**Зміст**

1. Короткі теоретичні відомості

2. Розробка алгоритму розв’язання задачі

3. Результати обчислень і оцінка похибки

Висновки

Література

Додаток

**1. Короткі теоретичні відомості**

Часто задачі техніки і природознавства математично зводяться до відшукання розв’язку певного диференціального рівняння (або системи таких рівнянь), який задовольняє певні початкові умови (задачі Коші). Про інтегрувати таке рівняння в скінченому вигляді вдається досить рідко. при цьому дістають здебільшого такий вигляд, до якого шукана функція входить неявно, а тому користуватись ним не зручно.

На практиці застосовують здебільшого наближене інтегрування диференціальних рівнянь. Воно дає змогу знайти наближений розв’язок задачі Коші або у вигляді певного аналітичного виразу (наприклад, ряду Тейлора), або у вигляді деякої таблиці значень.

Розглянемо окремі методи чисельного розв’язування задачі Коші для звичайного диференціального рівняння першого порядку, розв’язаного відносно похідної. Наближений розв’язок задачі Коші записують у вигляді певної таблиці значень.

Задача Коші полягає в тому, щоб знайти розв’язок *y(x)* диференціального рівняння

, (1.1)



який задовольняє початкову умову

(1.2)



Геометрично це означає, що треба знайти ту інтегральну криву *y(x)* рівняння (1.1), яка проходить через точку *(x0,y0).*

Задача Коші (1.1) – (1.2) має єдиний розв’язок, наприклад при виконанні умови такої теореми.

**Теорема** (Пікара). Якщо функція *f(x,y)* двох змінних х і у неперервна в замкнутому прямокутнику



з центром у точці (х0,у0) і задовольняє в ньому умову Лівшиця по змінній у, тобто існує число *K*>0, яке не залежить від х і у, таке, що

(1.3)



для будь-яких точок *(х1,у1)* і *(х2,у2)* , то існує єдина диференційована функція , яка є розв’язком диференціального рівняння (1.1). Цей розв’язок визначений і неперервно диференційований принаймні на відрізку [x0-h; x0+h], де



(1.4)



Розглянемо так звані однокрокові чисельні методи розв’язування задачі Коші (1.1)-(1.2), в яких, щоб знайти наближений розв’язок у точці *хk+1=xk+h*, досить знайти її розвязок в точці *хk*.

І оскільки розв’язок задачі в точці *х0* відомий з початкових умов, то ці методи дають змогу послідовно обчислити значення розв’язку в наступних точках х1=х0+h, x2=x1+h,...

Окремим представником однокрокових чисельних методів є методи типу Ейлера. Надалі припускатимемо, що функція *f(x,y)* рівняння (1.1) задовольняє умови теореми Пікара [1].

*Метод Ейлера*

Нехай на відрізку *[x0,x0+l]* треба знайти чисельний розв’язок задачі Коші(1.1)-(1.2). Для цього відрізок *[x0,x0+l]* поділимо на n (для простоти) рівних частин точками

*х0, х1, х2,..., хn=x0+l,* де *хk=x0+kh (k=0,1,2,...,n), .*



Величину *h* називають кроком чисельного інтегрування диференціального рівняння (1.1).

Розв’язати задачу (1.1)-(1.2) чисельно – це означає для заданої послідовності *х0, х1,…, хn=b=x0+l* незалежної змінної *х* та числа *у0* знайти числову послідовність *у1, у2,…, уn*, тобто для заданої послідовності значень незалежної змінної xk=x0+kh (k=0, 1, ..., n) побудувати таблицю наближених значень шуканого розв’язку задачі Коші.

Якщо наближений розв’язок задачі (1.1)-(1.2) в точці хk відомий, то, проінтегрувавши рівняння (1.1) в межах від хk до хk+1, знайдемо його розв’язок в точці хk+1 за формулою:

(1.5)



Саме ця формула є вихідною для побудови багатьох чисельних методів розв’язування задачі (1.1) - (1.2). Якщо інтеграл у правій частині формули (1.5) обчислити за формулою лівих прямокутників, то знайдемо

(1.6)



Відкинувши в цій рівності доданок порядку О(h2), дістанемо розрахункову формулу:

(1.7)



яку називають *формулою Ейлера.* уk i y(xk) – відповідно наближене і точне значення шуканого розв’язку задачі (1.1) і (1.2) у точці хk. Різницю уk-y(xk) називають похибкою наближеного значення уk у точці xk.

