Отчет по задаче практикума «Уравнения с частными производными»

Лобзин Фёдор, 407 группа

9 апреля 2021 г.

1 Постановка задачи

Задача. Найти решение задачи (3.13)

$$\begin{cases} \frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2} - x^2 u - 1, \\ \alpha \in \{1.0; 0.1\}, \\ x = 0: \ \frac{\partial u}{\partial x} = 1; \\ x = 1: \ \frac{\partial u}{\partial x} = 0; \\ t = 0: \ u = x\left(1 - x\right)^2. \end{cases}$$

2 Метод решения

Будем использовать метод сеток. Приближенное решение задачи ищем в виде сеточной функции, т.е. функции, определенной в каждом узле сетки (N, M). Эта функция обозначается $\{u_m^n\}$.

Значение u_m^n будем трактовать как приближенное значение функции $u(t,x)$ в узле (t_n, x_m), т.е.

$$u_m^n \sim u(t_n, x_m).$$

Сеточную функцию получим как решение разностного уравнения. Принятый способ разностной аппроксимации называют *схемой*. Для решения нашей задачи будем использовать схему с весами, где вес $\delta \in [0,1]$ будет выбран позднее:

$$\boxed{\frac{u_m^{n+1} - u_m^n}{\tau} = \alpha\left(\delta\frac{u_{m-1}^n - 2u_m^n + u_{m+1}^n}{h^2} + (1-\delta)\frac{u_{m-1}^{n+1} - 2u_m^{n+1} + u_{m+1}^{n+1}}{h^2}\right) + \frac{f_m^n + f_m^{n+1}}{2}}.$$

Идея: последовательно выражать неизвестные сеточные функции из верхнего слоя через известные с нижних слоев с помощью разностного уравнения и граничных условий.

(I) Рассмотрим уравнение из условия

$$\frac{\partial u}{\partial t} = \alpha\frac{\partial^2 u}{\partial x^2} - x^2 u - 1,$$

Запишем для него параметрическое семейство разностных схем с весом $\delta = \frac{1}{2}$:

$$\frac{u_m^{n+1} - u_m^n}{\tau} = \alpha\left(\frac{u_{m-1}^n - 2u_m^n + u_{m+1}^n}{2h^2} + \frac{u_{m-1}^{n+1} - 2u_m^{n+1} + u_{m+1}^{n+1}}{2h^2}\right) - \left(hm^2\right)\frac{u_m^n + u_m^{n+1}}{2} - 1 \qquad (1)$$

(II) Изучим порядок аппроксимации в зависимости от веса δ:
Разложим имеющиеся функции в ряд Тейлора в окрестности точки (t_n, x_m):

$$u_m^{n+1} = u(t_n+\tau, x_m) = u(t_n,x_m) + \tau u_t(t_n,x_m) + \frac{1}{2}\tau^2 u_{tt}(t_n,x_m) + \frac{1}{6}\tau^3 u_{ttt}(t_n,x_m) + o(\tau^3)$$

$$u_{m\pm1}^{n} = u(t_n, x_m \pm h) = u(t_n,x_m) \pm h u_x(t_n,x_m) + \frac{1}{2}h^2 u_{xx}(t_n,x_m) \pm \frac{1}{6}h^3 u_{xxx}(t_n,x_m) + o(h^3)$$

$$u_{m\pm1}^{n+1} = u(t_n, x_m - h) = u(t_n,x_m) - h u_x(t_n,x_m) + \tau u_t(t_n,x_m) + \frac{1}{2}h^2 u_{xx}(t_n,x_m) +$$
$$+ h\tau u_{xt}(t_n,x_m) + \frac{1}{2}\tau^2 u_{tt}(t_n,x_m) + \frac{1}{6}tau^3 u_{ttt}(t_n,x_m) \pm \frac{1}{2}\tau^2 h u_{ttx}(t_n,x_m) +$$
$$+ \frac{1}{2}\tau h^2 u_{txx}(t_n,x_m) \pm \frac{1}{6}h^3 u_{xxx}(t_n,x_m) + o(h^3+\tau^3)$$

