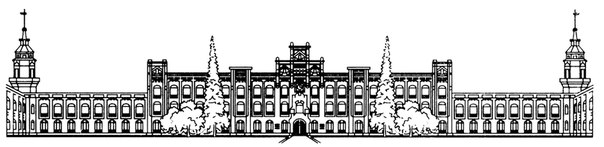
Національний технічний університет України «КПІ ім. Ігоря Сікорського»



Факультет інформатики та обчислювальної техніки

Кафедра інформаційних систем та технологій

Лабораторна робота №7

з дисципліни «Спеціальні розділи математики-2.  
Чисельні методи»

на тему

***"***Чисельне диференціювання та інтегрування***"***

Виконав:

студент групи ІС-31

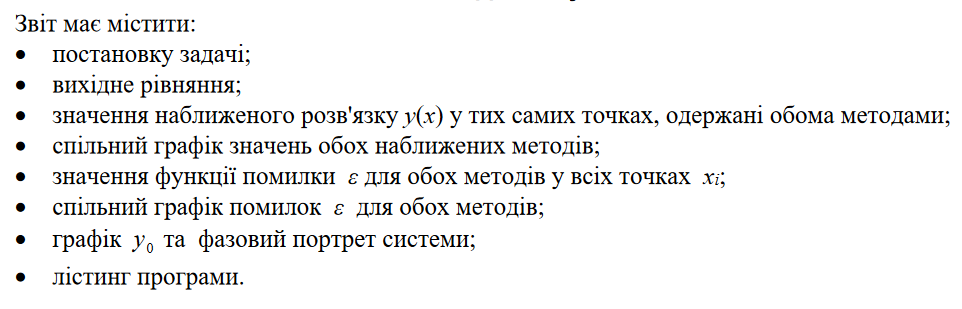
Коваль Богдан

Викладач:

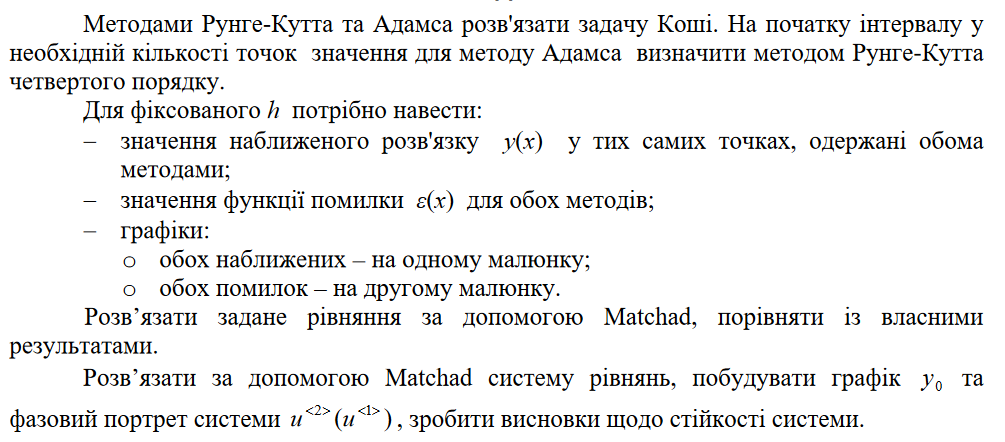
доц. Рибачук Л.В.

