mirza nur hidayat

fisika komputasi
(modul kuliah)

program studi pendidikan fisika
fkip uhamka

2021

Pengantar

Bismillahirrahmanirrahim

Fisika komputasi merupakan satu dari tiga pilar dalam fisika, selain teori dan eksperimen. Fisika komputasi juga dapat dikatakan sebagai irisan dari tiga disiplin ilmu, yaitu fisika (kasus), matematika (numerik), dan ilmu komputer (pemrograman).

Dokumen ini merupakan modul kuliah fisika komputasi untuk mahasiswa semester 3 di Program Studi Pendidikan Fisika FKIP UHAMKA.

Modul terdiri atas empat bagian, yaitu persamaan nonlinear, diferensiasi numerik, integrasi numerik, serta persamaan diferensial. Untuk setiap bagiannya, praktis hanya disajikan satu metode numerik yang digunakan dalam proses komputasi. Metode Bisection dalam persamaan nonlinear, forward difference dalam diferensiasi numerik, metode Trapezoid dalam integrasi numerik, serta metode Euler dalam persamaan diferensial. Namun demikian, di akhir setiap bagiannya mahasiswa diberikan saran untuk mempelajari dan mencoba metode yang lainnya.

Struktur modul di setiap bagiannya yaitu teori numerik, koding, serta projek. Dengan harapan, mahasiswa lebih mudah memahami dan lebih kreatif dalam proses komputasi. Proses komputasi yang dimaksud meliputi kasus fisika, analisis matematis, analisis numerik, algoritma, koding, data dan visualisasi, serta analisis hasil komputasi.

Tidak seperti pada modul untuk kakak-kakak kelas sebelumnya yang menggunakan bahasa pemrograman Fortran, Scilab, atau Octave, modul ini dibuat dengan bahasa Python. Tidak lupa pula, sebagai mahasiswa program studi pendidikan yang notebene akan menjadi guru di sekolah kelak, mahasiswa juga dikenalkan dengan bahasa pemrograman berbasis blok, yaitu Blockly atau Code with Chrome. Hal ini merupakan bagian dari tren pembelajaran STEM dan *computational thinking* bagi anak-anak sekolah.

Modul mengacu pada modul yang pernah penulis buat sebelumnya, yaitu Fisika Komputasi dengan Fortran 95 dan gnuplot (2015) serta Modul Praktikum Fisika Komputasi (2020).

Adapun referensi modul yaitu Introductory Computational Physics karya Andi Klein dan Alexander Godunov (Cambridge University Press, 2006) dan Computational Quantum Mechanics karya Joshua Izaac dan Jingbo Wang (Springer, 2018).

Last but not the least, dokumen ini dapat digunakan secara bebas untuk tujuan pendidikan dan pembelajaran.

Semoga bermanfaat.

Jakarta, 27 Muharram 1443 / 4 September 2021
–Mirza N. Hidayat

Daftar Isi

Modul 3 Integrasi Numerik

Modul 4 Persamaan Diferensial

Modul 1

Persamaan Nonlinear

Sebuah fungsi $f(x) \equiv ax - b = 0$, maka dengan cepat didapat bahwa $x = \frac{b}{a}$. Pun demikian dengan fungsi $f(x) \equiv ax^2 + bx + c = 0$, maka penyelesaian x adalah

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} \tag{1.1}$$

Lalu, bagaimanakah jika fungsi $f(x)$ adalah fungsi polinomial, transedental, atau kombinasi dari berbagai variasi fungsi? Persoalan ini dapat diselesaikan dengan pendekatan numerik. Salah satu metode yang digunakan adalah metode Bisection.

Diasumsikan fungsi $f(x)$ kontinyu di titik a dan b. Maka prinsip metode Bisection adalah masukkan nilai a dan b. Kemudian tentukan titik tengah antara a dan b, misal titik tengah ini ditulis sebagai x_1, sehingga $x_1 = \frac{a+b}{2}$. Langkah berikutnya adalah tentukan $f(a)f(x_1)$. Ada tiga sifat dari perkalian ini, yaitu

$$f(a)f(x_1) = \begin{cases} < 0, & \text{ada akar di } [a, x_1] \\ > 0, & \text{ada akar di } [x_1, b] \\ = 0, & x_1 \text{ akar penyelesaian} \end{cases} \tag{1.2}$$

Ulangi langkah di atas sampai diperoleh akar persamaannya.