Подставим результаты в схему:

$$u_t + o(\tau^2) = \alpha\left(\frac{1}{2}(u_{xx}+o(h^2)) + \frac{1}{2}(u_{xx}+o(h^2))\right) - x^2u - 1;$$

$$u_t = \alpha u_{xx} - x^2 u - 1 + o(h^2) + o(\tau^2);$$

(III) Изучим устойчивость схемы для $\delta = \frac{1}{2}$:
Варьированием (1) по u_m^n получаем линейное уравнение на Δu_m^n:

$$\frac{\Delta u_m^{n+1} - \Delta u_m^n}{\tau} = \alpha\left(\frac{\Delta u_{m-1}^n - 2\Delta u_m^n + \Delta u_{m+1}^n}{2h^2} + \frac{\Delta u_{m-1}^{n+1} - 2\Delta u_m^{n+1} + \Delta u_{m+1}^{n+1}}{2h^2}\right) -$$
$$-(hm)^2\frac{\Delta u_m^n + \Delta u_m^{n+1}}{2}.$$

Замораживая коэффициенты в нем, приходим к линейному уравнению с постоянными коэффициентами, решение всегда имеет вид:

$$\Delta u_m^n = \lambda^n e^{ik\phi} u_0^0 \tag{2}$$

Подставим (3) в (2):

$$\frac{\lambda-1}{\tau} = \alpha\left(\frac{\cos(\phi)-1}{2h^2} + \lambda\frac{\cos(\phi)-1}{2h^2}\right) - x^2;$$

$$\lambda = \frac{1+\tau\alpha\frac{\cos(\phi)-1}{h^2}}{1-\tau\left(\alpha\frac{\cos(\phi)-1}{h^2}\right)} - \frac{\tau x^2}{1-\tau\left(\alpha\frac{\cos(\phi)-1}{h^2}\right)} \Rightarrow$$

По неравенству треугольника:

$$|\lambda| \le \left|\frac{1+\tau\alpha\frac{\cos(\phi)-1}{h^2}}{1-\tau\left(\alpha\frac{\cos(\phi)-1}{h^2}\right)}\right| + \left|\frac{\tau x^2}{1-\tau\left(\alpha\frac{\cos(\phi)-1}{h^2}\right)}\right|.$$

сравним $|1-\beta|$ и $|1+\beta|$, где $\beta \le 0$:

1) $|\beta| \ge 1 \Rightarrow |1-\beta| = 1-\beta, |1+\beta| = -\beta - 1 \Rightarrow |1-\beta| \ge |1+\beta|$
2) $|\beta| \le 1 \Rightarrow |1-\beta| = 1-\beta, |1+\beta| = 1+\beta \Rightarrow |1-\beta| \ge |1+\beta|$

Итого $\forall\beta \le 0$ $|1-\beta| \ge |1+\beta|$,так как $cos(\phi)-1 \le 0$. то $\forall h,\tau : h,\tau \ge 0$ верно:

$$\left|1+\tau\alpha\frac{\cos(\phi)-1}{h^2}\right| \le \left|1-\tau\left(\alpha\frac{\cos(\phi)-1}{h^2}\right)\right| \Rightarrow$$

$$|\lambda| \le 1+\tau\cdot\left|\frac{x^2}{1-\tau\left(\alpha\frac{\cos(\phi)-1}{h^2}\right)}\right| \le 1+\tau\cdot\left|\frac{x^2}{1}\right| \le 1+\mathfrak{o}(\tau)$$

$\Rightarrow$ схема безусловно устойчива.

