

Diktat Praktikum

KOMPUTASI DAN SIMULASI FISIKA

UND 2025 14004/Semester 4

Oleh:

Dr. Budi Adiperdana, M.Si.

Ferry Faizal, Ph.D.

Laboratorium Komputasi

Departemen Fisika

Fakultas Matematika dan Ilmu Pengetahuan Alam

Universitas Padjadjaran

Jatinangor, Februari 2026

MODUL PRAKTIKUM KOMPUTASI DAN SIMULASI FISIKA

Copyright © 2026 – UND202514004 / 3rd Semester, Laboratorium of Computation.

Diktat ini merupakan panduan bagi mahasiswa S-1 *Fisika Unpad* dalam memahami mata kuliah komputasi dan simulasi fisika secara praktek.

- *Departemen Fisika, Universitas Padjadjaran*

Modul-1: Model Persamaan Diferensial

Modul-2: Stabilitas Numerik dan Resolusi Diskritisasi

Modul-3: Skema Waktu: Metode Eksplisit dan Implisit

Modul-4: Gerak Partikel, Syarat Batas dan Gerak Realistik (Air drag, Air density)

Modul-5: Gerak Osilasi (Bandul dan Pegas)

Modul-6: Bilangan Acak dan Fenomena Stokastik (Integral di kelas, Random Walk)

Modul-7: Persamaan Medan: Laplace dan Poisson

Modul-8: Persamaan Evolusi Kontinu: Difusi

Modul-9: Persamaan Evolusi Kontinu: Gelombang

Projek Simulasi

MODUL – 2: Stabilitas Numerik dan Resolusi Diskritisasi

1. Tujuan Praktikum

Setelah menyelesaikan praktikum ini, mahasiswa diharapkan mampu:

1. Memahami apa yang dimaksud dengan stabilitas numerik dalam simulasi fisika.
2. Menjelaskan pengaruh langkah waktu dan grid terhadap hasil simulasi.
3. Mengenali solusi numerik yang tidak stabil dan membedakannya dari perilaku fisika yang sebenarnya.
4. Menguji dan membandingkan hasil simulasi dengan parameter numerik yang berbeda.
5. Menentukan parameter numerik yang layak agar simulasi memberikan hasil yang masuk akal secara fisika.

2. Teori

1. Apa itu Stabilitas Numerik?

Dalam fisika komputasi, solusi numerik diperoleh dengan mendiskritkan persamaan kontinu. Proses ini selalu menimbulkan kesalahan kecil.

Stabilitas numerik akan menjawab pertanyaan “*Apakah kesalahan kecil ini akan tetap kecil, atau justru tumbuh dan merusak solusi?*”

- Apabila stabil, maka kesalahan akan tetap terkendali
- Apabila tidak stabil, maka kesalahan akan tumbuh secara eksponensial
- Apabila tidak akurat tapi stabil, maka solusi halus tapi salah secara kuantitatif

2. Diskritisasi Waktu

Untuk persamaan diferensial:

$$\frac{dy}{dt} = f(y, t)$$

Pendekatan numerik paling sederhana:

$$y(n + 1) = y(n) + f[y(n), t(n)]\Delta t$$

di mana

Δt adalah langkah waktu

indeks n adalah waktu diskrit

Nilai Δt sangat menentukan bagaimana solusi berperilaku

- Apabila Δt terlalu besar, maka solusi bisa tidak stabil
- Apabila Δt terlalu kecil, maka solusi stabil tapi mahal komputasi (perhitungan akan lebih memakan banyak waktu).

3. Model Uji

3.1. Contoh-1: Peluruhan Eksponensial

Gunakan persamaan sederhana:

$$\frac{dy}{dt} = -\lambda y$$

Solusi analitiknya adalah

$$y(t) = y(0) \exp(-\lambda t)$$

Pendekatan numeriknya adalah

$$y(n + 1) = y(n) - \lambda y(n) \Delta t$$

Persamaan ini secara fisika selalu stabil, sehingga jika solusi numerik divergen maka disebabkan oleh kesalahan numerik murni.

Untuk menguji kualitas metode numerik, kita tidak hanya membandingkan dengan solusi analitik, tetapi juga melakukan tiga pengujian: konvergenitas, konservasi positività, dan stabilitas.

- Uji konvergensi menghitung error pada t_{maks} terhadap solusi referensi dengan langkah waktu sangat kecil:

$$error = |y_{numerik}(t_{maks}) - y_{referensi}(t_{maks})|$$

Dengan memperkecil Δt , error diharapkan mengecil secara proporsional terhadap Δt (orde satu).

- Uji konservasi positività memeriksa apakah solusi numerik menjadi negatif. Syarat agar adalah $y_{n+1} \geq 0$

$$1 - \lambda \Delta t \geq 0 \rightarrow \Delta t \leq \frac{1}{\lambda}$$

Jika dilanggar, meskipun stabil, nilai negatif muncul dan melanggar sifat fisis.