Оскільки дотична до графіка функція у(х) в точці (xk,yk) матиме вигляд:

або



Звідси для ординати точки уk+1 перетину цієї дотичної з прямою х=хk+1 дістанем формулу (1.7), а це означає, що на кожному з відрізків [xk,xk+1], (k=0, 1, 2, ..., n-1 ) інтегральна крива наближено замінюється відрізком дотичної до неї в точці (xk,yk). Якщо в площині Оху позначити точки Мk(xk;yk), k=0, 1, 2,...,n і сполучити їх по порядку відрізками, то дістанемо ламану (її називають ламаною Ейлера), яка наближено зображує графік шуканого розв’язку задачі (1.1) – (1.2). У цьому і полягає геометричний зміст методу Ейлера (див. рис. 1)

l3

у

l1

l2

l4

у4

у=f(x)

у1

у0

Рисунок 1.

х4

х1

х

0

х0

Зазначимо, що похибка методу Ейлера на кожному кроці є величина порядку О(h2). Точність методу досить мала і переходом від точки xk до точки xk+1 її похибка систематично зростає.

*Виправлений метод Ейлера.*

Якщо інтеграл у правій частині формули (1.5) обчислити за формулою середніх прямокутників, тобто значення підінтегральної функції f(x,y(x)) обчислити в точці

, то знайдемо



(1.8)



Величину невідомого значення функції у() обчислимо за формулою (1.6) з кроком . Матимемо:



Підставивши це значення у() в (1.8), дістанемо



Відкинувши тут доданок пропорційний h3, матимемо



Розрахункові формули вдосконаленого методу Ейлера можна записати у вигляді



Отже, в удосконаленому методі Ейлера спочатку за метод Ейлера обчислюють наближений розв’язок у задачі (1.1)-(1.2) в точці а потім наближений розв’язок уk+1 у точці хk+1; на кожному кроці інтегрування праву частину рівняння (1.1) обчислюють двічі (у точках (хk,уk) і ()).



Геометрично це означає, що на відрізку [xk,xk+1] графік інтегральної кривої задачі (1.1)-(1.2) замінюється відрізком прямої, яка проходить через точку (xk,yk) і має кутовий коефіцієнт k=. Іншими словами, ця пряма утворює з додатним напрямом осі Ох кут .



0

xk

xk+

xk+1

x

Y

Yk+

Yk

y=f(x)

l2

l1

М

Yk+1

Рисунок 2.

Що ж до точки (), то це точка перетину дотичної до інтегральної кривої задачі (1.1)-(1.2) в точці (хk,yk) з прямою Похибка на кожному кроці має порядок *О(h3)*.



*Модифікований метод Ейлера.*

Якщо інтеграл в правій частині формули (1.5)обчислити за формулою трапеції, то матимемо

(1.11)



Невідоме значення у(хk+1), що входить до правої частини цієї рівності, можна обчислити за формулою (1.7). Підставивши його в праву частину рівності (1.11), дістанемо рівність:



Звідси для удосконаленого методу Ейлера-Коші матимемо такі розрахункові формули:

(1.12)



(1.13)



Отже, і в цьому методі на кожному кроці інтегрування праву частину рівняння (1.1) обчислюють двічі: спочатку за методом Ейлера (формула (1.12)) обчислюють наближене значення шуканого розв’язку у точці хk+1, яке потім уточнюють за формулою (1.13). Похибка методу на кожному кроці має порядок О(h3).



Така побудова наближеного розв’язку задачі (1.1)1(1.2) з геометричної точки зору означає, що на відрізку [xk,xk+1] графік інтегральної кривої наближають відрізком прямої, яка проходить через точку (xk,yk) і має кутовий коефіцієнт Тобто ця пряма утворює з додатним напрямком осі Ох кут



M

у

l1

l

l

l2

k+1

yk+1

yk

0

хk

хk+1

х

Рисунок 3.

Координати точки (xk+1,) визначають як точку перетину дотичної у=уk+f(xk,yk)(x-xk) до графіка інтегральної кривої задачі (1.1)-(1.2) в точці (xk,yk) з прямою х=хk [2].



