**Лабораторна робота**

**Розв’язування звичайних диференційних рівнянь на ЕОМ. Задача Коші**

**Тема:** Розв’язування звичайних диференційних рівнянь на ЕОМ. Задача коші

**Мета роботи**: вивчення алгоритмів чисельних методів розв’язання звичайних диференціальних рівнянь (ЗДР) і систем ЗДР на ЕОМ.

**Порядок виконання роботи**

1. Ознайомитись з відомими методами розв’язування ЗДР і систем ЗДР, користуючись рекомендованою літературою.

2. Скласти схему алгоритму розв’язування системи ЗДР методом, запропонованим викладачем.

. Скласти програму розв’язування системи ЗДР алгоритмічною мовою за вказівкою викладача.

4. Використовуючи дані відповідного варіанту (табл. 1), розв’язати систему ЗДР на ЕОМ (початкові дані та результати надрукувати).

5. Розв’язати систему ЗДР в середовищі MathCAD.

. Порівняти результати, які отримані при виконані пунктів 4,5, зробити висновки.

7. Оформити звіт.

Примітки: пункти 1-3 повинні бути виконані до початку занять у лабораторії.

Варіанти лабораторної роботи наведені в табл. 1.

**Опис роботи**

1. Для розв’язування *N* звичайних диференційних рівнянь необхідно розробити:

основну програму, яка виконує опис даних, введення початкових даних, відкликання підпрограми (або підпрограм), розв’язування системи ЗДР, вивід початкових даних та отриманих результатів на друкування;

підпрограму (або декілька підпрограм) розв’язування системи ЗДР методами запропонованими викладачем.

підпрограму побудови графіків;

підпрограму - функцію, яка складається з правих частин системи рівнянь.

. В список параметрів підпрограм слід включити такі параметри: вхідні

*N -* число ЗДР у системі;

*Х -* параметр інтегрування;

*Н -* крок інтегрування;

*P -* ім’я зовнішньої підпрограми, яка обчислює праві частини системи ЗДР;

*F -* масив з *N* дійсних чисел (значення похідних). Для деяких методів цей параметр може бути вихідним;

*E -* точність розв’язку (використовується в багатоточкових методах (методах прогнозу та корекції));

вихідні

*У -* масив із *N* дійсних чисел (розв’язок системи);

*Х -* параметр інтегрування.

Примітка: результати розв'язання системи звичайних диференційних рівнянь на ЕОМ бажано звести у таку таблицю:

Проміжок інтегрування *A-B*

Крок інтегрування *H=…*

1. Підпрограма розв’язування систем ЗДР не повинна, бажано, залежати від параметрів системи та від кількості рівнянь системи.

2. Результати розв’язку системи ЗДР необхідно представляти в табличній та графічній формах.

3. Якщо в завданні не задано проміжок інтегрування, його необхідно визначити експериментальним шляхом, аналізуючи результати інтегрування.

Розв’язати на відрізку [0,3] з кроком 0,1 задачу Коші для системи рівнянь 2-го порядку:



**Хід роботи**

1. Схема алгоритм розв’язання системи ЗДР методом Рунге-Кутта зображена на рисунку 1.





Рисунок 1 - Схема алгоритму методу Рунге-Кутта

Лістинг програми:

#include <stdio.h>

#include <conio.h>

#include <stdlib.h>

#include <malloc.h>

#include <math.h>

#define K 2eps=0.0001;struct E1

{x;y0;y1;

};

R\_Kutt (int N, float x, float \*y, float h);System (float x, float \*Y, float \*F);

main()

{i=0, N=0;t0, tn, h=0.05, \*X=NULL,\*y=NULL, \*\*Y=NULL, x=0.0;y1=0, y2=0.1;(«Vvedite granicu:\n»);(«%f % f»,&t0,&tn);(«Vvedite shag h:\n»);(«%f»,&h);(«\nVvedite y1\_0:»);(«%f»,&y1);(«\nVvedite y2\_0:»);(«%f»,&y2);

=(tn-t0)/h +1; ///kilkist vidrizkiv=t0;=(float\*) calloc (N, sizeof(float));(! X) exit(1);=(float\*\*) calloc (K, sizeof(float));(! Y) exit(1);(i=0; i<K; i++)

{[i]=(float\*) calloc (N, sizeof(float));(! Y[i]) exit(1);

}=(float\*) calloc (K, sizeof(float));(! y) exit(1);[0]=y1;[1]=y2;

// \*\*\*\*\*\*\*\*\*\*\*\*\*(i=0; i<N; i++)

{[0] [i]=y[0];[1] [i]=y[1];[i]=x;\_Kutt (K, x, y, h);=x+h;

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*(«\n# X Y1 Y2 \n\n»);(i=0; i<N; i++)

{(«%d\t%.4f\t\t%.4f\t\t%.4f\n», i, X[i], Y[0] [i], Y[1] [i]);

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\_getch();(X);(i=0; i<K; i++)(Y[i]);(Y);(y);0;

}

// \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*R\_Kutt (int N, float x, float \*y, float h)

{i=0;\*k1=NULL, \*k2=NULL, \*k3=NULL, \*k4=NULL, \*y\_temp=NULL, \*F=NULL;

=(float\*) calloc (N, sizeof(float));=(float\*) calloc (N, sizeof(float));=(float\*) calloc (N, sizeof(float));=(float\*) calloc (N, sizeof(float));\_temp=(float\*) calloc (N, sizeof(float));=(float\*) calloc (N, sizeof(float));(! k1 ||! k2 ||! k3 ||! k4 ||! y\_temp ||! F) exit(1);(x, y, F);(i=0; i<N; i++)

{[i]=h\*F[i];\_temp[i]=y[i]+0.5\*k1 [i];

}=x+h\*0.5;(x, y\_temp, F);(i=0; i<N; i++)

{[i]=h\*F[i];\_temp[i]=y[i]+0.5\*k2 [i];

}(x, y\_temp, F);(i=0; i<N; i++)

{[i]=h\*F[i];\_temp[i]=y[i]+k3 [i];

}=x+h\*0.5;(x, y\_temp, F);(i=0; i<N; i++)

{[i]=h\*F[i];[i]=y[i]+(k1 [i]+2.0\*k2 [i]+2.0\*k3 [i]+k4 [i])/6;

}

// \*\*\*Free memory\*\*\*(k1);(k2);(k3);(k4);(y\_temp);(F);

}System (float x, float \*y, float \*F)

{[0] =y[1];[1] = -0.01 \* exp(-x);

}

3. Тестування програми

Результат виконання програми зображено на рисунку 2.



Рисунок 2 - Результат виконання програми

диференціальний рівняння алгоритм

4. Виконання програми в середовищі MathCad

Результат виконання програми в середовищі MathCad зображено на рисунку 3.



Рисунок 3 - Результат виконання програми в середовищі MathCad

**Висновки**: вивчили алгоритми наближених методів розв’язання систем лінійних алгебраїчних рівнянь на ЕОМ.