(IV) Из показанного выше, следует,что предложенная схема безусловно устойчива и имеет порядок аппроксимации $o(\tau^2+h^2)$.

$$\boxed{\frac{u_m^{n+1}-u_m^n}{\tau} = \alpha\left(\frac{u_{m-1}^n-2u_m^n+u_{m+1}^n}{2h^2}+\frac{u_{m-1}^{n+1}-2u_m^{n+1}+u_{m+1}^{n+1}}{2h^2}\right)-(hm)^2\frac{u_m^n+u_m^{n+1}}{2}-1}.$$

Идея: составить систему линейных уравнений с трехдиагональной матрицей на неизвестные $u_{m-1}^{n+1}, u_m^{n+1}, u_{m+1}^{n+1}$ с коэффициентами и свободным членом, вычисленными с нижнего слоя с использованием граничных условий.
Из граничных условий:

1) $x=0:$ аппроксимируем $\frac{\partial u}{\partial x}=1$: $\frac{u_1^{n+1}-u_0^{n+1}}{h}-\frac{h}{2\alpha}\left(\frac{u_0^{n+1}-u_0^n}{\tau}+1\right)=1 \;\Rightarrow\; u_0^{n+1}=\left(-\frac{u_1^{n+1}}{h}+\frac{hu_0^n}{2\alpha\tau}+\frac{h}{2\alpha}-1\right)\cdot\left(-\frac{1}{h}-\frac{h}{2\alpha\tau}\right)^{-1}$
2) $x=1:$ аппроксимируем $\frac{\partial u}{\partial x}=0$: $\frac{u_{M-1}^{n+1}-u_{M-2}^{n+1}}{h}+\frac{h}{2\alpha}\left(\frac{u_{M-1}^{n+1}-u_{M-1}^n}{\tau}+1+h^2(M-1)^2u_{M-1}^{n+1}\right)=0 \;\Rightarrow\; u_{M-1}^{n+1}=\left(\frac{u_{M-2}^{n+1}}{h}+\frac{hu_{M-1}^n}{2\alpha\tau}-\frac{h}{2\alpha}\right)\cdot\left(\frac{1}{h}+\frac{h}{2\alpha\tau}+\frac{h^3(M-1)^2}{2\alpha}\right)^{-1}$
3) $t=0:$
$$x(1+x)^2,\; 0\le x\le 1.$$

Тогда:

$$m=1:\;\; u_1^{n+1}\left(\frac{1}{\tau}-\frac{\alpha}{2h^3\left(\frac{1}{h}+\frac{h}{2\alpha\tau}\right)}+\frac{\alpha}{h^2}+\frac{h^2}{2}\right)-u_2^{n+1}\frac{\alpha}{2h^2}=$$
$$\frac{\alpha}{2h^2}\left(\frac{hu_0^n}{2\alpha\tau}+\frac{h}{2\alpha}-1\right)\cdot\left(-\frac{1}{h}-\frac{h}{2\alpha\tau}\right)^{-1}+\frac{u_1^n}{\tau}+\frac{\alpha}{2}\frac{u_0^n-2u_1^n+u_2^n}{h^2}-(h)^2\frac{u_1^n}{2}-1$$

$$m\in\{2,..,M-3\}:\;\; -u_{m-1}^{n+1}\frac{\alpha}{2h^2}+u_m^{n+1}\left(\frac{1}{\tau}+\frac{\alpha}{h^2}+\frac{hm^2}{2}\right)-u_{m+1}^{n+1}\frac{\alpha}{2h^2}=$$
$$=\frac{u_m^n}{\tau}+\frac{\alpha}{2}\frac{u_{m-1}^n-2u_m^n+u_{m+1}^n}{h^2}-(hm)^2\frac{u_m^n}{2}-1$$

