bawah integral: -2/5  Masukkan batas atas integral: k/5  Masukkan banyaknya subinterval besar N untuk metode komposit: k+2  Masukkan toleransi (epsilon) untuk metode Simpson adaptif: 10\*\*(-4)  Masukkan pengkali toleransi untuk metode Simpson adaptif: 10  Masukkan orde kuadratur Gauss (bilangan bulat positif n): 4  Berikut hasil aproksimasi integral dengan closed Newton-Cotes:  n=1: 1.106306324388451 (Trapezoidal rule)  n=2: 1.3826385700649404 (Simpson's rule)  n=3: 1.4079799354890195 (Simpson's Three-Eights rule)  n=4: 1.422883671030931 (Boole's rule)  Berikut hasil aproksimasi integral dengan open Newton-Cotes:  n=0: 1.5208046929031849 (Midpoint rule)  n=1: 1.5085378058558756  n=2: 1.458098134376173  n=3: 1.4414550991781234  Hasil tabel metode Newton-Cotes:  +--------+--------------------+--------------------+--------------------+--------------------+-------------------+  | n | 0 | 1 | 2 | 3 | 4 |  +--------+--------------------+--------------------+--------------------+--------------------+-------------------+  | closed | | 1.106306324388451 | 1.3826385700649404 | 1.4079799354890195 | 1.422883671030931 |  | open | 1.5208046929031849 | 1.5085378058558756 | 1.458098134376173 | 1.4414550991781234 | |  +--------+--------------------+--------------------+--------------------+--------------------+-------------------+  Hasil integrasi numerik komposit dengan 11 partisi:  1.4174845534240823 (Metode Simpson Komposit)  1.414381675987464 (Metode Trapezoidal Komposit)  1.4190359921423914 (Metode Midpoint Komposit)  Hasil Simpson Adaptif:  1.4174889915706232  dihitung dengan toleransi 0.0001 dan faktor/pengkali toleransi 10  Hasil kuadratur Gauss dengan n = 4:  1.41804559794528 |
| --- |

(Soal bagian bonus hanya memanfaatkan program yang sudah dibuat dari bagian wajib, sehingga kami tidak memberikan contoh *output* untuk soal bagian bonus.)