МІНІСТЕРСТВО ОСВІТИ І НАУКИ
НАВЧАЛЬНО-НАУКОВИЙ КОМПЛЕКС
«ІНСТИТУТ ПРИКЛАДНОГО СИСТЕМНОГО АНАЛІЗУ»
НАЦІОНАЛЬНОГО ТЕХНІЧНОГО УНІВЕРСИТЕТУ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
КАФЕДРА МАТЕМАТИЧНИХ МЕТОДІВ СИСТЕМНОГО АНАЛІЗУ

Лабораторна робота № 8

з курсу «Чисельні методи»

Тема: Крайова задача другого порядку

Варіант 16

Виконав: студент 3 курсу

групи КА-32

Пустовіт Д.Т.

Прийняв:

Коновалюк М.М.

Київ – 2015

Завдання:

Постановка задачі.
Розв'язати крайову задачу

$$Ay = f; \quad x_1 < x < x_2;$$
$$a_1 y = f_1; \quad x = x_1;$$
$$a_2 y = f_2; \quad x = x_2;$$

де $Ay = y'' + p(x)\, y' + q(x)\, y = f(x)$; $p(x), q(x)$ - задані функції; a_1 , a_2 - задані оператори; x_1 , x_2 - задані числа.

16	$y'' + 2\,x\,y' - 2\,y$	$y'(2)$	$0{,}4\,y(2{,}3) - y'(2{,}3)$

16	1.135	-2.643	-1.410	-2.767	1.777

Математичний розв'язок задачі:

Метод кінцевих різниць для задачі

$$\begin{aligned} & y'' + p(x)\cdot y' + q(x)\cdot y = f(x), \\ & a \le x \le b, \\ & \left.\begin{aligned} \alpha_0 \cdot y(a) + \alpha_1 \cdot y'(a) = A, \\ \beta_0 \cdot y(b) + \beta_1 \cdot y'(b) = B, \end{aligned}\right\} \\ & (|\alpha_0| + |\alpha_1| \ne 0, \quad |\beta_0| + |\beta_1| \ne 0), \end{aligned}$$

$$y'_i \approx \frac{y_{i+1} - y_{i-1}}{2h}, \qquad y''_i \approx \frac{y_{i+1} - 2y_i + y_{i-1}}{h^2}, \qquad i = 1,2,\ldots,n.$$

Розібємо відрізок [a,b] на n часин з довжиною h. Похідні вважатимемо наближеними кінцевими різницями. Тоді похибку можна обчислити як $r_i(h) = -\frac{h^2}{6}\cdot y'''(\xi), \;\; x_{i-1} < \xi < x_{i+1},$

Підставляючи дані різниці у початкову задачу отримаємо наступну лінійну систему:

$$\begin{cases} y_{i+1} + m_i \cdot y_i + n_i \cdot y_{i-1} = \dfrac{2\cdot h^2}{2 + h\cdot p_i}\cdot f_i, \quad i = 1,2,\ldots,n. \\ \alpha_0 \cdot y_0 + \alpha_1 \cdot \dfrac{y_1 - y_0}{h} = A, \\ \beta_0 \cdot y_n + \beta_1 \cdot \dfrac{y_{n+1} - y_{n-1}}{2\cdot h} = B, \end{cases}$$

, де $\quad m_i = \dfrac{2\cdot q_i \cdot h^2 - 4}{2 + h\cdot p_i}, \quad n_i = \dfrac{2 - h\cdot p_i}{2 + h\cdot p_i}.$

Цю систему можна легко розвязати методом прогонки:

Розвязок будемо шукати з умови $y_{i+1} = c_i \cdot (d_i - y_{i+2}),$ Пошук векторів c та d зветься прямим ходом методу прогонки. Ці вектори можна знайти рекурентно, за формулами:

$$c_1 = \frac{\alpha_1 - \alpha_0 \cdot h}{m_1\cdot(\alpha_1 - \alpha_0\cdot h) + n_1 \cdot \alpha_1}, \qquad d_1 = \frac{2\cdot f_1 h^2}{2 + p_1 \cdot h} + n_1 \frac{A\cdot h}{\alpha_1 - \alpha_0 \cdot h}.$$

$$c_i = \frac{1}{m_i - n_i \cdot c_{i-1}}, \qquad d_i = \frac{2 f_i \cdot h^2}{2 + h\cdot p_i} - n_i \cdot c_{i-1}\cdot d_{i-1},$$

Тоді розвязок одержимо в результаті зворотнього ходу:

$$y_n = \frac{2\cdot B\cdot h - \beta_1 \cdot (d_n - c_{n-1}\cdot d_{n-1})}{2\cdot \beta_0 \cdot h + \beta_1 \cdot (c_{n-1} - \dfrac{1}{c_n})}.$$

$$y_i = c_i \cdot (d_i - y_{i+1}), \qquad i = n-1,\; n-2, \ldots, 1,$$
$$y_0 = \frac{A\cdot h - \alpha_1 \cdot y_1}{\alpha_0 \cdot h - \alpha_1}.$$

Лістинг:

```
#include "stdafx.h"
#include <iostream>
#include <vector>
#include <fstream>
#include <cmath>

#define ca 1.135
#define cb -2.643
#define cc -1.41
#define cd -2.767
#define ce 1.777
#define cf1 2.09303
#define cf2 -3.3906
#define calp0  0
#define calp1  1
#define cbet0  0.4
#define cbet1  -1
#define cx1 2
#define cx2 2.3


using namespace std;

inline double answ(double x){
        return ca*pow(x, 2) + cb*x + cc + 1.0 / (cd*x + ce);
}
inline double dansw(double x){
        return 2 * ca * x + cb - cd / pow(ce + cd*x, 2);
}
inline double ddansw(double x){
        return 2 * ca + 2 * cd * cd / pow(ce + cd*x, 3);
}
void findparams(){
        cout << "cf1=" << dansw(cx1);
        cout << "\ncf2=" << cbet0*answ(cx2) + cbet1*dansw(cx2)<<endl;
}

double err(vector<double> y, double a, double b, int n){
        ofstream fout("error.txt");
        fout << "помилка методу кінцевих різниць:" << endl;
        double max=0,x = a, e, h = (b - a) / (n - 1);
        for (int i = 0; i < n; i++){
                e = fabs(answ(x) - y[i]);
                if (e > max) max = e;
                fout << " err[ " << x << " ] = " << e << endl;
                x += h;
        }
        return max;
}

}
void progonka(vector<double> &y, vector<double> m, vector<double> n, vector<double>
p,vector<double> f, double h, int N){
        vector<double> c(N+1), d(N+1);
        c[1] = (calp1 - calp0*h) / (m[1] * (calp1 - calp0*h) + n[1] * calp1);
        d[1] = (2 * f[1] * h*h) / (2 + p[1] * h) + n[1] * cf1*h / (calp1 - calp0*h);
        for (int i = 2; i <= N; i++){
                c[i] = 1.0 / (m[i] - n[i] * c[i - 1]);
                d[i] = 2 * f[i] * h*h / (2 + h*p[i]) - n[i] * c[i - 1] * d[i - 1];
        }
        y[N] = (2 * cf2*h - cbet1*(d[N] - c[N - 1] * d[N - 1])) / (2 * cbet0*h + cbet1*(c[N - 1] -
1.0 / c[N]));
        for (int i = N - 1; i > 0; i--) y[i] = c[i] * (d[i] - y[i + 1]);
        y[0] = (cf1*h - calp1*y[1]) / (calp0*h - calp1);
}


void kin_rizn(double a, double b, int n, vector<double> &y){
        double h = (b - a) / (n-1);
        double x=a, q=-2.0;
        vector<double> p(n), f(n), t1(n),t2(n);
        for (int i = 0; i < n; i++){
                p[i] = 2 * x;
                t1[i] = (2 * q*h*h - 4) / (2+h*p[i]);       //m
                t2[i] = (2-h*p[i])/(2+h*p[i]);                      //n
                f[i] = ddansw(x) + p[i] * dansw(x) + q * answ(x);
                x += h;
```

```
	}
	progonka(y,t1, t2, p,f, h, n-1);
}

int _tmain(int argc, _TCHAR* argv[])
{
	int n = 10;
	vector<double> y(n);
	kin_rizn(cx1,cx2,n,y);
	cout << "Max error value = " << err(y, cx1, cx2, n);
	system("Pause");
	return 0;
}
```

