

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ  
ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені Ігоря СІКОРСЬКОГО»  
Фізико-технічний інститут

КОМП'ЮТЕРНИЙ ПРАКТИКУМ № 6.  
РОЗВ'ЯЗАННЯ ЗАДАЧІ КОШІ МЕТОДАМИ РУНГЕ-КУТТА ТА АДАМСА

Виконав  
студент 3 курсу ФТІ  
групи ФІ-21  
Климентьев Максим Андрійович

Перевірів:

Оцінка:

## **Зміст**

<b>1</b>	<b>Завдання</b>	<b>3</b>
<b>2</b>	<b>Вимоги до звіту</b>	<b>3</b>
<b>3</b>	<b>ПОСТАНОВКА ЗАДАЧІ</b>	<b>3</b>
<b>4</b>	<b>Вихідна система</b>	<b>3</b>
<b>5</b>	<b>Таблиця результатів</b>	<b>4</b>
<b>6</b>	<b>Графіки</b>	<b>5</b>
<b>7</b>	<b>Лістинг програми</b>	<b>6</b>

## 1 Завдання

Методами Рунге-Кутта та Адамса-Башфорта четвертого порядку розв'язати задачу Коші. На початку інтервалу у необхідній кількості точок значення для методу Адамса визначити методом Рунге-Кутта четвертого порядку.

## 2 Вимоги до звіту

Для деякого фіксованого  $h$  потрібно навести:

- значення точної функції розв'язку  $y(x)$ ;
- значення наближеного розв'язку  $y(x)$  у тих самих точках, одержані обома методами;
- значення функції помилки  $e(x)$  для обох методів (порівняти із «теоретичною» точністю);

У звіті наводять:

- графіки точного розв'язку та обох наближених - на одному рисунку;
- графіки обох помилок - на другому рисунку;
- лістинг програми.

## 3 ПОСТАНОВКА ЗАДАЧІ

Рівняння має вигляд:  $y' = (1-y) \cdot x^2 + F(x)$  Покласти  $h = 0.1$ . Початкові умови  $x(0)$  визначити, використовуючи точне значення розв'язку.

Нехай розв'язок відомий та визначається згідно з варіантами:

Необхідно підставити розв'язок у рівняння та визначити  $F(x)$  у правій частині. Таким чином, відомим є вигляд рівняння та його точний розв'язок, за допомогою числових методів далі будемо наближений розв'язок.

## 4 Вихідна система

Варіант	Точний розв'язок
10	$y = x \cdot \sin(x)$

## 5 Таблиця результатів

$x$	0.0	0.1	0.2	0.3	0.4
$y_{true}$	0.0	0.00998	0.03973	0.08866	0.15577
$y_{runge}$	0.0	0.00998	0.03973	0.08866	0.15577
$y_{adam}$	0.0	0.00998	0.03973	0.08866	0.15576

Табл. 1: Порівняння рішень на відрізку  $[0.0, 0.4]$

$x$	0.5	0.6	0.7	0.8	0.9
$y_{true}$	0.23971	0.33879	0.45095	0.57388	0.70499
$y_{runge}$	0.23971	0.33879	0.45095	0.57388	0.70499
$y_{adam}$	0.2397	0.33877	0.45093	0.57385	0.70495

Табл. 2: Порівняння рішень на відрізку  $[0.5, 0.9]$

$x$	1.0	1.1	1.2	1.3	1.4
$y_{true}$	0.84147	0.98033	1.11845	1.25263	1.37963
$y_{runge}$	0.84147	0.98033	1.11845	1.25262	1.37963
$y_{adam}$	0.84142	0.98026	1.11837	1.25255	1.37955

Табл. 3: Порівняння рішень на відрізку  $[1.0, 1.4]$

$x$	1.5	1.6	1.7	1.8	1.9
$y_{true}$	1.49624	1.59932	1.68583	1.75293	1.79797
$y_{runge}$	1.49624	1.59931	1.68582	1.75291	1.79794
$y_{adam}$	1.49616	1.59924	1.68575	1.75285	1.79791

Табл. 4: Порівняння рішень на відрізку  $[1.5, 1.9]$

$x$	2.0	2.1	2.2	2.3	2.4
$y_{true}$	1.81859	1.81274	1.77869	1.71512	1.62111
$y_{runge}$	1.81856	1.81269	1.77862	1.71503	1.62099
$y_{adam}$	1.81854	1.81269	1.77865	1.71509	1.62108

Табл. 5: Порівняння рішень на відрізку  $[2.0, 2.4]$

$x$	2.5	2.6	2.7	2.8	2.9
$y_{true}$	1.49618	1.3403	1.15393	0.93797	0.69382
$y_{runge}$	1.49603	1.34013	1.15371	0.93772	0.69354
$y_{adam}$	1.49616	1.34029	1.15392	0.93796	0.69382

Табл. 6: Порівняння рішень на відрізку  $[2.5, 2.9]$

$x$	3.0	3.1	3.2	3.3	3.4
$y_{true}$	0.42336	0.1289	-0.1868	-0.52056	-0.86884
$y_{runge}$	0.42305	0.12857	-0.18714	-0.5209	-0.86916
$y_{adam}$	0.42336	0.1289	-0.18679	-0.52055	-0.86883

