

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Навчально-науковий інститут атомної та теплової енергетики
Кафедра інженерії програмного забезпечення в енергетиці

ДОМАШНЯ РОБОТА №3
з дисципліни «Математичні моделі процесів і систем»

тема «Однокрокові методи розв’язання задачі Коші»

Варіант № 13

Виконав:

Студент 3 курсу, групи TI-01

Круть Катерина
(прізвище ім’я)

Дата здачі 2023-03-14

Київ – 2023

Завдання.

З точністю до 0.0001 знайти розв'язок задачі Коші для звичайного диференційного рівняння першого порядку $y' = f(x, y)$ на відрізку $[a, b]$ з кроком $h=0.1$ за початкових умов $y(x_0) = y_0$:

- 1) чисельними методами (розробити програму на одній з мов програмування та в Excel):
 - 1) методом Ейлера (явним);
 - 2) методом Ейлера-Коші;
 - 3) вдосконаленим методом Ейлера;
 - 4) методом Рунге-Кутта 4-го порядку;
- 2) використовуючи точний розв'язок диференціального рівняння, отриманий одним з математичних пакетів або онлайн-калькулятора;
- 3) в одній системі координат побудувати для кожного методу графіки наближеного розв'язку та інтегральну криву знайденого точного розв'язку в Excel або в розробленій програмі;
- 4) порівняти отримані наближені значення розв'язку задачі з точним розв'язком чисельно та графічно (табл. 1 та рис.1).

Дані:

13	$y' = y + \cos\left(\frac{x}{13}\right)$	$y(1.4)=2.5$	$[1.4;2.4]$
----	--	--------------	-------------

Алгоритми реалізації:

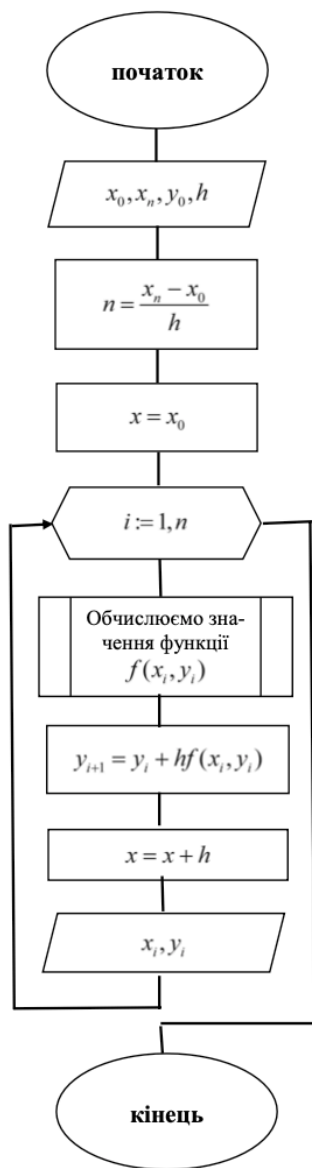


Рисунок 8. Схема алгоритму метода Ейлера.