Київ – 2024

Зміст

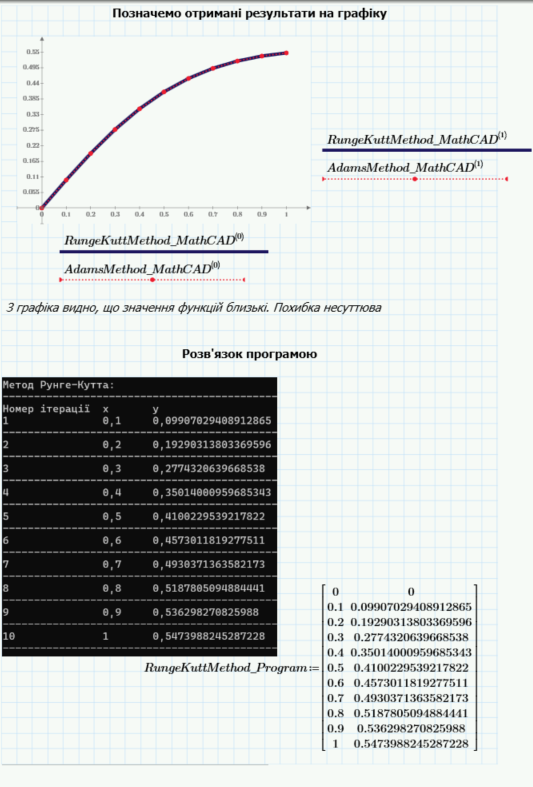


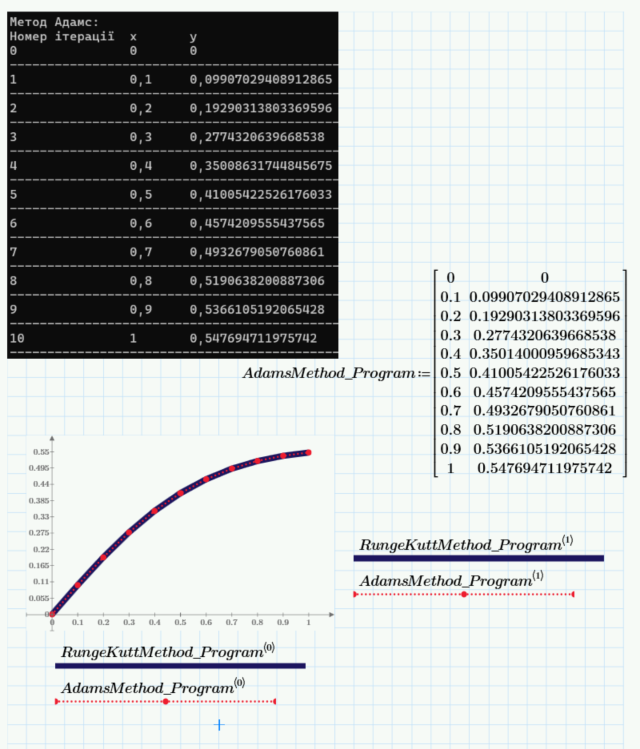
Завданя

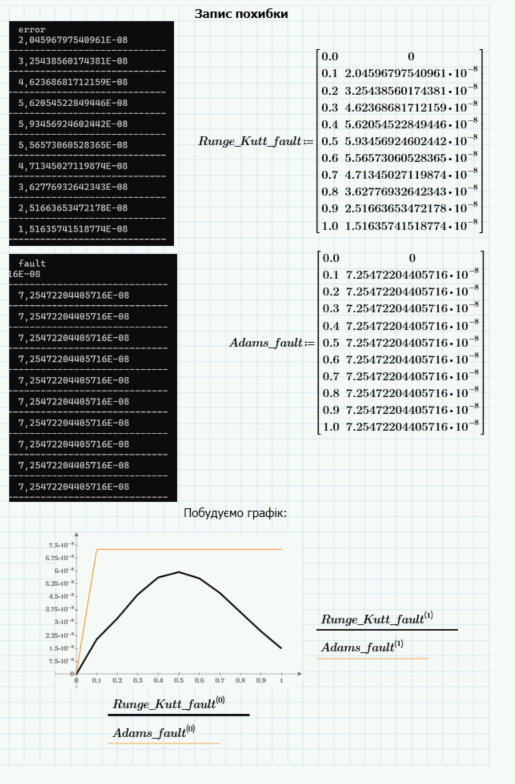


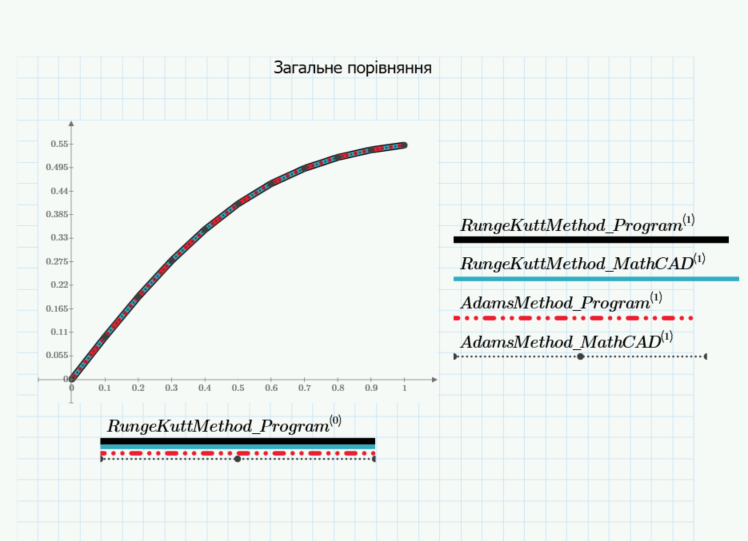
Постановка задачі. Вихідне рівняння

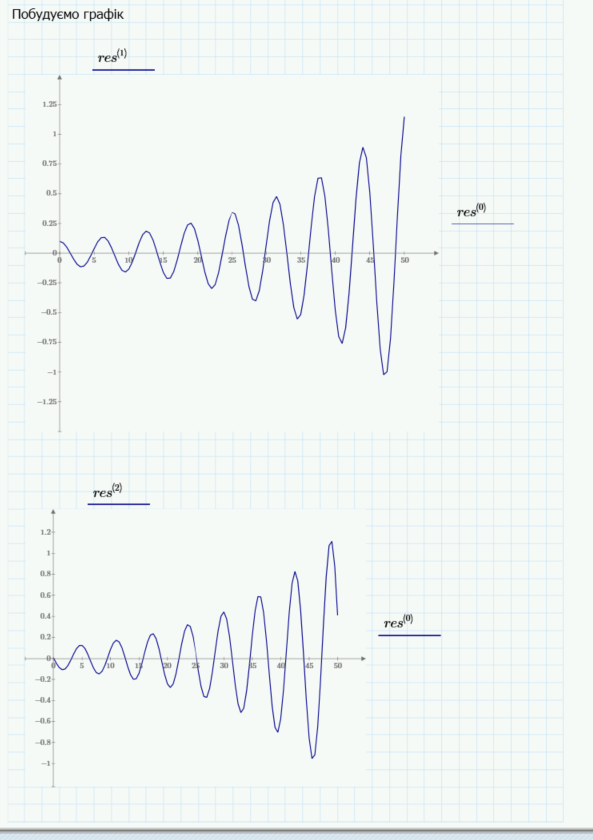


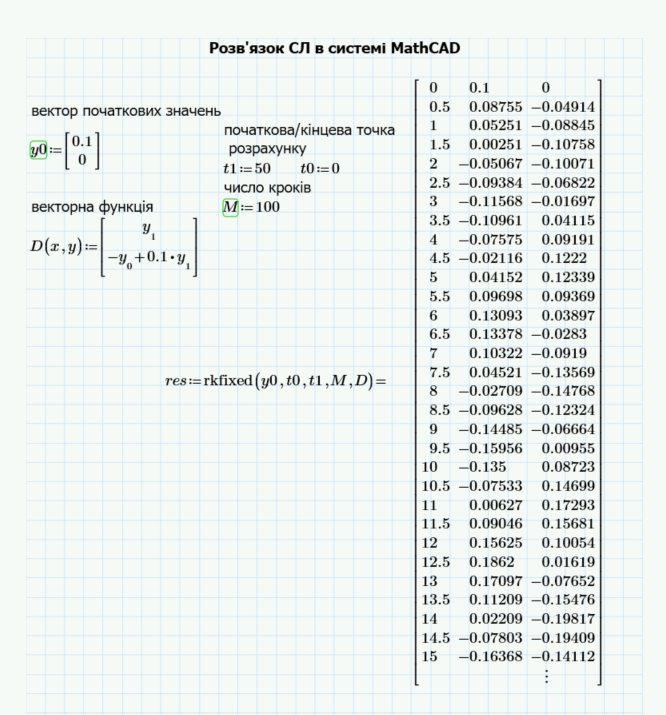


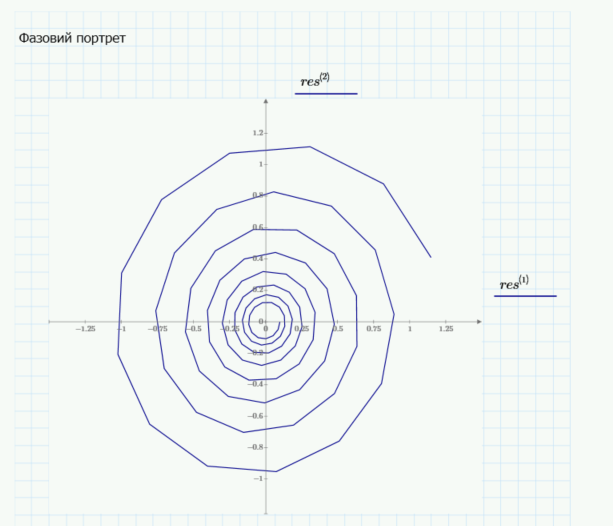












Висновок: Метод Рунге-Кутта четвертого порядку демонструє високу точність при розв'язанні задачі Коші і може бути ефективно використаний для визначення початкових значень для методу Адамса.

* Метод Адамса також показав гарні результати, але з дещо більшою похибкою в порівнянні з методом Рунге-Кутта.
* Використання Mathcad підтвердило правильність та точність обчислень, виконаних чисельними методами.
