Fundusze Europejskie Wiedza Edukacja Rozwój | Rzeczpospolita Polska | Unia Europejska Europejski Fundusz Społeczny

Politechnika Śląska jako Centrum Nowoczesnego Kształcenia opartego o badania i innowacje

POWR.03.05.00-IP.08-00-PZ1/17

Projekt współfinansowany przez Unię Europejską ze środków Europejskiego Funduszu Społecznego

---

Metody numeryczne w fizyce
Instytut Fizyki
Fizyka Techniczna, stopień 1

Oprogramowanie dla ćwiczeń laboratoryjnych

- **Python 3.7**
  oraz biblioteki:
  - **NumPy**
  - **Matplotlib**
  - **SciPy**

---

1. Reprezentacja zmiennoprzecinkowa i błędy

1.1 Przedyskutować zachowanie poniższej pętli.

```
x = 1E20
dx = 1E20

while x != x + dx:
    dx = dx/2.

print(x - (x + dx), dx)
```

a) Dlaczego w programie `x == x + dx`?
b) Jak zmiana `x` i `dx` wpłynie na działanie programu?

1.2 Sprawdzić wpływ zmiennych np. float32 i np.float64 na działanie programu.

```
import numpy as np

x = np.float64(1E20)
dx = np.float64(1E20)

while x != x + dx:
    dx = dx/2.
print(x - (x + dx), dx)
```

---

2. Stabilność metod i kumulacja błędów

2.1 Dany jest problem układania monet (zobacz Johnson, 1995).

Johnson, Paul B. (April 1955). "Leaning Tower of Lire". *American Journal of Physics*. 23 (4): 240–240. Bibcode:1955AmJPh..23..240J. doi:10.1119/1.1933957

```
d = 0.5     #cm
i, l_c, l_p = 1, 1., 0.

while l_c > l_p:    #sprawdzamy czy pętla działa poprawnie
    l_p = l_c
    l_c += d/i
    i += 1
    print('iteracja: ', i, ', wysunięcie: ', l_c, ', dodane wysunięcie: ', d/i)
```

a) Po ilu iteracjach dodawanie `l_c += d/i` przestanie działać poprawnie?

2.2 Dane jest prawo $dN/N=-\lambda dt$ opisujące rozpad izotopu węgla ^{14}C. Przy pomocy iteracji sprawdzić zależność $N(t)$ dla wybranych Δt, N_0. Dlaczego poniższy kod przestaje działać znacznie szybciej niż w przykładzie 2.1?

```
lbd = 1./8266.6426 #yr**-1 stała zaniku C-14
delta_t =  1000. #yr krok iteracji
N0 = 1E30  #początkowa liczba atomów C-14
N = N0

i, N_p = 0, N * 2

while N < N_p: #sprawdzamy czy N maleje wraz z czasem
    N_p = N
    N -= N * lbd * delta_t
    i += 1
    print('iteracja: ', i,', N(t): ', N)
```

2.3 Równanie $dN/N=-\lambda dt$ posiada rozwiązanie analityczne $N=N_0\exp(-\lambda t)$. Sprawdzić różnicę pomiędzy metodą iteracyjną a rozwiązaniem analitycznym. Narysować wykresy i porównać dla różnych kroków iteracji Δt.

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

lbd = 1./8266.6426 #yr**-1 stała zaniku C-14
delta_t =  1000. #yr krok iteracji
N_0 = 1.  # początkowa liczba radionuklidów

time = np.arange(0., 50000., delta_t)
N_iter = np.zeros(time.size)

N_a = N_0*np.exp(-lbd*time) #rozwiązanie analityczne

N = N_0 # ustawiamy początkową liczbę radionuklidów dla metody iteracyjnej

for i, t in enumerate(time):
    N_iter[i] = N
    N -= N * lbd * delta_t


fig, ax = plt.subplots()

ax.plot(time, N_iter, '.', label='Rozwiązanie iteracyjne')
ax.plot(time, N_a, '.', label='Rozwiązanie analityczne')
ax.plot(time, N_a-N_iter, '.', label='N_a-N_iter')

ax.set(xlabel='time (yr)', ylabel='N')
plt.legend()
plt.show()
```

Zadanie do samodzielnego wykonania.

Rozwiązać iteracyjne poniższy układ

$$dN_A = -\lambda_{AB} dt\, N_A - \lambda_{AC} dt\, N_A$$
$$dN_B = \lambda_{AB} dt\, N_A - \lambda_{BD} dt\, N_B$$
$$dN_C = \lambda_{AC} dt\, N_A - \lambda_{CD} dt\, N_C$$
$$dN_D = \lambda_{BD} dt\, N_B + \lambda_{CD} dt\, N_C$$

oraz narysować wykres. Przyjąć λ_{AB} = 1, λ_{AC} = 2, λ_{BD} = 3, λ_{CD} = 4 oraz dla czasu „0" (t_0=0):
$N_{A,0}$ = 1, $N_{D,0}$ = $N_{B,0}$ = $N_{D,0}$ = 0