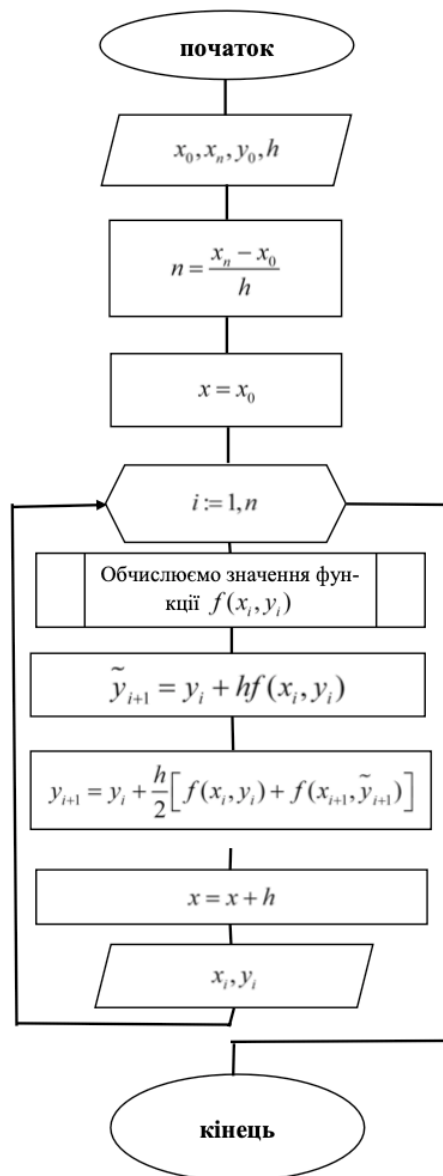


Рисунок 10. Схема алгоритму метода Ейлера-Коші.

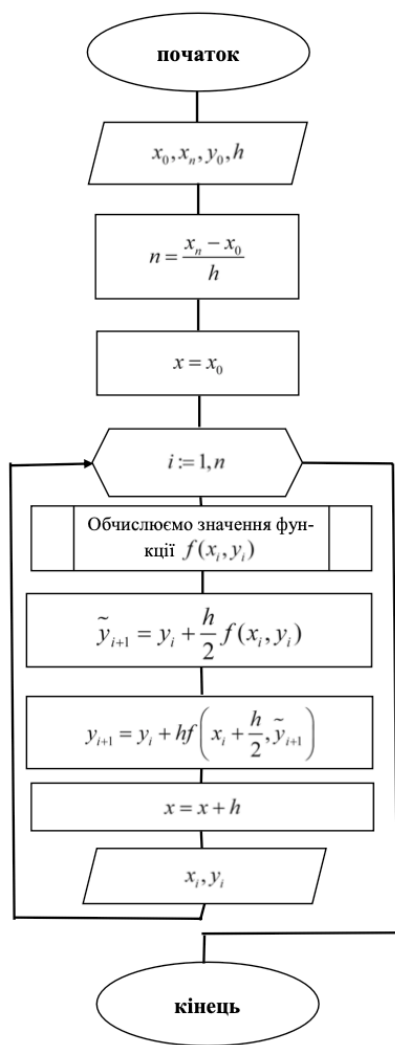


Рисунок 11. Схема алгоритму вдосконаленого метода Ейлера.

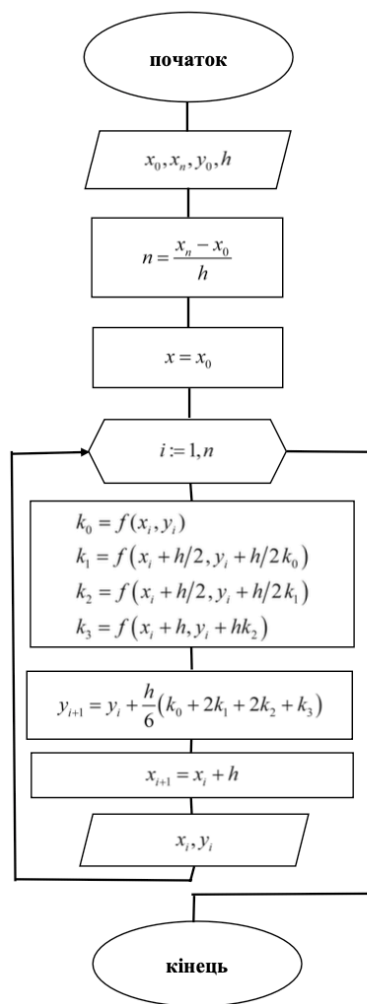


Рисунок 13. Схема алгоритму метода Рунге – Кутта 4-го порядку.

