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 № 9
з курсу «Чисельні методи»
Тема: ДРЧП

Варіант 16

Виконав: студент 3 курсу
групи КА-32
Пустовіт Д.Т.
Прийняв:
Коновалюк М.М.

Київ – 2015

Завдання:

Розв'язати рівняння гіперболічного типу

$u_{tt} = u_{xx} + F(t,x), \quad 0 < x < L = 1, \quad (2)$

для функції $u(t,x)$ з початковими $u(0,x) = u_0(x)$, $u_t(0,x) = 0$

та крайовими $u(t,0) = u_1(t)$; $u(t,L) = u_2(t)$

умовами.

№вар	$u_0(x)$
11	$(1-x)\cos(\pi x/2)$
12	$0{,}5x(x-1)$
13	$0{,}5(x^2+1)$
14	$(x+1)\sin(\pi x/2)$
15	$x^2\cos(\pi x)$
16	$(1-x^2)\cos(\pi x)$

2. Вважаючи точним розв'язком задачи (2) функцію

$$u(t,x) = u_0(x)\cos(\pi t)$$

де $u_0(x)$ - функція з таблиці 4, підставити її у рівняння (2), початкові та крайові умови та знайти функції $F(t,x)$, $u_0(x)$, $u_1(t)$, $u_2(t)$, що відповідатимуть Вашому варіантові.

Математичний розв'язок задачі:

Для початку знайдемо всі крайові умови (на зворотній стороні аркуша).

Розіб'ємо відрізок $[0,L]$ на N відрізків довжиною $h = \dfrac{L}{N}$, $x_n = nh$, $n = 0,\dots,N$.

Позначимо $t_k = k\tau$, $k = 0,1,2,\dots$. U_n^k означатиме наближення до $U(nh,k\tau) = U(x_n,t_k)$.

Чисельне розв'язання задачи полягатиме в обчисленні наближеного значення U_n^k для всіх $n = 0,\dots,N$, $k = 0,1,2,\dots$.

<u>Симетрична різницева схема з вагами.</u>

Нехай

$$\frac{\partial^2 U}{\partial t^2} \mapsto U_{t\bar t} = \frac{U^{k+1} - 2U^k + U^{k-1}}{\tau^2};$$

$$\frac{\partial^2 U}{\partial x^2} \mapsto U_{x\bar x} = \frac{U_{n+1} - 2U_n + U_{n-1}}{h^2}.$$

Тоді рівняння (1) можна апроксимувати схемою

$$\frac{U_n^{k+1} - 2U_n^k + U_n^{k-1}}{\tau^2} =$$

$$= c^2\left(\sigma\frac{U_{n+1}^{k+1} - 2U_n^{k+1} + U_{n-1}^{k+1}}{h^2} + (1-2\sigma)\frac{U_{n+1}^{k} - 2U_n^{k} + U_{n-1}^{k}}{h^2} + \sigma\frac{U_{n+1}^{k-1} - 2U_n^{k-1} + U_{n-1}^{k-1}}{h^2}\right) +$$

$$+ \left(\sigma f_n^{k+1} + (1-2\sigma) f_n^k + \sigma f_n^{k-1}\right).$$

(4)

Тут σ – параметр.

Схема (4) є лінійним алгебраїчним рівнянням відносно дев'яти значень U_n^k (з різними n та k), з яких U_{n-1}^{k+1}, U_n^{k+1}, U_{n+1}^{k+1} вважаються невідомими, а значення кроків k та $k-1$ – відомими.