$$m=M-2:\;\; -u_{M-3}^{n+1}\frac{\alpha}{2h^2}+u_{M-2}^{n+1}\left(\frac{1}{\tau}-\frac{\alpha}{2h^3}\cdot\left(\frac{1}{h}+\frac{h}{2\alpha\tau}+\frac{h^3(M-1)^2}{2\alpha}\right)^{-1}+\frac{\alpha}{h^2}+\frac{(h(M-2))^2}{2}\right)=$$
$$\frac{\alpha}{2h^2}\left(\frac{hu_{M-1}^n}{2\alpha\tau}-\frac{h}{2\alpha}\right)\cdot\left(\frac{1}{h}+\frac{h}{2\alpha\tau}+\frac{h^3(M-1)^2}{2\alpha}\right)^{-1}+\frac{u_{M-2}^n}{\tau}+\frac{\alpha}{2}\frac{u_{M-3}^n-2u_{M-2}^n+u_{M-1}^n}{h^2}-(h(M-2))^2\frac{u_{M-2}^n}{2}-1$$

3 Вычислительный эксперимент

Ниже представлено вычисленное значение в точке минимума производной по t при разных n,при $\alpha = 1$

$t \backslash x$	0	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{3}{5}$	$\frac{4}{5}$
0.1	0.00401677	-0.02265996	-0.02859585	-0.03209837	-0.03612519
$t_{MaxD_t}0$	0.01567904	0.12876498	0.14261301	0.09186723	0.02752281
0.9	0.00034823	-0.13726957	-0.23498783	-0.30051979	-0.33680155

Таблица 1: $\tau = \frac{h^2}{3}, m = 50$

$t \backslash x$	0	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{3}{5}$	$\frac{4}{5}$
0.1	0.00201473	-0.02284867	-0.02883320	-0.03217236	-0.03607934
$t_{MaxD_t}0$	0.00802581	0.12851090	0.14341267	0.09400564	0.02975813
0.9	0.00016322	-0.13461593	-0.23209078	-0.29789172	-0.33487399

Таблица 2: $\tau = \frac{h^2}{3}, m = 100$

$t \backslash x$	0	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{3}{5}$	$\frac{4}{5}$
0.1	0.00134435	-0.02291361	-0.02891587	-0.03220261	-0.03606920
$t_{MaxD_t}0$	0.00539224	0.12836924	0.14363221	0.09468678	0.03050546
0.9	0.00010622	-0.13373190	-0.23112757	-0.29701723	-0.33422997

Таблица 3: $\tau = \frac{h^2}{3}, m = 150$

Изучим главный член аппроксимации в опорных точках, ниже представлены значения Δu в них:

$t \backslash x$	0	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{3}{5}$	$\frac{4}{5}$
0.1	0.00200204	0.00018871	0.00023735	0.00007398	-0.00004585
t_{MaxD_t}	0.00765323	0.00025408	-0.00079965	-0.00213841	-0.00223532
0.9	0.00018501	-0.00265364	-0.00289705	-0.00262807	-0.00192756

Таблица 4: $\tau = \frac{h^2}{3}, \Delta u$ при $m = 50 \to m = 100$

$t \backslash x$	0	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{3}{5}$	$\frac{4}{5}$
0.1	0.00067038	0.00006493	0.00008267	0.00003026	-0.00001014
t_{MaxD_t}	0.00263358	0.00014166	-0.00021954	-0.00068114	-0.00074733
0.9	0.00005701	-0.00088402	-0.00096321	-0.00087449	-0.00064402

Таблица 5: $\tau = \frac{h^2}{3}, \Delta u$ при $m = 100 \to m = 150$

Итого, для $\frac{u(m=50)-u(m=100)}{u(m=100)-u(m=150)}$ в опорных точках:

$t \backslash x$	0	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{3}{5}$	$\frac{4}{5}$
0.1	2.98642399	2.90621140	2.87099593	2.44514228	4.52125991
t_{MaxD_t}	2.90602152	1.79361260	3.64233501	3.13945039	2.99106912
0.9	3.24539333	3.00177503	3.00771380	3.00525843	2.99299613

- Аналогично $\alpha = 0.01$.