Формули:

Явний метод Ейлера:

$$y_{i+1} = y_i + f(x_i, y_i)h - \text{послідовність Ейлера.}$$

Метод Ейлера-Коші:

$$y_{i+1} = y_i + \frac{h}{2} [f(x_i, y_i) + f(x_{i+1}, y_{i+1})],$$
$$y_{i+1} = y_i + \frac{h}{2} [f(x_i, y_i) + f(x_{i+1}, y_i + hf(x_i, y_i))], i = 0, 1, 2, \dots, n-1.$$

Вдосконалений метод Ейлера:

$$y_{i+1} = y_i + f\left(x_i + \frac{h}{2}, y_{i+1}\right) \cdot h$$
$$y_{i+1} = y_i + f\left(x_i + \frac{h}{2}, y_i + \frac{h}{2} f(x_i, y_i)\right) \cdot h.$$

Метод Рунге-Кутта 4-го порядку:

$$y_{i+1} = y_i + \frac{h}{6} (k_0 + 2k_1 + 2k_2 + k_3), i = 0, 1, \dots, n-1,$$
$$\begin{cases} k_0 = f(x_i, y_i), \\ k_1 = f\left(x_i + \frac{h}{2}, y_i + \frac{hk_0}{2}\right), \\ k_2 = f\left(x_i + \frac{h}{2}, y_i + \frac{hk_1}{2}\right), \\ k_3 = f(x_i + h, y_i + hk_2). \end{cases}$$

Програмна реалізація:

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

"""
З точністю до 0.0001 знайти розв'язок задачі Коші для звичайного
диференційного рівняння першого порядку  $y' = y + \cos(x / 13)$  на відрізку  $[1.4; 2.4]$  з
кроком  $h=0.1$  за початкових умов  $y(1.4) = 2.2$ 
"""

def f(x, y):
    """  $f(x, y) = y + \cos(x/13)$  """
    return y + math.cos(x / 13)

def df(x):
    """
 $y' = y + \cos(x/13)$ 
 $y' = Ce^x + 13\sin(x/13) / 170 - 169\cos(x/13) / 170$ 
 $x_0 = 1.4; y_0 = 2.5 \Rightarrow$ 
 $C = - (13 * \sin(7/65) - 169 * \cos(7/65) - 425) / 170 * e^{1.4}$ 
 $y' = (1/170) * (e*\sin(x/e) - e^2 * \cos(x/ e) - e^{(x - 1.4)}*(e*\sin(7/(5e)) -$ 
 $e^2*\cos(7/(5e)) - 5))$ 
"""
    return -((13 * math.sin(7 / 65) - 169 * math.cos(7 / 65) - 425) * math.e **
(x - 1.4)) / 170 + 13 * math.sin(
    x / 13) / 170 - 169 * math.cos(x / 13) / 170

def precise_solution(x0, xn, y0, h):
    n = (xn - x0) / h
    x = x0
    y = y0
    result = {'x': [], 'y': []}
    for _ in range(int(n)):
        result['x'].append(round(x, 4))
        result['y'].append(round(y, 4))
        x = x + h
        y = df(x)
    result['x'].append(round(x, 4))
    result['y'].append(round(y, 4))
    return result
```

```

def euler(x0, xn, y0, h):
    n = (xn - x0) / h

    x = x0
    y = y0

    result = {'x': [], 'y': []}
    for _ in range(int(n)):
        result['x'].append(round(x, 4))
        result['y'].append(round(y, 4))
        x = x + h
        y = y + h * f(x, y)
    result['x'].append(round(x, 4))
    result['y'].append(round(y, 4))
    return result