* Аналіз стійкості системи за допомогою фазового портрету в Mathcad показав, що система є стійкою, що відповідає теоретичним очікуванням.

Результати роботи демонструють ефективність чисельних методів для розв'язання диференціальних рівнянь та підтверджують їхню точність і надійність.

Лістинг програми

using System;

using System.Collections.Generic;

using System.Linq;

namespace Lab\_8

{

internal class Program

{

/// <summary>

/// Функція обчислення заданої функції

/// </summary>

static double calc\_fun(double x, double y)

{

return Math.Cos(1.4 \* x + y) + (x - y);

}

/// <summary>

/// Вивід розділювальної лінії

/// </summary>

static void PrintSep()

{

Console.WriteLine(string.Concat(Enumerable.Repeat("-", 75)));

}

/// <summary>

/// Метод Рунге-Кутта за формулою (2)

/// </summary>

/// <returns></returns>

static (double, double) RungeKuttaStep(double x, double y, double h)

{

double k1 = h \* calc\_fun(x, y);

double k2 = h \* calc\_fun(x + h / 2, y + k1 / 2);

double k3 = h \* calc\_fun(x + h / 2, y + k2 / 2);

double k4 = h \* calc\_fun(x + h, y + k3);

double del\_y = (k1 + 2 \* k2 + 2 \* k3 + k4) / 6.0;

return (y + del\_y, del\_y);

}

static void RungeKuttMethod(double a, double b, double x0, double y0)

{

double h = 0.1; // Початковий крок, взятий з умови

int M = 10; // Початкова кількість ітерацій

Console.WriteLine("Номер iтерацiї\tx\ty\t\t\tпомилка");

for (int i = 0; i < M; i++)

{

// Один крок з кроком h

(double y\_h, double del\_y\_h) = RungeKuttaStep(x0, y0, h);