$t \backslash x$	0	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{3}{5}$	$\frac{4}{5}$
0.1	-0.00011867	0.02040660	0.03475597	-0.01130469	-0.06974567
t_{MaxD_t}	0.00012108	0.12913138	0.14281776	0.09191034	0.02740428
0.9	-0.00048092	-0.55979796	-0.58602151	-0.60504503	-0.61127800

Таблица 6: $\tau = \frac{h^2}{3}, m = 50$

$t \backslash x$	0	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{3}{5}$	$\frac{4}{5}$
0.1	-0.00005656	0.01981100	0.03523544	-0.00946934	-0.06774774
t_{MaxD_t}	0.00009649	0.12860290	0.14346467	0.09401764	0.02973013
0.9	-0.00023619	-0.55902881	-0.58563683	-0.60446091	-0.61147560

Таблица 7: $\tau = \frac{h^2}{3}, m = 100$

$t \backslash x$	0	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{3}{5}$	$\frac{4}{5}$
0.1	-0.00003737	0.01961077	0.03538571	-0.00887018	-0.06707993
$t_{MaxD_t}0$	0.00007232	0.12841018	0.14365544	0.09469229	0.03049326
0.9	-0.00015704	-0.55878089	-0.58551863	-0.60427463	-0.61154632

Таблица 8: $\tau = \frac{h^2}{3}, m = 150$

Изучим главный член аппроксимации в опорных точках, ниже представлены значения Δu в них:

$t \backslash x$	0	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{3}{5}$	$\frac{4}{5}$
0.1	-0.00006211	0.00059560	-0.00047947	-0.00183535	-0.00199792
t_{MaxD_t}	0.00002459	0.00052848	-0.00064690	-0.00210730	-0.00232586
0.9	-0.00024473	-0.00076914	-0.00038468	-0.00058412	0.00019760

Таблица 9: $\tau = \frac{h^2}{3}, \Delta u$ при $m = 50 \to m = 100$

$t \backslash x$	0	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{3}{5}$	$\frac{4}{5}$
0.1	-0.00001919	0.00020024	-0.00015027	-0.00059917	-0.00066781
t_{MaxD_t}	0.00002417	0.00019271	-0.00019077	-0.00067465	-0.00076313
0.9	-0.00007915	-0.00024792	-0.00011820	-0.00018628	0.00007072

Таблица 10: $\tau = \frac{h^2}{3}, \Delta u$ при $m = 100 \to m = 150$

Итого, для $\frac{u(m{=}50)-u(m{=}100)}{u(m{=}100)-u(m{=}150)}$ в опорных точках:

$t \backslash x$	0	$\frac{1}{5}$	$\frac{2}{5}$	$\frac{3}{5}$	$\frac{4}{5}$
0.1	3.23731522	2.97449199	3.19073326	3.06317160	2.99174022
t_{MaxD_t}	1.01727277	2.74229010	3.39094042	3.12353649	3.04780168
0.9	3.09198	3.10237	3.25448	3.13571	2.79412

Нетрудно заметить, что вычислительный эксперимент соответствует теоретическим результатам, а именно:

1) увеличение числа шагов приводит к более точному, (ограниченному) результату, что соответствует устойчивости схемы и граничных условий.

2) изменение функции, при увеличении числа шагов, уменьшается пропорционально квадрату длины малых отрезков, на которые разбивается изначальный, что соотносится с порядком аппроксимации, полученным теоретически.