def euler_cauchy(x0, xn, y0, h):
    n = (xn - x0) / h

    xi = x0 # x0 = 1.4
    yi = y0 # y0 = 2.2

    result = {'x': [], 'y': []}

    for _ in range(int(n)):
        result['x'].append(round(xi, 4)) # записуємо результуюче значення xi
        result['y'].append(round(yi, 4)) # записуємо результуюче значення yi
        yi1_ = yi + h * f(xi, yi) # yi+1 = yi + hf(xi, yi)
        yi1 = yi + 0.5 * h * (f(xi, yi) + f(xi + h, yi1_)) # yi+1 = yi + h / 2
        [ f(xi, yi) + f(xi+1, yi1) ]
        xi = xi + h # перевизначаємо x | x = x + h
        yi = yi1 # перевизначаємо yi

```

13	$y' = y + \cos\left(\frac{x}{13}\right)$	$y(1.4)=2.5$	[1.4;2.4]
----	--	--------------	-----------

```

    result['x'].append(round(xi, 4)) # записуємо результуюче значення xn
    result['y'].append(round(yi, 4)) # записуємо результуюче значення yn
    return result

```

```

def euler_enhanced(x0, xn, y0, h):
    n = (xn - x0) / h

```

```

xi = x0 # x0 = 1.4
yi = y0 # y0 = 2.2

result = {'x': [], 'y': []}

for _ in range(int(n)):
    result['x'].append(round(xi, 4)) # записуємо результуюче значення xi
    result['y'].append(round(yi, 4)) # записуємо результуюче значення yi
    yi1 = yi + h * f(xi + h / 2, yi + h / 2 * f(xi, yi)) # yi+1 = yi +
hf(xi + h / 2, yi + h / 2 * f(xi, yi))
    xi = xi + h # перевизначаємо x | x = x + h
    yi = yi1 # перевизначаємо yi
    result['x'].append(round(xi, 4)) # записуємо результуюче значення xn
    result['y'].append(round(yi, 4)) # записуємо результуюче значення yn
return result

def runge_kutte(x0, xn, y0, h):
    n = (xn - x0) / h

    xi = x0 # x0 = 1.4
    yi = y0 # y0 = 2.2

    result = {'x': [], 'y': []}

    for _ in range(int(n)):
        result['x'].append(round(xi, 4)) # записуємо результуюче значення xi
        result['y'].append(round(yi, 4)) # записуємо результуюче значення yi
        k0 = f(xi, yi)
        k1 = f(xi + 0.5 * h, yi + 0.5 * h * k0)
        k2 = f(xi + 0.5 * h, yi + 0.5 * h * k1)
        k3 = f(xi + h, yi + h * k2)
        yi1 = yi + h / 6 * (k0 + 2 * k1 + 2 * k2 + k3)
        xi = xi + h # перевизначаємо x | x = x + h
        yi = yi1 # перевизначаємо yi
        result['x'].append(round(xi, 4)) # записуємо результуюче значення xn
        result['y'].append(round(yi, 4)) # записуємо результуюче значення yn
    return result

def draw(data):
    """Function for drawing graphic"""
    x = np.linspace(1.4, 2.4, len(data))
    y = data
    plt.plot(x, y, 'green', label='y=sin(x)')
    plt.xlabel('x')

```



```

plt.ylabel('y')
plt.grid(color='black', linestyle='--', linewidth=0.5)
plt.legend(loc='best')
plt.show()

def draw_all(euler_, euler_cauchy_, euler_enh, runge, precise):
    """Function for drawing graphic"""
    x = np.linspace(1.4, 2.4, len(euler_))
    plt.plot(x, euler_, 'green', linewidth=0.2, label='Явний метод Ейлера')
    plt.plot(x, euler_cauchy_, 'black', linewidth=0.2, label='метод Ейлера-Коші')
    plt.plot(x, euler_enh, 'yellow', linewidth=0.2, label='Вдосконалений метод Ейлера')
    plt.plot(x, runge, 'red', linewidth=0.2, label='Рунге-Кутта')
    plt.plot(x, precise, 'orange', linewidth=0.2, label='Точне')
    plt.xlabel('x')
    plt.ylabel('y')
    plt.grid(color='black', linestyle='--', linewidth=0.5)
    plt.legend(loc='best')
    plt.show()