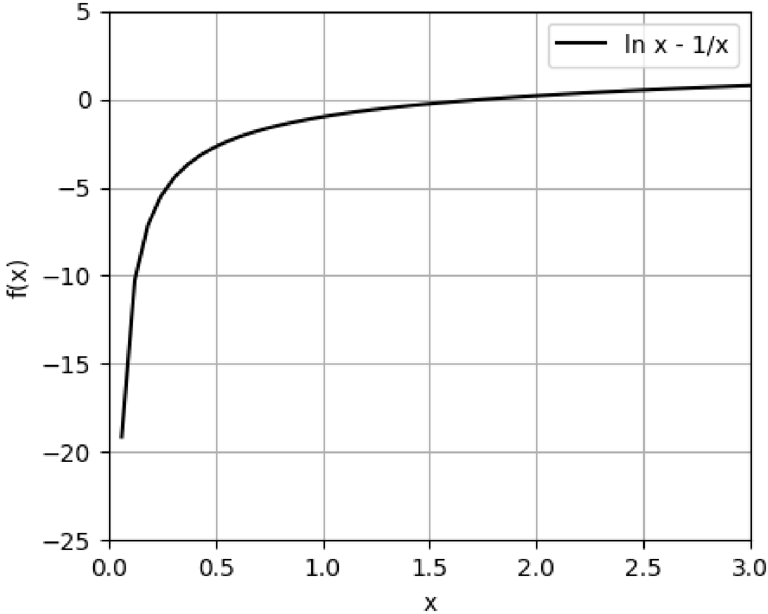
Koding 1. Diberikan sebuah fungsi

$$f(x) \equiv \ln x - \frac{1}{x} = 0 \tag{1.3}$$

Tentukan penyelesaian dari fungsi di atas.

```
import numpy as np

def f(x):
   return np.log(x) - (1 / x)

iteration = 0
max_iteration  = 100
```

```
a = 1
b = 3

while True:
   iteration += 1

   x1 = (a + b) / 2

   if f(a) * f(x1) < 0:
       b = x1
   elif f(a) * f(x1) > 0:
       a = x1
   else:
       break

   if (a >= b):
       print('Error')

   if iteration == max_iteration:
       break

x = x1

print(x)
```



Grafik fungsi $f(x) = \ln x - \frac{1}{x}$. Dari hasil komputasi dengan metode Bisection didapat bahwa $f(x) = 0$ pada $x = x_1$.

Projek 1. Pilih sembarang fungsi nonlinear. Lalu selesaikan fungsi tersebut dan plot grafiknya.

Saran 1. Silakan juga dipelajari dan dicoba metode Secant dan metode Newton-Raphson.

Modul 2

Diferensiasi Numerik

Sebuah fungsi $f(x)$ dapat ditulis ke dalam deret Taylor di sekitar $x + h$:

$$f(x+h) = f(x) + \frac{h}{1!}f'(x) + \frac{h^2}{2!}f''(x) + \frac{h^3}{3!}f'''(x) + ... \quad (2.1)$$

Selanjutnya, dari (2.1) dapat ditentukan atau dicari nilai $f'(x)$ yaitu

$$f'(x) = \frac{1}{h}\left(f(x+h) - f(x) - \frac{h^2}{2!}f''(x) - \frac{h^3}{3!}f'''(x) - ...\right) \quad (2.2)$$

Dalam bentuk lain, (2.2) dapat ditulis sebagai

$$f'(x) = \frac{f(x+h) - f(x)}{h} - \left(\frac{h}{2!}f''(x) + \frac{h^2}{3!}f'''(x) + ...\right) \quad (2.3)$$

Sehingga didapatkan pendekatan

$$f'(x) \approx \frac{f(x+h) - f(x)}{h} \tag{2.4}$$

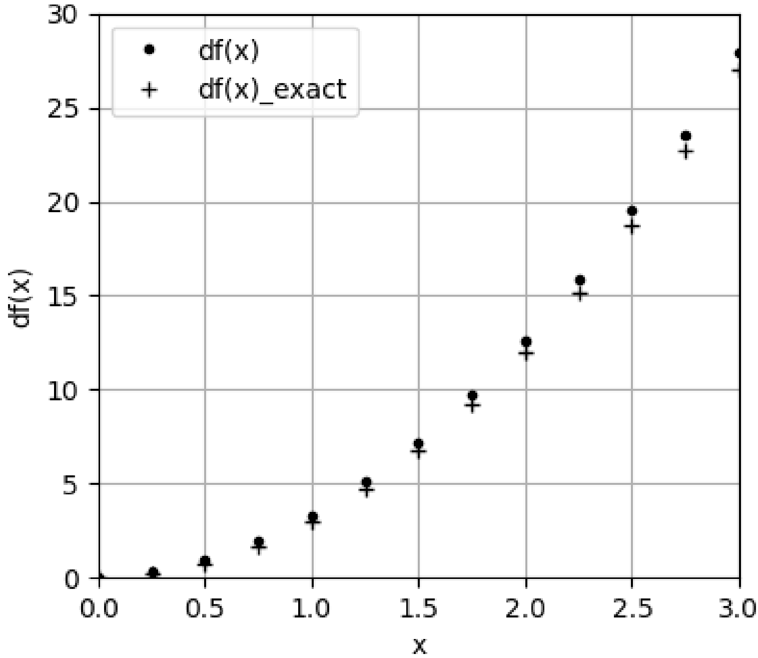
Pendekatan (2.4) dapat digunakan sebagai formula dalam komputasi diferensiasi numerik dan dikenal sebagai metode forward difference.