Табл. 7: Порівняння рішень на відрізку  $[3.0, 3.4]$

$x$	3.5	3.6	3.7	3.8	3.9
$y_{true}$	-1.22774	-1.59307	-1.96039	-2.32506	-2.68229
$y_{runge}$	-1.22801	-1.59326	-1.96045	-2.32493	-2.68192
$y_{adam}$	-1.22773	-1.59306	-1.96038	-2.32505	-2.68227

Табл. 8: Порівняння рішень на відрізку  $[3.5, 3.9]$

$x$	4.0	4.1	4.2	4.3	4.4
$y_{true}$	-3.02721	-3.35494	-3.66062	-3.93951	-4.18705
$y_{runge}$	-3.02652	-3.35383	-3.65898	-3.93721	-4.18392
$y_{adam}$	-3.0272	-3.3549	-3.66069	-3.93915	-4.18857

Табл. 9: Порівняння рішень на відрізку  $[4.0, 4.4]$

## 6 Графіки

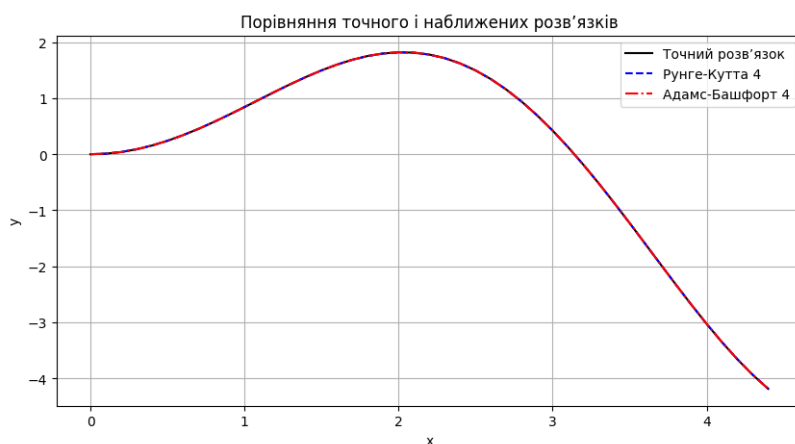


Рис. 1: Графік точного та наближених розв'язків

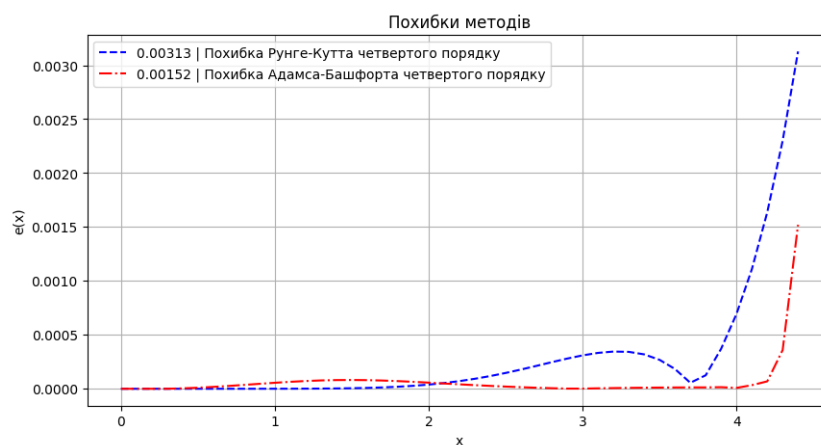


Рис. 2: Графік похибок методів Рунге-Кутта четвертого порядку і Адамса-Башфорта четвертого порядку

## 7 Лістинг програми

```

1 def f(x):
2     return x * np.sin(x)
3 def f_s(x, y):
4     # y'(x) = sin(x) + x cos(x)
5     # sin(x) + x cos(x) = (1 - x * sin(x)) * x**2 + F(x)
6     # F(x) = sin(x) + x * cos(x) - (1 - x * sin(x)) * x**2
7     Fx = np.sin(x) + x * np.cos(x) - (1 - x * np.sin(x)) * x**2
8     return (1 - y)*x**2 + Fx
9 def runge_kutta_4(x0, y0, h, n, f_s):
10    x = np.zeros(n)

```

```

11     y = np.zeros(n)
12     x[0] = x0
13     y[0] = y0
14     for i in range(n-1):
15         x_i = x[i]
16         y_i = y[i]
17
18         k1 = h * f_s(x_i, y_i)
19         k2 = h * f_s(x_i + h/2, y_i + 1/2*k1)
20         k3 = h * f_s(x_i + h/2, y_i + 1/2*k2)
21         k4 = h * f_s(x_i + h, y_i + k3)
22
23         x[i+1] = x_i + h
24         y[i+1] = y_i + (k1 + 2*k2 + 2*k3 + k4)/6
25     return x, y
26 def adams_bashforth_4(x_init, y_init, h, n, f_s):
27     x = x_init.copy()
28     y = np.zeros_like(y_init)
29     y[:4] = y_init[:4]
30     for i in range(3, n-1):
31         f_n = f_s(x[i], y[i])
32         f_n1 = f_s(x[i-1], y[i-1])
33         f_n2 = f_s(x[i-2], y[i-2])
34         f_n3 = f_s(x[i-3], y[i-3])
35
36         y[i+1] = y[i] + h*(55*f_n - 59*f_n1 + 37*f_n2 - 9*f_n3)/24
37     return x, y
38

```

Лістинг 1: Code