- Uji stabilitas didasarkan pada faktor amplifikasi $|1 - \lambda \Delta t|$. Agar solusi tidak berosilasi dan divergen, harus dipenuhi

$$|1 - \lambda \Delta t| \leq 1 \rightarrow \Delta t \leq \frac{2}{\lambda}$$

Bila Δt melebihi batas ini, solusi numerik akan meledak.

Kode ini berisi contoh implementasi python untuk kasus peluruhan menggunakan lima nilai Δt yang berbeda untuk diamati kestabilannya terhadap parameter yang memberikan galat terkecil.

Kode-2.1

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

# Parameter
l, y0, T = 1.0, 1.0, 5.0
dt_list = np.array([0.05, 0.1, 0.5, 1.0, 2.0])

# Solusi referensi (analitik Euler dengan dt kecil)
dt_ref = 0.001
N_ref = int(round(T / dt_ref))
t_ref = np.linspace(0, T, N_ref + 1)
y_ref = y0 * (1 - l * dt_ref) ** np.arange(N_ref + 1)
y_ref_end = y_ref[-1]
```

```
# Plot
errors = []
fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(12, 5))

for dt in dt_list:
    N = int(round(T / dt))
    t = np.linspace(0, T, N + 1)
    y = y0 * (1 - 1 * dt) ** np.arange(N + 1) # closed-form Euler iterate

    ax1.plot(t, y, 'o-', label=f'dt={dt}')
    errors.append(abs(y[-1] - y_ref_end))

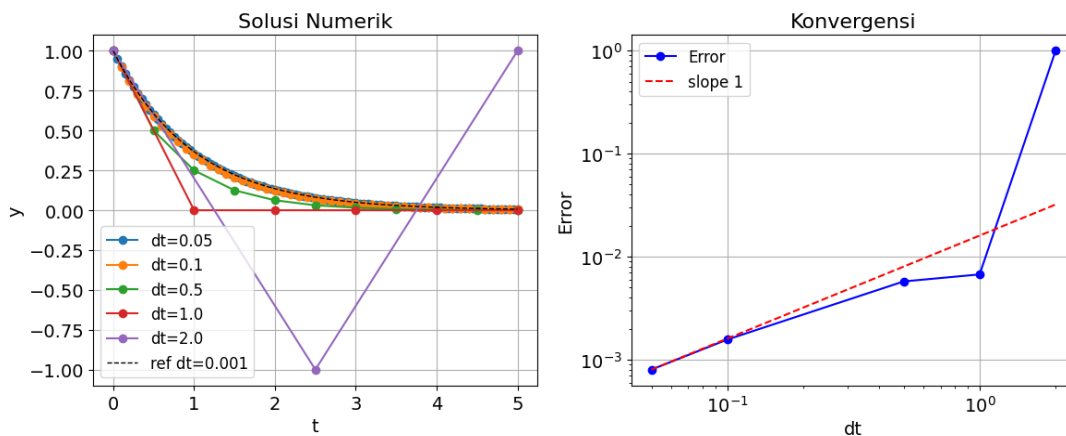
    if np.any(y < 0):
        print(f'Peringatan: dt={dt} menghasilkan nilai negatif')
    if np.max(np.abs(y)) > 1e3:
        print(f'Peringatan: dt={dt} divergen')

# Referensi & konfigurasi plot
ax1.plot(t_ref, y_ref, 'k--', label=f'ref dt={dt_ref}', lw=1)
ax1.set(xlabel='t', ylabel='y', title='Solusi Numerik')
ax1.legend(); ax1.grid(True)

errors = np.array(errors)
ax2.loglog(dt_list, errors, 'bo-', label='Error')
ax2.loglog(dt_list, errors[0] * (dt_list / dt_list[0]), 'r--', label='slope
1')
ax2.set(xlabel='dt', ylabel='Error', title='Konvergensi')
ax2.legend(); ax2.grid(True)

plt.tight_layout(); plt.show()