Koding 2.1. Diberikan sebuah fungsi $f(x) = x^2$. Tentukan nilai $f'(3)$ secara komputasi dengan metode forward difference.

```
def f(x):
   return x ** 2

x = 3
h = 0.1

df = (f(x + h) - f(x)) / h

print(df)
```

Koding 2.2. Diberikan sebuah fungsi $f(x) = x^3$. Plot grafik $f'(x)$ dengan $0 \le x \le 3$ secara komputasi dan eksak.

```
import numpy as np
import matplotlib.pyplot as plt

def f(x):
   return x ** 3

x = np.linspace(0, 3, 13)
h = 0.1

df = (f(x + h) - f(x)) / h
df_exact = 3 * x ** 2

plt.plot(x, df, 'k.', x, df_exact, 'k+')
plt.axis([0, 3, 0, 30])
plt.title('')
plt.legend(['df(x)', 'df(x)_exact'])
plt.xlabel('x')
plt.ylabel('df(x)')
plt.grid(True)
plt.show()
```



Projek 2. Diketahui sebuah fungsi $f(x) = 2^x + \sin x$. Jika diketahui $0 \leq x \leq \pi$, plot grafik $f'(x)$ secara komputasi dan eksak. Beri analisis errornya.

Saran 2. Silakan juga dipelajari dan dicoba metode central dan backward difference, serta diferensiasi numerik orde dua.

Modul 3

Integrasi Numerik

Sebuah fungsi $f(x)$ dengan $a \leq x \leq b$, maka hasil integral

$$\int_a^b f(x)\,\mathrm{d}x \tag{3.1}$$

adalah luasan di bawah grafik $f(x)$ dengan batas bawah a dan batas atas b.

Dalam teori analisis numerik, solusi integral tersebut dapat diselesaikan dengan menggunakan pendekatan

$$\int_a^b f(x)\,\mathrm{d}x \approx \frac{h}{2}\left(f(a) + f(b)\right) \tag{3.2}$$

dengan $h = b - a$. Pendekatan ini dikenal sebagai metode Trapezoid.

Koding 3.1. Tentukan integral berikut dengan metode Trapezoid.

$$\int_{2}^{5} x^2 \, \mathrm{d}x \tag{3.3}$$

```
def f(x):
   return x ** 2

a = 2
b = 5
h = b - a

F = (h / 2) * (f(a) + f(b))

print(F)
```

Bandingkan output komputasi di atas dengan hasil eksak. Terdapat selisih atau eror bukan? Guna memperkecil eror, maka dibuatlah metode Trapezoid dengan jumlah cacah n.

Koding 3.2. Koding ini merupakan solusi komputasi (3.3) dengan metode Trapezoid dengan jumlah cacah $n = 3$.

```
def f(x):
   return x ** 2

a = 2
b = 5
n = 3
h = (b - a) / n

F = 0

for i in range(1, n + 1):
   ai = (a - h) + (i * h)
   bi = ai + h
   Fi = (h / 2) * (f(ai) + f(bi))
   F = F + Fi

print(F)
```

Projek 3. Tentukan integral berikut dengan metode Trapezoid dengan jumlah cacah sembarang n.

$$\int_{-2}^{2} \frac{1}{\sqrt{4 + x^2}} \, \mathrm{d}x \tag{3.4}$$

Tidak lupa solusi eksak dan analisis erornya.

Saran 3. Silakan juga dipelajari dan dicoba metode Midpoint dan metode Simpson.