// Два кроки з кроком h/2

(double y\_half1, double \_) = RungeKuttaStep(x0, y0, h / 2); // Додатково для обрахунку похибки за правилом Рунге

(double y\_h2, double del\_y\_h2) = RungeKuttaStep(x0 + h / 2, y\_half1, h / 2);

// Оцінка похибки за правилом рунте. Формула (8)

double error = Math.Abs((y\_h2 - y\_h) / 15.0);

y0 = y\_h;

x0 += h;

Console.WriteLine($"{i + 1}\t\t{Math.Round(x0, 5)}\t{y0}\t{error}");

PrintSep();

}

}

static void AdamsMethod(double x0, double y0)

{

int k = 3; // Початок ітерацій

double h = 0.1; // Початковий крок

// Початковий список, перші взяті перші 3 знач. із минулого метода

List<double> y = new List<double> { 0, 0.09907029408912865, 0.19290313803369596, 0.2774320639668538 };

List<double> x = new List<double> { 0, 0.1, 0.2, 0.3 };

double y\_predicted, y\_adjusted, y\_predicted\_half, y\_adjusted\_half, error;

Console.WriteLine("Номер iтерацiї\tx\ty\t\t\tпомилка");

do

{

// Прогнозоване значення з кроком h. За формулою (4)

y\_predicted = y[k] + (h / 24) \* (55 \* calc\_fun(x[k], y[k]) - 59 \* calc\_fun(x[k - 1], y[k - 1]) + 37 \* calc\_fun(x[k - 2], y[k - 2]) - 9 \* calc\_fun(x[k - 3], y[k - 3]));

// Скориговане значення з кроком h. За формулою (5)

y\_adjusted = y[k] + (h / 24) \* (9 \* y\_predicted + 19 \* calc\_fun(x[k], y[k]) - 5 \* calc\_fun(x[k - 1], y[k - 1]) + calc\_fun(x[k - 2], y[k - 2]));

// Прогнозоване значення з кроком h/2

double h\_half = h / 2;

y\_predicted\_half = y[k] + (h\_half / 24) \* (55 \* calc\_fun(x[k], y[k]) - 59 \* calc\_fun(x[k - 1], y[k - 1]) + 37 \* calc\_fun(x[k - 2], y[k - 2]) - 9 \* calc\_fun(x[k - 3], y[k - 3]));

// Скориговане значення з кроком h/2

y\_adjusted\_half = y[k] + (h\_half / 24) \* (9 \* y\_predicted\_half + 19 \* calc\_fun(x[k], y[k]) - 5 \* calc\_fun(x[k - 1], y[k - 1]) + calc\_fun(x[k - 2], y[k - 2]));

// Оцінка похибки методом Рунте за правилом Рунге (8)

error = Math.Abs((y\_adjusted\_half - y\_adjusted) / 15.0);

if (Math.Abs(y\_adjusted - y\_predicted) <= Math.Pow(10, -4))

{

h /= 2; // Перевірка на похибку

y\_predicted = y\_adjusted;

}

x.Add(x[k] + h); // Запис у масив

y.Add(y\_adjusted);

k++;

} while (Math.Round(x[k], 5) <= 1);

for (int i = 0; i < k; i++)

{

Console.WriteLine($"{i}\t\t{Math.Round(x[i], 5)}\t{y[i]}\t{error \* Math.Pow(10, -4)}");

PrintSep();

}

}

static void Main(string[] args)

{

PrintSep();

Console.WriteLine("\tЛабораторна робота #8");

Console.WriteLine("Виконав студент групи IC-31 Коваль Богдан");

PrintSep();

double a = 0;

double b = 1;

double x0 = 0;

double y0 = 0;

Console.WriteLine("Метод Рунге-Кутта:");

PrintSep();

RungeKuttMethod(a, b, x0, y0);

Console.WriteLine("\n");

Console.WriteLine("Метод Адамса:");

AdamsMethod(x0, y0);

}

}

}