def print_euler(data):
    for i in range(len(data['y'])):
        print(f'x{i}={data["x"][i]}; y{i}={data["y"][i]}')

def print_data_table(euler_, euler_cauchy_, euler_enhanced_, runge_kutte_, precise_):
    data = [euler_['x'], euler_['y'], euler_cauchy_['y'], euler_enhanced_['y'], runge_kutte_['y'], precise_['y']]
    rows = [f'n = {x}' for x in range(len(data[0]))]
    data = np.transpose(data)
    columns = ('x', 'y Ейлер Явний', 'y Ейлера-Коші', 'y Ейлер Вд.', 'y Рунге-Кутта', 'y Точне')
    colors = plt.cm.BuPu(np.linspace(0, 0.5, len(rows)))
    cell_text = [[f'{x}' for x in data[row]] for row in range(len(data))]

    plt.box(on=None)

    plt.table(cellText=cell_text, rowLabels=rows, rowColours=colors, colLabels=columns, loc='center')

    plt.xticks([])
    plt.yticks([])
    ax = plt.gca()

```

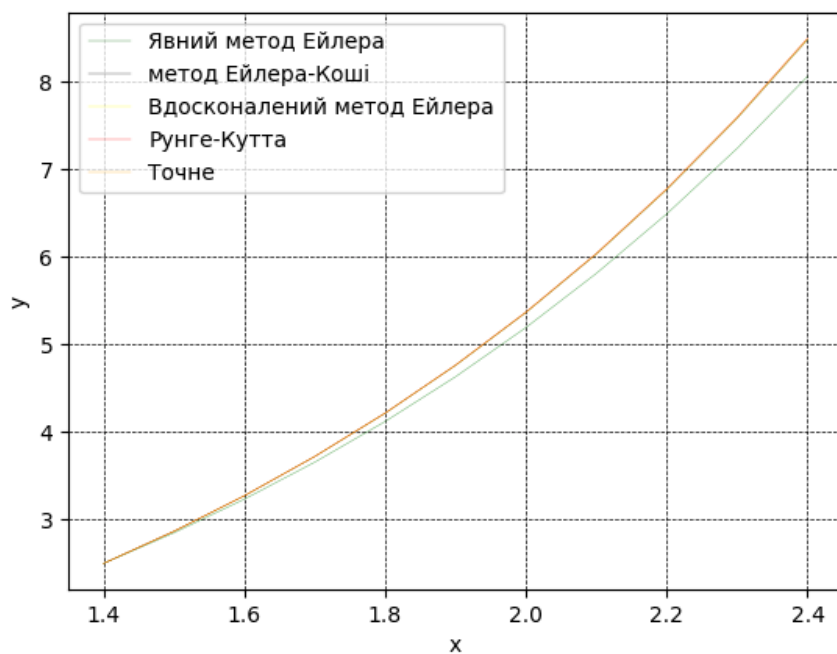
```
ax.get_xaxis().set_visible(False)
ax.get_yaxis().set_visible(False)
plt.show()
```

```
x0_ = 1.4
xn_ = 2.4
y0_ = 2.5
h_ = 0.1
```

```
euler_data = euler(x0_, xn_, y0_, h_)
euler_cauchy_data = euler_cauchy(x0_, xn_, y0_, h_)
euler_enhanced_data = euler_enhanced(x0_, xn_, y0_, h_)
runge_kutte_data = runge_kutte(x0_, xn_, y0_, h_)
precise_data = precise_solution(x0_, xn_, y0_, h_)
```