Modul 4

Persamaan Diferensial

Persamaan diferensial biasa atau ordinary differential equations (ODEs) sering digunakan dalam dunia sains dan teknik, tak terkecuali fisika. ODEs dapat ditulis dalam bentuk

$$u'(t) = f(u(t), t) \tag{4.1}$$

Asumsikan (4.1) pada saat $t = t_i$ maka (4.1) menjadi

$$u'(t_i) = f(u(t_i), t_i) \tag{4.2}$$

Pendekatan forward difference

$$u'(t_i) \approx \frac{u(t_{i+1}) - u(t_i)}{t_{i+1} - t_i} \tag{4.3}$$

atau

$$u'(t_i) \approx \frac{u(t_{i+1}) - u(t_i)}{\Delta t} \tag{4.4}$$

Dari (4.2) dan (4.4) didapat

$$\frac{u(t_{i+1}) - u(t_i)}{\Delta t} \approx f(u(t_i), t_i) \tag{4.5}$$

sehingga

$$u(t_{i+1}) \approx u(t_i) + \Delta t f(u(t_i), t_i) \tag{4.6}$$

Persamaan (4.6) dikenal sebagai metode Euler.

Koding 4. Tentukan penyelesaian persamaan diferensial berikut secara komputasi dengan menggunakan metode atau algoritma Euler. Asumsikan bahwa $y(0) = 1$.

$$y'(t) = y(t) \tag{4.7}$$

```
import numpy as np
import matplotlib.pyplot as plt

dt = 0.1
n = 10

t = np.zeros(n + 1)
y = np.zeros(n + 1)

t[0] = 0
y[0] = 1

for i in range(n):
   t[i + 1] = t[i] + dt
   y[i + 1] = y[i] + dt * y[i]

plt.plot(t, y, 'k')
plt.show()
```

Projek 4.1. Diberikan sebuah kasus fisika sistem osilator harmonik sederhana. Sistem terdiri atas benda bermassa m yang diikatkan secara horisontal ke pegas dengan konstanta k. Ujung pegas yang lain terikat pada dinding. Sistem berosilasi secara harmonik sederhana. Problem yang akan diselesaikan dari sistem ini yaitu bagaimana visualisasi perilaku posisi x dan kecepatan v sebagai fungsi waktu t.

Problem di atas secara numerik dapat diselesaikan dengan memanfaatkan algoritma Euler.

Telah diketahui bersama bahwa gaya F memiliki persamaan $F = ma$ serta hubungan $F = -kx$. Dari kedua persamaan ini didapatkan

$$a = \frac{\mathrm{d}v}{\mathrm{d}t} = -\frac{k}{m}x \tag{4.8}$$

Jika $k = m = 1$, maka $a = -x$. Dengan pendekatan

$$\frac{\mathrm{d}v(t)}{\mathrm{d}t} = \frac{v(t+\Delta t) - v(t)}{\Delta t} \tag{4.9}$$

maka

$$a(t) = \frac{v(t+\Delta t) - v(t)}{\Delta t} \tag{4.10}$$

atau

$$v(t+\Delta t) = v(t) + a(t)\Delta t \tag{4.11}$$

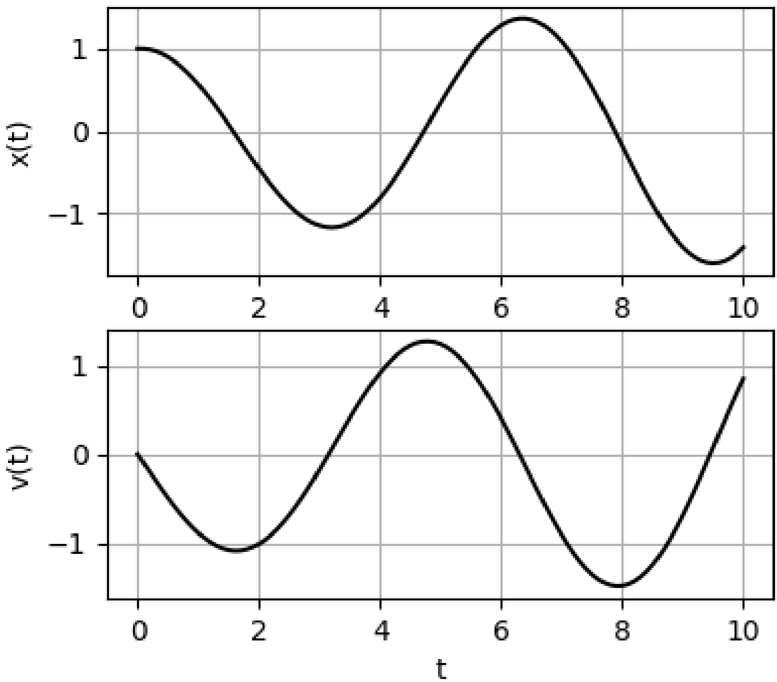
Karena $a = -x$ maka (4.11) dapat ditulis menjadi