Початкова умова на $U(x,0)$ дозволяє знайти всі U_n^k для $k=0$:

$$U_n^0 = U_0(nh),\ n = 0,\dots,N$$

(5)

Крім того, для довільного $k > 0$ крайові умови дозволять знайти

$$U_0^k = \gamma_1(k\tau) \text{ та } U_N^k = \gamma_2(k\tau),\ k = 1,2,\dots.$$

(6)

Для знаходження $\{U_n^1\}$ використаємо ряд Тейлора:

$$U(x,\tau) = U(x,0) + \tau\frac{\partial U}{\partial t}(x,0) + \frac{\tau^2}{2}\frac{\partial^2 U}{\partial t^2}(x,0) + O(\tau^3).$$

Приймемо

$$U_n^1 = U_n^0 + \tau v_0(x_n) + \frac{\tau^2}{2}\frac{\partial^2 U}{\partial t^2}(x_n,0) =$$

$$= U_n^0 + \tau v_0(x_n) + \frac{\tau^2}{2}\left(c^2\frac{\partial^2 U}{\partial x^2} + f(x_n,0)\right) =$$

$$= U_n^0 + \tau v_0(x_n) + \frac{\tau^2}{2}\left(c^2\frac{U_{n+1}^0 - 2U_n^0 + U_{n-1}^0}{h^2} + f(x_n,0)\right) + O(\tau^2 h^2)$$

(7)

Рівняння (6) та (4), $n = 1,\dots,N-1$, утворюють тридіагональну СЛАР, яку можна розв'язувати методом прогонки. Покладаючи $k = 1$ та знайшовши всі U_n^0 та U_n^1 за (5) та (7), розв'язують цю систему та знаходять всі U_n^2, $n = 0,\dots,N$.

Далі збільшують k на 1 та знаходять вектор розв'язку для наступного часового кроку.

Схема (4) дає похибку $e = O(\tau^2, h^2)$. За умови $\sigma \ge \frac{1}{4}$ схема є стійкою за будь-яких τ та h. За $\sigma < \frac{1}{4}$ схема є стійкою за умови $\tau \le \frac{h}{c(1-4\sigma)}$. Обидва випадки можна об'єднати нерівністю

$$\sigma \ge \frac{1}{4} - \frac{h}{4\tau},$$

за виконанням якої треба слідкувати, задаючи параметри схеми.

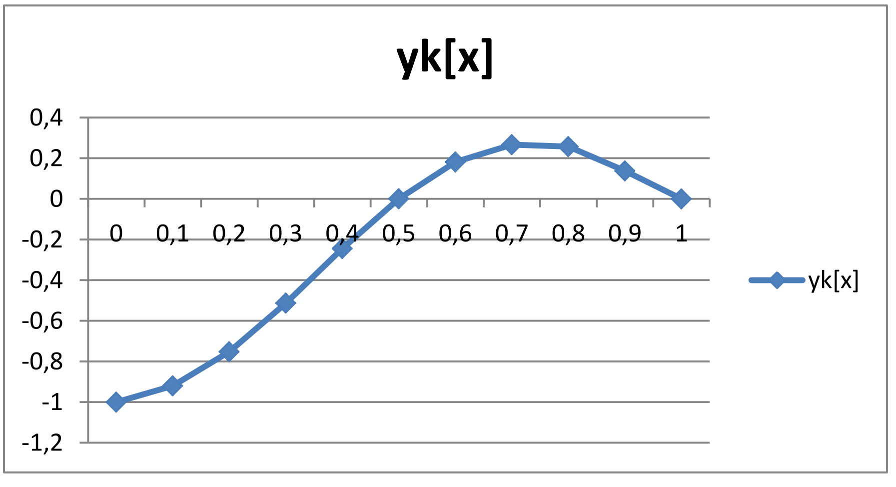
Лістинг:

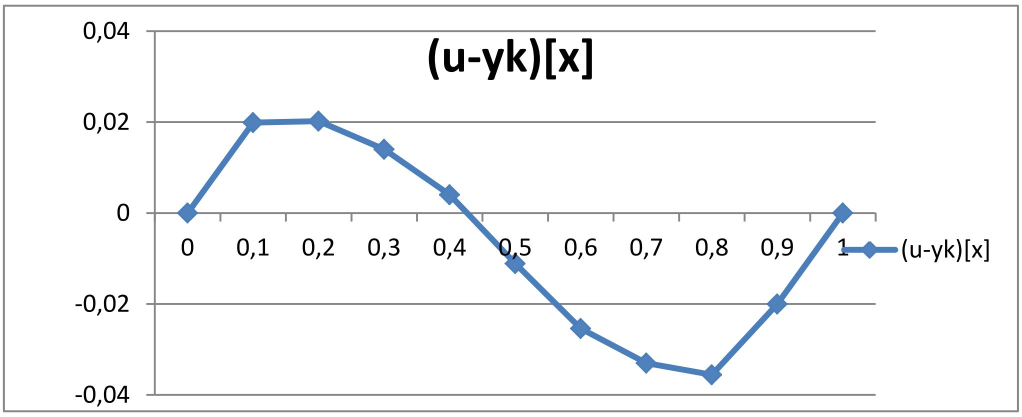
```
// lab_9.cpp : Defines the entry point for the console application.
#include "stdafx.h"
#include <iostream>
#include <fstream>
#define _USE_MATH_DEFINES
#include <math.h>
#include <vector>
#define L 1.0
#define T 5.0
#define p M_PI

using namespace std;

double u0(double x){
		return (1 - x*x)*cos(p*x);
}
double u1(double t){
		return cos(p*t);
}
```

```
double u2(double t){
	return 0;
}
double F(double x, double t){
	return 2 * cos(p*t)*cos(p*x)-4*p*x*cos(p*t)*sin(p*x);
}
double ansv(double x, double t){
	return u0(x)*cos(p*t);
}



int _tmain(int argc, _TCHAR* argv[])
{
	int N = 100, M=1000;
	double dt = T / (M-1), dx = L /(N-1);
	double sigma=0.00001;
	vector<vector<double>> y;
	y.resize(N); // к-ть строк
	for (int i = 0; i < y.size(); i++) y[i].resize(M);
	for (int i = 0; i < N; i++) y[i][0] = u0(dx*i);
	for (int i = 0; i < M; i++){
		y[0][i] = u1(dt*i);
		y[N-1][i] = u2(dt*i);
	}
	for (int i = 1; i < N-1; i++)	{
		y[i][1] = y[i][0] + (dt*dt/2)*( ( (y[i + 1][0] - 2 * y[i][0] + y[i - 1][0]) / (dx*dx) ) + F(i*dx, 0) );
	}
	vector<double> a(N), b(N), z(N), w(N), q(N), s(N);
	int k = 1;
	while (k < M-1)
	{
		for (int i = 1; i < N-1; i++)
		{
			a[i] = z[i] = (sigma*dt*dt) / (dx*dx);
			b[i] = -(1 + 2 * ((sigma*dt*dt) / (dx*dx)));
			s[i] = (2 * y[i][k] - y[i][k - 1] + (((1 - 2 * sigma)*dt*dt) / (dx*dx))*(y[i + 1][k] - 2 * y[i][k] + y[i - 1][k])
				+ ((sigma*dt*dt) / (dx*dx))*(y[i + 1][k - 1] - 2 * y[i][k - 1] + y[i - 1][k - 1])
				+ dt*dt*(sigma*F(i*dx, (k + 1)*dt) + (1 - 2 * sigma)*F(dx*i, (k)*dt) + sigma*F(dx*i, (k - 1)*dt)));
			//
		}
		a[1] = 0;
		z[N - 2] = 0;

		w[1] = z[1] / b[1];
		q[1] = -s[1] / b[1];
		for (int i = 2; i < N - 2; i++)
		{
			w[i] = z[i] / (b[i] - a[i] *w[i - 1]);
			q[i] = (a[i] * q[i - 1] - s[i]) / (b[i] - a[i] * w[i - 1]);
		}

		y[N - 2][k + 1] = (a[N - 2] * q[N - 3] - s[N - 2]) / (b[N - 2] - a[N - 2] * w[N - 3]);

		for (int i = N - 2; i >= 1; i--)
			y[i][k + 1] = w[i] * y[i + 1][k + 1] + q[i];
		y[0][k + 1] = u1((k + 1)*dt);
		y[N-1][k + 1] = u2((k + 1)*dt);
		k++;
	}

	ofstream fout("out.txt");
	double norm = 0, tmp,t,maxy=0;
	for (int i = 0; i < N; i++)
		for (int j = 0; j < M; j++){
			if (abs(y[i][j])>maxy) maxy = abs(y[i][j]);
			tmp = abs(y[i][j] - ansv(dx*i, dt*j));
			if (tmp > norm) norm = tmp;
		}
		fout << "|| y || = " << maxy << " || u - y || = " << norm << endl;
		for (int i = 0; i < N; i++){
			fout << " yk[" << dx*i << "] = " << y[i][M - 1] <<
				"  \t(yk - u)[" << dx*i << "] = " << y[i][M - 1] - ansv(dx*i,T) << endl;
		}
	system("Pause");
	return 0;
}
```