4 Листинг программы

```
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
void RESHI(double *a, double *x, double *b,int N);
int Max(double *p, int nn, int M);
int Min(double *p,int w, int nn, int M);
void makeAandB(double *p,double *A, double *B, double t,
double h,double a, double b, int nn, int M);
void makeDt(double *Dt,double *p,double t, int nn, int M);


double
lambda (double x)
{
    if(x<=1){
        return x * (1 - x) * (1 - x);}
    else return 0;
}


double
diff (double a, double b, double p[], int nn, int M, int T,double strMax[]
```

```
,int numMax,double str01[],double str09[],double s)
{
    int  MAX, MIN, num = 0, minDx, maxDx, minDt, maxDt;
    double x = 0, *Dx, *Dt;
    double t, n1, M1, h;
    n1 = nn;
    M1 = M;
    t = T / (n1 - 1);
    h = 1 / (M1 - 1);
    for (int i = M  ; i <= M * 2 - 1; i++)
    {
        p[i] = lambda ((i-M)* h);
    }




    for (int j = nn - 2; j >= 0; j--)
    {
        double *A,*B,*X;
        B = (double *) malloc ((M - 2) * sizeof (double));
        X = (double *) malloc ((M - 2) * sizeof (double));
        A = (double *) malloc ((M - 2) * 3 * sizeof (double));

        makeAandB (p, A, B, t, h, a, b, nn,M);
        RESHI (A, X, B, M - 2);

        for (int i = 1; i <= M - 2; i++)
        {
            p[i] = X[i - 1];
        }

        p[0] = (-p[1]/h+(h*p[(1)*M])/(2*a*t)+(h)/(2*a)-1)/((-1/h-(h)/(2*a*t)))  ;
        p[M - 1] =(p[M - 2]/h+(h*p[(1)*M+M-1])/(2*a*t)-(h)/(2*a))/(1/h+(h)/(2*a*t)+h/(2*a)
        if(j==(nn-1)/10){
            for(int i=0;i<5;i++)
        str09[i]=p[(i*M)/5];
       // printf("%d////", j);
       // printf("\n");
        }
        if(j==9*(nn-1)/10){
            for(int i=0;i<5;i++)
        str01[i]=p[(i*M)/5];
        //printf("%d////", j);
        }
        for( int i=0;i<M-1;i++)

        {
            if(fabs(p[i]-p[M+i])>=fabs(s)){s=p[i]-p[M+i];numMax=j;
                 //printf("%d////", numMax);
                 //printf("%.8f    ", p[i]-p[M+i]);
                 for(int k=0; k<5;k++){
                     strMax[k]=p[(k*M)/5];
                 }
```

```
            }
        }
        for (int i = 0; i <= M - 1; i++){

            //printf("%.8f   ", p[i+M]);
            p[i+M]=p[i];
        }
        //printf("\n");


    }


    /*for(int i=0; i<=2*M-1; i++)
    {
        if (i%M==0) printf("\n");
        printf("%.8f   ", p[i]);
    }*/
    //printf("\n");
   // printf("\n");
      /*for(int i=0; i<=nn*M-1; i++)
      {
      if (i%M==0) printf("\n");
      printf("%.8f   ", Dx[i]);
      }
      printf("\n");*/
    return 0;
}


int main ()
{
    int nn=10,M=10,T=1,numMax,M1,nn1, numMax1, nn2, numMax2, M2;
    double  s=0,b = 0.1, a = 0.01,strMax[5],str01[5],str09[5],strMax1[5],str011[5],str091[5