$$v(t+\Delta t) = v(t) - x(t)\Delta t \tag{4.12}$$

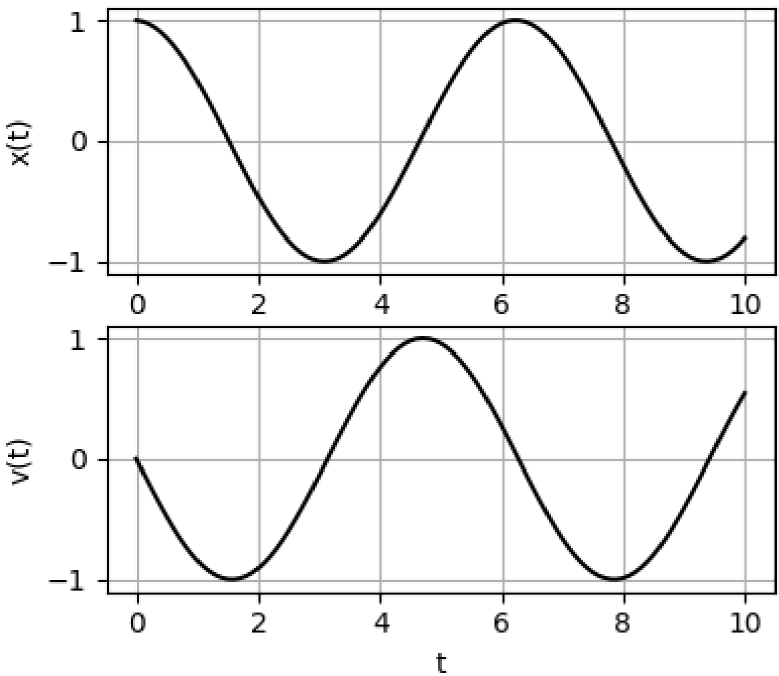
Dengan pendekatan yang sama didapatkan

$$x(t + \Delta t) = x(t) + v(t)\Delta t \tag{4.13}$$

Gunakan (4.12) dan (4.13) dalam komputasi sehingga didapatkan grafik visualisasi sistem osilator harmonik sederhana berikut.



Projek 4.2. Perhatikan kembali (4.12) dan (4.13). Modifikasi $v(t)$ pada (4.13) menjadi $v(t + \Delta t)$. Gunakan persamaan yang baru ini dalam komputasi sehingga didapatkan grafik berikut. Modifikasi ini dikenal sebagai algoritma Euler-Cromer.



Projek 4.3. Perhatikan kembali Projek 4.1 dan 4.2. Dengan menggunakan algoritma Euler-Cromer, lakukan proses komputasi sehingga didapatkan visualisasi 2D energi mekanik dari sistem tersebut.

Projek 4.4. Diberikan materi fisika tentang sistem sirkuit RC (resistor kapasitor). Persamaan sistem yaitu

$$Ri + \frac{1}{C}\int i \, \mathrm{d}t = V \tag{4.14}$$

Buktikan secara analitis matematis bahwa penyelesaian persamaan arus listrik $i(t)$ sebagai fungsi waktu t dari (4.14) yaitu

$$i(t) = \frac{V}{R}e^{-t/RC} \tag{4.15}$$

Dari (4.15), buat grafik 2D-nya (grafik solusi eksak). Selanjutnya, (4.14) dapat juga diselesaikan secara numerik dengan persamaan diferensial

$$R\frac{\mathrm{d}i}{\mathrm{d}t} + \frac{i}{C} = 0 \tag{4.16}$$

Dari (4.16), buat grafik 2D-nya dengan algoritma Euler (grafik solusi komputasi). Beri analisis dari kedua grafik yang telah diperoleh.

Projek 4.5. Dengan petunjuk yang sama seperti pada Projek 4.4, selesaikan sistem RL (resistor induktor). Adapun persamaan sistem (sekaligus sebagai persamaan diferensial) dan persamaan arus secara berturut-turut disajikan dalam (4.17) dan (4.18).

$$Ri + L\frac{\mathrm{d}i}{\mathrm{d}t} = V \tag{4.17}$$

$$i(t) = \frac{V}{R}\left(1 - e^{-tR/L}\right) \tag{4.18}$$

Saran 4. Silakan juga dipelajari dan dicoba metode Runge-Kutta.

–o0o–