**2. Розробка алгоритму розв’язання задачі**

Стандартний спосіб розв’язання задачі Коші чисельними однокроковими методами – це зведення диференціальних рівнянь n-го порядку до систем з n рівнянь 1-го порядку і подальшого розв’язання цієї системи стандартними однокроковими методами:

Для рівняння введемо заміну тоді для даного рівняння можна записати еквівалентну систему із двох рівнянь:



Запишемо для кожного з цих рівнянь ітераційне рівняння:

для модифікованого методу :Ейлера:



для виправленого методу Ейлера:



Таким чином знаходиться масив точок функції ymn з різними кроками тобто n1=(b-a)/0,1=10+1 раз з кроком 0,1 і n2=(b-a)/0,05 раз з кроком 0,05. Це необхідно для оперативного визначення похибки за методом Рунге (екстраполяції Рідчардсона) [3].

Загальний вигляд похибки для цих двох методів , де с визначається саме за методом Рунге , звідки с на кожному кроці обчислень знаходиться за формулою:



.



Знаючи с можна знайти локальну похибку і просумувавши її по всьому діапазону інтегрування визначити загальну похибку обчислень.

Мовою програмування було обрано Turbo C++. Вона виявилась найзручнішою із тих мов, в яких мені доводилось працювати.

Програма складається з трьох допоміжних функцій float f(x,y,z), void eylermod() i eylerisp(). eylermоd() реалізовує модифікований метод Ейлера, eylerisp() – виправлений метод, а функція f(x,y,z) повертає значення другої похідної рівняння.

Лістинг програми приведено в додатку.

**3. Результати обчислень і оцінка похибки**

Результатом розв’язання задачі Коші являється функція. В даному випадку отримати цю функцію в аналітичному вигляді обчислювальні однокрокові методи не дозволяють. Вони представляють функцію в табличному вигляді, тобто набір точок значень х і відповідних їм значень функції у(х). Тому для більшої наглядності було вирішено по цим точкам намалювати графіки функцій у(х) для кожного з методів окремо (дивись рисунок 4). На тому ж малюнку виведені значення похибок для кожного методу окремо. На рисунку 5 виведено значення функції у(х) в дискретному вигляді з кроком h1=0.1.

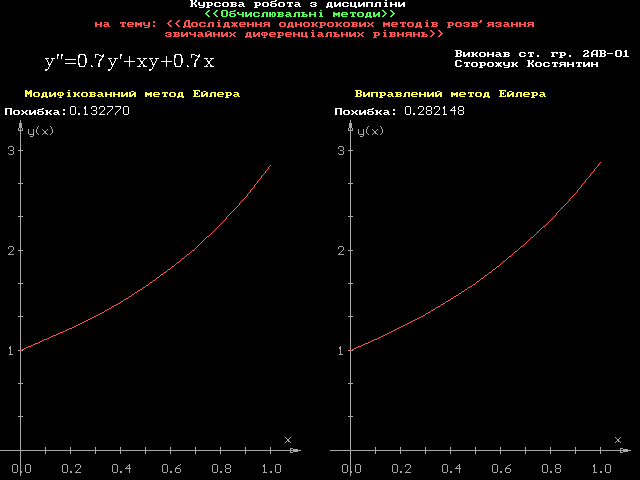


Рисунок 4.

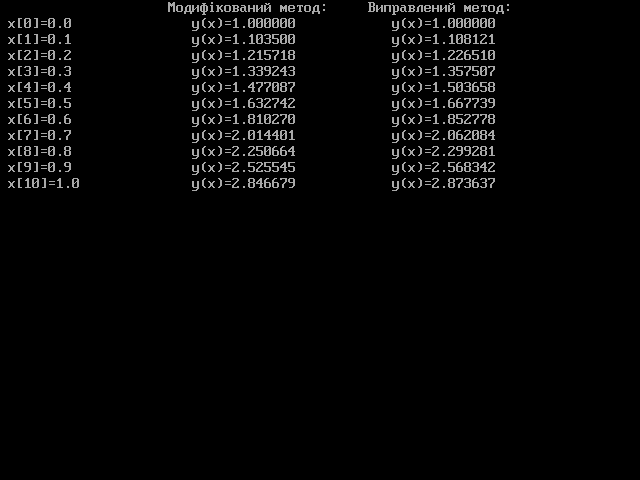


Рисунок 5.

**Висновки**

В даній курсовій роботі я ознайомився з однокроковими методами розв’язання звичайних диференціальних рівнянь. Завдяки їй я остаточно розібрався застосовуванням цих методів до розв’язання диференціальних рівнянь вищих порядків на прикладі рівняння другого порядку.

**Література**

1. Мудров А.Е. Численные методы для ПЭВМ на языках Бейсик, Фортран и Паскаль. – Томск: МП «Раско», 1991. – 272 с.