    t2,str012[5],str092[5],n3,t3,strMax2[5];
    for(int j=0; j<3;j++){
        M = 50*(j+1);
        nn=M*M*3;
        double *p,*p1,*p2;
        p = (double *)malloc((2*M)* sizeof (double));
        diff(a,b,p,nn,M,T,strMax,numMax,str01,str09,s);
        /*printf ( "0.1 & %.8f & %.8f  & %.8f & %.8f & %.8f",str01[0],str01[1],str01[2],str
        printf("\\\\");
        printf("\n");
        printf ( "$t_{Max D_t} %d$ & %.8f & %.8f  & %.8f & %.8f & %.8f",numMax, strMax[0],
        printf("\\\\");
        printf("\n");
        printf ( "0.9 & %.8f & %.8f  & %.8f & %.8f & %.8f",str09[0],str09[1],str09[2],str0
        printf("\\\\");
        printf("\n");*/

        M1 = 100*(j+1);
        nn1=M1*M1*3;
        p1 = (double *)malloc((2*M1)* sizeof (double));
```

```
diff(a,b,p1,nn1,M1,T,strMax1,numMax1,str011,str091,s);


M2 = 150*(j+1);
nn2=M2*M2*3;
p2 = (double *)malloc((2*M2)* sizeof (double));
  diff(a,b,p2,nn2,M2,T,strMax2,numMax2,str012,str092,s);

printf ( "0.1 & %.8f & %.8f  & %.8f & %.8f & %.8f",str01[0]-str011[0],str01[1]-str0
str01[2]-str011[2],str01[3]-str011[3],str01[4]-str011[4]);
printf("\\\\");
printf("\n");
printf ( "$t_{Max_D_t}$ & %.8f & %.8f  & %.8f & %.8f & %.8f", strMax[0]-strMax1[0]
strMax[2]-strMax1[2],strMax[3]-strMax1[3],strMax[4]-strMax1[4])  ;
printf("\\\\");
printf("\n");
printf ( "0.9 & %.8f & %.8f  & %.8f & %.8f & %.8f",str09[0]-str091[0],str09[1]-str0
str09[2]-str091[2],str09[3]-str091[3],str09[4]-str091[4]);
printf("\\\\");

printf("\n");
printf("\n");
printf("\n");

printf ( "0.1 & %.8f & %.8f  & %.8f & %.8f & %.8f",str011[0]-str012[0],str011[1]-st
str011[3]-str012[3],str011[4]-str012[4]);
printf("\\\\");
printf("\n");
printf ( "$t_{Max_D_t}$ & %.8f & %.8f  & %.8f & %.8f & %.8f", strMax1[0]-strMax2[0]
strMax1[2]-strMax2[2],strMax1[3]-strMax2[3],strMax1[4]-strMax2[4])  ;
printf("\\\\");
printf("\n");
printf ( "0.9 & %.8f & %.8f  & %.8f & %.8f & %.8f",str091[0]-str092[0],str091[1]-st
str091[3]-str092[3],str091[4]-str092[4]);
printf("\\\\");

printf("\n");
printf("\n");
printf("\n");

printf ( "0.1 & %.8f & %.8f  & %.8f & %.8f & %.8f",(str01[0]-str011[0])/(str011[0]-
(str01[1]-str011[1])/(str011[1]-str012[1]),
(str01[2]-str011[2])/(str011[2]-str012[2]),

(str01[3]-str011[3])/(str011[3]-str012[3]),
(str01[4]-str011[4])/(str011[4]-str012[4]));
printf("\\\\");
printf("\n");
printf ( "$t_{Max_D_t}$ & %.8f & %.8f  & %.8f & %.8f & %.8f", (strMax[0]-strMax1[0]
(strMax[1]-strMax1[1])/(strMax1[1]-strMax2[1]),
(strMax[2]-strMax1[2])/(strMax1[2]-strMax2[2]),
(strMax[3]-strMax1[3])/(strMax1[3]-strMax2[3]),
(strMax[4]-strMax1[4])/(strMax1[4]-strMax2[4]));
printf("\\\\");
```

```
        printf("\n");
        printf ( "0.9 & %.8f & %.8f  & %.8f & %.8f & %.8f",(str09[0]-str091[0])/(str091[0]-
        (str09[1]-str091[1])/(str091[1]-str092[1]),