Результати роботи

```
(N=100, M=1000)
|| y || = 1.00021 || u - y || = 0.063845
 yk[0] = -1                         (yk - u)[0] = 0
 yk[0.010101] = -0.992658           (yk - u)[0.010101] = 0.00673647
 yk[0.020202] = -0.985954           (yk - u)[0.020202] = 0.0116257
       .
       .
       .
 yk[0.141414] = -0.859405           (yk - u)[0.141414] = 0.0254652
 yk[0.151515] = -0.842628           (yk - u)[0.151515] = 0.0258026
 yk[0.161616] = -0.82412            (yk - u)[0.161616] = 0.026905
 yk[0.171717] = -0.807019           (yk - u)[0.171717] = 0.0256649
 yk[0.181818] = -0.791144           (yk - u)[0.181818] = 0.0222993
 yk[0.191919] = -0.772128           (yk - u)[0.191919] = 0.0212097
 yk[0.20202] = -0.752192            (yk - u)[0.20202] = 0.020213
       .
       .
       .
 yk[0.313131] = -0.485052           (yk - u)[0.313131] = 0.014556
 yk[0.323232] = -0.458157           (yk - u)[0.323232] = 0.0139847
 yk[0.333333] = -0.431719           (yk - u)[0.333333] = 0.0127258
 yk[0.343434] = -0.406885           (yk - u)[0.343434] = 0.00968275
       .
       .
       .
 yk[0.484848] = -0.0465348          (yk - u)[0.484848] = -0.0101384
 yk[0.494949] = -0.0235501          (yk - u)[0.494949] = -0.0115709
 yk[0.505051] = 0.000704176         (yk - u)[0.505051] = -0.0111148
 yk[0.515152] = 0.0236968           (yk - u)[0.515152] = -0.0112577
 yk[0.525253] = 0.044065            (yk - u)[0.525253] = -0.0133206
       .
       .
       .
 yk[0.707071] = 0.269525            (yk - u)[0.707071] = -0.0333112
 yk[0.717172] = 0.272402            (yk - u)[0.717172] = -0.0338355
 yk[0.727273] = 0.275545            (yk - u)[0.727273] = -0.0329433
 yk[0.737374] = 0.277286            (yk - u)[0.737374] = -0.032304
 yk[0.747475] = 0.275952            (yk - u)[0.747475] = -0.0335962
       .
       .
       .
 yk[0.888889] = 0.166148            (yk - u)[0.888889] = -0.0310712
 yk[0.89899] = 0.152323             (yk - u)[0.89899] = -0.0299166
 yk[0.909091] = 0.13768             (yk - u)[0.909091] = -0.0288441
 yk[0.919192] = 0.121651            (yk - u)[0.919192] = -0.0284648
       .
       .
       .
yk[0.979798] = 0.0191497            (yk - u)[0.979798] = -0.0207657
yk[0.989899] = 0                    (yk - u)[0.989899] = -0.0200899
 yk[1] = 0                          (yk - u)[1] = 0
```





Висновок:

В результаті проробленої роботи було реалізовано метод симетричних різниць з вагами для гіперболічного рівняння. В ході аналізу результатів методу було виявлено зміни норми похибки при різних параметрах сігма. Автор встановив що з збільшенням кількості вузлів сітки норма похибки зменшується. Для розмірів сітки 100*1000 , була отримана норма похибки || u - y || = 0.063845. Також було показано з допомогою графіка що норма менша ближче до границь сітки, та більша всередині.