2. Бортків А.Б., Гринчишин Я.Т. Turbo Pascal: Алгоритми і програми: чисельні методи в фізиці і математиці. Навчальний посібник. – К.: Вища школа, 1992. – 247 с.

3. Квєтний Р.Н. Методи комп’ютерних обчислень. Навчальний посібник. – Вінниця: ВДТУ, 2001 – 148 с.

**Додаток**

**Лістинг програми**

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include<graphics.h>

float f(float x,float y,float z)

{return 0.7\*z+x\*y+0.7\*x;}

float h1=0.1;

float h2=0.05;

float a=0;

float b=1;

float x2[21],ye2[21],ym1[11],zm2[21],ym2[21],ye1[11];

float ze1[11],zm1[11],ze2[21],x1[11],yi1[11],yi2[21];

float zi1[11],zi2[21];

int n1=(b-a)/h1;

int n2=(b-a)/h2;

void eylermod()

{// printf("[0] %5.2f %5.2f %5.2f",x2[0],y2[0],z2[0]);

// moveto((x2[0])\*100,480-((ym2[0])\*100));

for(int i=1;i<=n2+1;i++)

{x2[i]=x2[i-1]+h2;

ze2[i]=ze2[i-1]+h2\*f(x2[i-1],ye2[i-1],ze2[i-1]);

ye2[i]=ye2[i-1]+h2\*ze2[i-1];

zm2[i]=zm2[i-1]+(h2/2)\*(f(x2[i-1],ye2[i-1],zm2[i-1])+f(x2[i],ye2[i],ze2[i]));

ym2[i]=ym2[i-1]+(h2/2)\*(ze2[i]+zm2[i-1]);

// printf("\n[%d] %5.2f %5.2f %5.2f",i,x2[i],ye2[i],ym2[i]);

// setcolor(YELLOW);

// lineto((x2[i])\*100,480-((ym2[i])\*100));}

moveto((x1[0])\*250+20,480-((ym1[0])\*100)-30);

for(i=1;i<=n1+1;i++)

{x1[i]=x1[i-1]+h1;

ze1[i]=ze1[i-1]+h1\*f(x1[i-1],ye1[i-1],ze1[i-1]);

ye1[i]=ye1[i-1]+h1\*ze1[i-1];

zm1[i]=zm1[i-1]+(h1/2)\*(f(x1[i-1],ye1[i-1],zm1[i-1])+f(x1[i],ye1[i],ze1[i]));

ym1[i]=ym1[i-1]+(h1/2)\*(ze1[i]+zm1[i-1]);

// printf("\n[%d] %5.2f %5.2f %5.2f",i,x1[i],ye1[i],ym1[i]);

setcolor(12);

lineto((x1[i])\*250+20,480-((ym1[i])\*100)-30);}

float c;

float s=0;

for(i=0;i<=n1+1;i++)

{c=(ym2[i\*2]-ym1[i])/(h1\*h1\*h1-h2\*h2\*h2);

s+=c\*h1\*h1\*h1;}

char \*ch;

sprintf(ch,"%f",fabs(s));

setcolor(15);

settextstyle(0,0,1);

outtextxy(5,108,"Похибка:");

settextstyle(2,0,5);

outtextxy(70,102,ch);}

void eylerisp()

{// printf("[0] %5.2f %5.2f %5.2f",x2[0],y2[0],z2[0]);

// moveto((x2[0])\*100,480-((ym2[0])\*100));

for(int i=1;i<=n2+1;i++)

{x2[i]=x2[i-1]+h2/2;

ze2[i]=ze2[i-1]+h2/2\*f(x2[i-1],ye2[i-1],ze2[i-1]);

ye2[i]=ye2[i-1]+h2/2\*ze2[i];

zi2[i]=zi2[i-1]+h2\*f(x2[i],ye2[i],ze2[i]);

yi2[i]=yi2[i-1]+h2\*zi2[i];

x2[i]+=h2/2;

// printf("\n[%d] %5.2f %5.2f %5.2f",i,x2[i],ye2[i],ym2[i]);

// setcolor(YELLOW);

// lineto((x2[i])\*100,480-((ym2[i])\*100));}

moveto((x1[0])\*250+350,480-((yi1[0])\*100)-30);

for(i=1;i<=n1+1;i++)