        (str09[2]-str091[2])/(str091[2]-str092[2]),
        (str09[3]-str091[3])/(str091[3]-str092[3]),
        (str09[4]-str091[4])/(str091[4]-str092[4]));
        printf("\\\\");


    }

    return 0;
}




int Max(double *p,int nn, int M)
{
    double s=0.0;
    int i,k;
    for(i=0;i<nn*M;i++)
    {
        if(p[i]>=s && !(p[i]>=100 && p[i]<=100)){s=p[i];k=i;}
    }
    return k;
}
int Min(double *p,int w, int nn, int M)
{
    double s=2.0;
    int i,k;
    for(i=0;i<nn*M;i++)
    {
        if(p[i]<s && i>=w*M){s=p[i];k=i;}
    }
    return k;
}
void RESHI(double *a, double *x, double *b,int N)
{
    double *s,*p,*y;
    s = (double *) malloc (N * sizeof (double));
    p = (double *) malloc (N* sizeof (double));
    y = (double *) malloc (N* sizeof (double));
    y[0]=a[0];
    s[0]=-a[1]/y[0];
    p[0]=b[0]/y[0];
    for(int i=1;i<N-1;i++)
    {
        y[i]=a[3*i]+a[3*i+2]*s[i-1];
        s[i]=-a[3*i+1]/y[i];
        p[i]=(b[i]-a[3*i+2]*p[i-1])/y[i];
    }
    y[N-1]=a[(N-1)*3]+a[(N-1)*3+2]*s[N-2];
    p[N-1]=(b[N-1]-a[(N-1)*3+2]*p[N-2])/y[N-1];
    x[N-1]=p[N-1];
```

```
    for(int i=N-2;i>=0;i--)
    {
        x[i]=s[i]*x[i+1]+p[i];
    }
    return;
}
void makeAandB (double *p, double *A, double *B, double t, double h, double a,
                double b, int nn, int M)
{
    int q1=0;
    A[0] = 1/t+a/(h*h)+(h*h)/2-a/(2*h*h*h*(1/h+h/(2*a*t)));
    A[1] = -a/(2*h*h);
    A[2] = 0;
    A[(M-2)*3-2] = 0;
    A[(M-2)*3-1]=-a/(2*h*h);
    A[(M-2)*3-3]=1/t+a/(h*h)+(h*(M-2)*h*(M-2))/2-a/(2*h*h*h*(1/h+(h)/(2*a*t)+
    (h/(2*a))));
    B[0]=p[(1)*(M)+1]*(1/t-(h*h)/2)+a*(p[(1)*M]-2*p[(1)*M+1]+p[(1)*M+2])/(2*h*h) -1
        +a*((h*p[(1)*M])/(2*a*t)+(h)/(2*a))/(2*h*h*(-1/h-(h)/(2*a*t)));
    B[M-3]=p[(1)*(M)+M-2]*(1/t-(h*(M-2)*(M-2)*h)/2)+a*(p[(1)*M+M-3]-2*p[(1)*M+M-2]+
    p[(1)*M+M-1])/(2*h*h) -1
           +a*((h*p[(1)*M+M-1])/(2*a*t)-(h)/(2*a))/(2*h*h*(1/h+(h)/(2*a*t)+h/(2*a)));
    for (int i = 1; i < M - 3; i++)

    {


        A[3*i+1]=-a/(2*h*h);
        A[3*i]=1/t+a/(h*h)+(h*h*(i+1)*(i+1))/2;

        A[3*i+2]=-a/(2*h*h);
       B[i]=p[(1)*(M)+1+i]*(1/t-(1+(h*(i+1)-1)*(h*(i+1)-1))/2)+
       a*(p[(1)*M+i]-2*p[(1)*M+1+i]+p[(1)*M+2+i])/(2*h*h)-1;
    }
/*for (int i = 0; i <= (M - 2) * 3 - 1; i++)
    {
if (i % (M - 2) == 0)
        {
          printf ("\n");
        }
        if (i % 3 == 0)
        {
          printf ("%.8f    ", B[8]);
          q1 = q1 + 1;
        }
printf ("%.8f    ", A[i]);
}
//printf ("\n");
//printf ("\n");
//printf ("\n");*/
    return;
}
```

```
void makeDt(double *Dt, double *p, double t,int nn, int M)
{
    int i,j;
    for(j=1;j<nn-1;j++)
    {
        for(i=j*M;i<(j+1)*M;i++)
        {
            Dt[i]=(p[i+M]-p[i-M])/2/t;
        }
    }
    return;
}
```