print('\nHasil uji konvergensi:')
for dt, err in zip(dt_list, errors):
    print(f'dt={dt:.3f}, error={err:.3e}')
```



Gambar 2.1. Uji stabilitas Langkah waktu

Hasil simulasi menunjukkan bahwa semakin kecil Δt , error semakin kecil (konvergen), namun Δt yang terlalu besar (misalnya $> 2/\lambda$) dapat menyebabkan solusi divergen. Di antara rentang stabil, ada pula Δt yang meskipun tidak divergen, tetap menghasilkan nilai negatif pada langkah awal, sehingga tidak memenuhi syarat konservasi. Dengan demikian, pemilihan Δt harus mempertimbangkan tidak hanya ketelitian, tetapi juga batasan stabilitas dan konservasi sifat fisis sistem. Di sisi lain, Δt yang sangat kecil akan meningkatkan jumlah langkah waktu dan waktu komputasi, sehingga diperlukan keseimbangan antara akurasi dan efisiensi.

3.2. Contoh 2: Stabilitas pada Sistem Osilator Harmonik

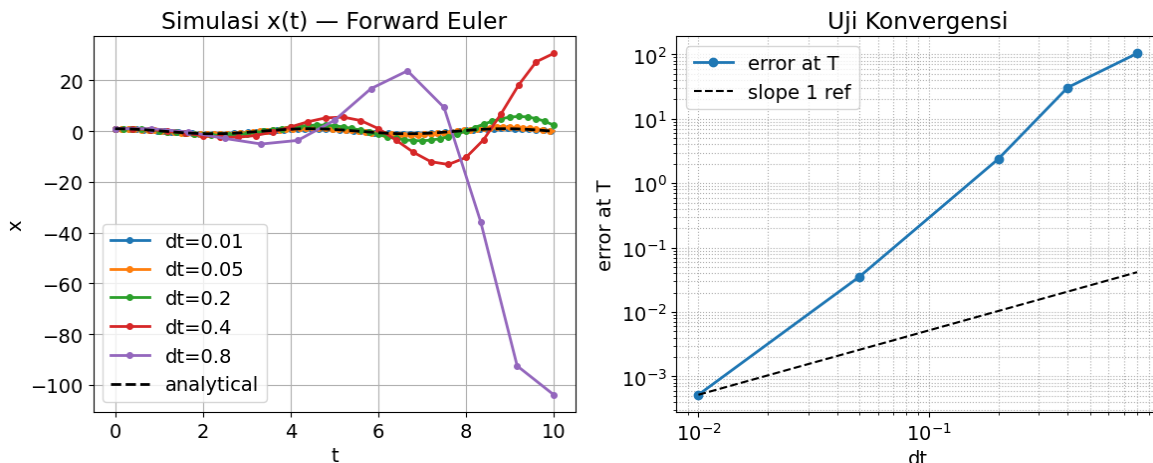
Persamaan model osilator harmonik adalah

$$\frac{d^2x}{dt^2} = -\omega^2 x$$

Ubah menjadi sistem orde-1

$$\frac{dx}{dt} = v$$

$$\frac{dv}{dt} = -\omega^2 x$$



Gambar 2.2. Kurva stabilitas numerik sistem osilator harmonik

Dengan mengubah persamaan tersebut menjadi sistem orde-1. Jika menggunakan $x(0) = 1.0$, $v(0) = 0$, $\omega = 2.0$, $\Delta t = 0.01, 0.05, 0.2, 0.4$ dan 0.8 akan didapat perbedaan kestabilan numerik seperti terlihat pada Gambar 2.2. Secara fisika seharusnya nilai amplitudo tetap terhadap waktu tetapi secara numerik amplitudo dapat tumbuh (tidak stabil) atau menyusut (*numerical damping*). Pada kasus osilasi terlihat pemilihan nilai Δt yang terlalu besar akan menyebabkan sistem semakin tidak stabil.

3. Tugas Praktikum

Petunjuk Umum

1. Gunakan bahasa pemrograman Python.
2. Setiap bagian harus disertai (1) grafik hasil simulasi dan (2) analisis singkat (narasi, bukan hanya angka).

3. Jawaban tidak dinilai dari benar–salah angka, tetapi dari (1) ketepatan interpretasi, (2) konsistensi fisika–numerik, (3) kualitas visualisasi

Tugas

1. Terapkan kode-2.1 untuk menguji stabilitas numerik pada kode anda dalam modul sebelumnya (**Modul-1**) untuk **Tugas Akhir Nomor 3** (persamaan peluruhan eksponensial terkopel). Tetapkan nilai $y(0) = 1$ dan $\lambda = 1$. Lakukan simulasi untuk minimal 5 nilai Δt berbeda (dua stabil, satu batas stabil dan dua tidak stabil). Bandingkan hasil numerik dengan solusi analitik.
 - a) Apa yang berbeda dari grafik errornya jika dibandingkan terhadap hasil dari kode-2.1?
 - b) Pada nilai Δt berapa solusi mulai menyimpang dan apakah penyimpangan tersebut bersifat fisis?
 - c) Apa hubungan Δt dengan skala waktu sistem?
 - d) Apakah solusi numerik “rusak” sejak awal atau secara bertahap?
2. Ubah persamaan pada contoh 2 sistem osilator harmonik menjadi sistem orde-1 dan pendekatan numeriknya. Terapkan $x(0) = 1.0$, $v(0) = 0.0$, $\omega = 2.0$. Simulasikan untuk minimal 5 nilai Δt (s) berbeda berbeda (dua stabil, satu batas stabil dan dua tidak stabil) dan amati:
 - a) Perubahan amplitudo dan munculnya redaman numerik
 - b) Bagaimana grafik konvergensinya
3. Jawab beberapa pertanyaan berikut secara naratif:
 - a) Apa perbedaan solusi tidak stabil dan solusi tidak akurat?
 - b) Mengapa simulasi yang terlihat halus belum tentu benar?
 - c) Jika Anda melakukan simulasi riset, uji apa yang wajib dilakukan terlebih dahulu?
 - d) Bagaimana Modul-2 ini mengubah cara Anda memandang hasil simulasi?

Dalam fisika komputasi, komputer tidak pernah “berbohong”, tetapi ia bisa “menyesatkan” jika kita tidak memahami batasannya.