{x1[i]=x1[i-1]+h1/2;

ze1[i]=ze1[i-1]+h1/2\*f(x1[i-1],ye1[i-1],ze1[i-1]);

ye1[i]=ye1[i-1]+h1/2\*ze1[i];

zi1[i]=zi1[i-1]+h1\*f(x1[i],ye1[i],ze1[i]);

yi1[i]=yi1[i-1]+h1\*zi1[i];

x1[i]+=h1/2;

// printf("\n[%d] %5.2f %5.2f %5.2f",i,x1[i],ye1[i],ym1[i]);

setcolor(12);

lineto((x1[i])\*250+350,480-((yi1[i])\*100)-30);}

float c;

float s=0;

for(i=0;i<=n1+1;i++)

{c=(yi2[i\*2]-yi1[i])/(h1\*h1\*h1-h2\*h2\*h2);

s+=c\*h1\*h1\*h1;}

char \*ch;

sprintf(ch,"%f",fabs(s));

setcolor(15);

settextstyle(0,0,1);

outtextxy(335,108,"Похибка:");

settextstyle(2,0,5);

outtextxy(405,102,ch);}

void main()

{float c=0,s=0;

int gdriver = DETECT, gmode, errorcode;

initgraph(&gdriver, &gmode, "");

cleardevice();

x2[0]=x1[0]=a;

ye2[0]=ye1[0]=1;

ym2[0]=ym1[0]=1;

ze2[0]=ze1[0]=1;

zm2[0]=zm1[0]=1;

yi2[0]=yi1[0]=1;

zi2[0]=zi1[0]=1;

char v=50;

while(v!=27)

{//setgraphmode(getgraphmode());

setbkcolor(16);

outtextxy(190,0,"Курсова робота з дисциплiни");

setcolor(10);

outtextxy(205,10,"<<Обчислювальнi методи>>");

setcolor(12);

outtextxy(95,20,"на тему: <<Дослiдження однокрокових методiв розв'язання");

outtextxy(165,30,"звичайних диференцiальних рiвнянь>>");

setcolor(14);

outtextxy(25,90,"Модифiкованний метод Ейлера");

outtextxy(355,90,"Виправлений метод Ейлера");

setcolor(15);

outtextxy(455,50,"Виконав ст. гр. 2АВ-01");

outtextxy(455,60,"Сторожук Костянтин");

settextstyle(8,0,1);

outtextxy(45,45,"y''=0.7y'+xy+0.7x");

settextstyle(0,0,1);

setcolor(7);

line(20,475,20,120); //левая ось у

line(0,450,300,450); //левая ось х

line(350,475,350,120);//правая ось у

line(330,450,630,450);//правая ось х

line(20,120,18,130);

line(20,120,22,130); //стрелки оу

line(18,130,22,130);

line(300,450,290,448);

line(300,450,290,452); //срелки ох

line(290,448,290,452);

line(350,120,348,130);

line(350,120,352,130); //стрелки оу

line(348,130,352,130);

line(630,450,620,448);

line(630,450,620,452); //срелки ох

line(620,448,620,452);

char t[5];

char m[5];

settextstyle(2,0,5);

outtextxy(285,430,"x");

outtextxy(28,122,"y(x)");

outtextxy(615,430,"x");

outtextxy(358,122,"y(x)");

for(float i=0;i<11;i++)

{line(20+i\*25,447,20+i\*25,453);

if(i<10)line(18,450-(i\*50)/1.5,22,450-(i\*50)/1.5);

sprintf(t,"%.1f",i/10);

if(int(i)%2==0) outtextxy(i\*25+12,460,t);

sprintf(m,"%.0f",i+1);

if(i<3)outtextxy(8,342-i\*100,m);}

for(i=0;i<11;i++)

{line(350+i\*25,447,350+i\*25,453);

if(i<10)line(348,450-(i\*50)/1.5,352,450-(i\*50)/1.5);

sprintf(t,"%.1f",i/10);

if(int(i)%2==0) outtextxy(i\*25+342,460,t);

sprintf(m,"%.0f",i+1);

if(i<3)outtextxy(338,342-i\*100,m);}

settextstyle(0,0,1);

eylermod();

eylerisp();

v=getch();

if(v==27)break;

restorecrtmode();

setgraphmode(getgraphmode());

printf("\t\t Модифiкований метод:\t Виправлений метод:");

for(i=0;i<n1+2;i++)

{printf("\n x[%.f]=%.1f\t\ty(x)=%f \t\t y(x)=%f",i,x1[i],ym1[i],yi1[i]);}

settextstyle(0,0,1);

v=getch();

cleardevice();}

closegraph();}