Результати роботи

(для n=10)

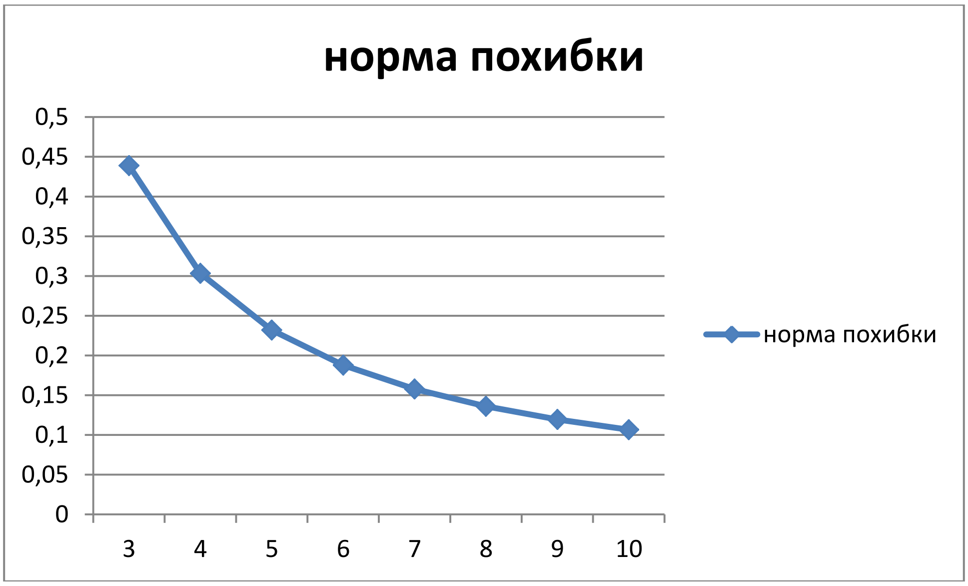
```
err[ 2 ] = 0.095063
err[ 2.03333 ] = 0.0961676
err[ 2.06667 ] = 0.0973324
err[ 2.1 ] = 0.0985497
err[ 2.13333 ] = 0.0998126
err[ 2.16667 ] = 0.101115
err[ 2.2 ] = 0.102452
err[ 2.23333 ] = 0.103819
err[ 2.26667 ] = 0.105211
err[ 2.3 ] = 0.106625
```

(для n=5)

```
err[ 2 ] = 0.206666
err[ 2.075 ] = 0.212281
err[ 2.15 ] = 0.218457
err[ 2.225 ] = 0.22504
err[ 2.3 ] = 0.231914
```

(для n=3)

```
err[ 2 ] = 0.390612
err[ 2.15 ] = 0.413224
err[ 2.3 ] = 0.438949
```



Висновок:

В результаті проробленої роботи було реалізовано метод кінцевих різниць. Був отриманий очевидний результат що зі збільшенням кількості точок похибка зменшується, а також як видно з графіка було підтверджено що вона зменшується квадратично (дивись мат. розв’язок). Також слід зауважити що величина похибки сильно залежить від виду крайових умов, і є меншою, якщо умови накладаються на саму функцію.