```
draw_all(euler_data['y'], euler_cauchy_data['y'], euler_enhanced_data['y'],
runge_kutte_data['y'], precise_data['y'])
print_data_table(euler_data, euler_cauchy_data, euler_enhanced_data,
runge_kutte_data, precise_data)
```

Результат роботи програми:



	х	у Ейлер Явний	у Ейлера-Коші	у Ейлер Вд.	у Рунге-Кутта	у Точне
n = 0	1.4	2.5	2.5	2.5	2.5	2.5
n = 1	1.5	2.8493	2.8668	2.8668	2.8674	2.8674
n = 2	1.6	3.2335	3.2721	3.2721	3.2734	3.2734
n = 3	1.7	3.656	3.7199	3.7199	3.722	3.722
n = 4	1.8	4.1207	4.2145	4.2145	4.2177	4.2177
n = 5	1.9	4.6317	4.761	4.761	4.7654	4.7654
n = 6	2.0	5.1936	5.3647	5.3647	5.3706	5.3706
n = 7	2.1	5.8117	6.0317	6.0317	6.0393	6.0393
n = 8	2.2	6.4914	6.7686	6.7686	6.7782	6.7782
n = 9	2.3	7.239	7.5827	7.5827	7.5946	7.5946
n = 10	2.4	8.0612	8.4822	8.4822	8.4968	8.4968

Аналітичний розв'язок задачі Коші:

1. Умова

$$y' = y + \cos\left(\frac{x}{13}\right), \quad x_0 = 1.4, \quad y(x_0) = 2.5$$

2. Загальний розв'язок

$$y = C e^x + \frac{13 \sin\left(\frac{x}{13}\right)}{170} - \frac{169 \cos\left(\frac{x}{13}\right)}{170}$$

3. Знаходимо константу

$$y = C e^x + \frac{13 \sin\left(\frac{x}{13}\right)}{170} - \frac{169 \cos\left(\frac{x}{13}\right)}{170}$$

$$\begin{array}{l} \text{при } x = \frac{7}{5} \\ y = \frac{5}{2} \end{array}$$

↓

$$\frac{5}{2} = \frac{13 \sin\left(\frac{7}{65}\right)}{170} - \frac{169 \cos\left(\frac{7}{65}\right)}{170} + e^{\frac{7}{5}} C \rightarrow C = -\frac{13 \sin\left(\frac{7}{65}\right) - 169 \cos\left(\frac{7}{65}\right) - 425}{170 \cdot e^{\frac{7}{5}}}$$

2. Точний розв'язок

$$y = -\frac{(13 \sin\left(\frac{7}{65}\right) - 169 \cos\left(\frac{7}{65}\right) - 425) e^{x - \frac{7}{5}}}{170} + \frac{13 \sin\left(\frac{x}{13}\right)}{170} - \frac{169 \cos\left(\frac{x}{13}\right)}{170}$$

Висновки:

Під час виконання домашньої роботи біло опрацьовано лекцію №5 «Однокрокові методи розв'язування задачі Коші для звичайних диференціальних рівнянь», набуто навичок для роботи з однокроковими методами розв'язання задач Коші для звичайних диференціальних рівнянь та розроблено програмний продукт для реалізації алгоритмів вирішення таких задач. Крім того, проведено порівняння отриманих даних у графічному й табличному вигляді.

хі	Ейлера Неяв.	у Точне	Рунге-Кутта	Ейлера-Коші	Ейлера Вдосконалений
1,4	2,5	2,5	2,5	2,5	2,5
1,5	2,8493	2,8674	2,8674	2,8668	2,8668
1,6	3,2335	3,2734	3,2734	3,2721	3,2721
1,7	3,6560	3,722	3,722	3,7199	3,7199
1,8	4,1207	4,2177	4,2177	4,2145	4,2145
1,9	4,6317	4,7654	4,7654	4,761	4,761
2	5,1936	5,3706	5,3706	5,3647	5,3647
2,1	5,8117	6,0393	6,0393	6,0317	6,0317
2,2	6,4914	6,7782	6,7782	6,7686	6,7686
2,3	7,2390	7,5946	7,5946	7,5827	7,5827
2,4	8,0612		8,4968	8,4822	8,4822

