МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИМ. М. В. ЛОМОНОСОВА

Физический Факультет



**Практическое задание № 2 по курсу «Основы математического моделирования»**

**Задача № 67**

Работу выполнил: Коновалов П.Е., 303 группа.

Москва 2019

1. Постановка задачи.

**№67.** Используя метод переменных направлений, решите краевую задачу:



**Указание:** при решении задачи не пользоваться стандартными пакетами программ; в качестве отчета представить собственную программу, реализующую метод прогонки, а также результаты в виде графиков.

1. Схема переменных направлений

Решаем нашу задачу численно.

1) Введём трёхмерную сетку по двум пространственным координатам x, y и по времени t в области

2) Введём сеточные функции:

- число узлов вдоль осей **x** и **y** соответственно;

- шаги по осям **x** и **y**;

τ – шаг по оси **t**.

3) Введём промежуточный слой по времени:

Составим разностную схему нашей задачи:

Уравнение для сеточной функции можно взять в виде:

где

Таким образом, мы получаем разностную схему нашей задачи:

Однако, если использовать явную схемуто объём вычислительной работы будет равен а устойчивость условная.

Если использовать неявную схему то объём вычислительной работы будет равен а устойчивость безусловная.

Будем применять схему переменных направлений (экономичную разностную схему), поскольку она сочетает достоинства явных и неявных схем, а именно:

1) безусловная устойчивость;

2) объём вычислительной работы

В этой схеме переход со слоя на слой осуществляется в 2 шага, с использованием промежуточного слоя.

Итак, схема переменных направлений для нашей задачи имеет вид:

Уравнение (3) является явным по направлению **y** и неявным по направлению **x**. Уравнение (4) явное по направлению **x** и неявное по направлению **y**. Поэтому каждое уравнение (3), (4) в отдельности может быть решено методом прогонки, суть которого будет изложена чуть ниже. В этом случае счет по схеме переменных направлений требует числа арифметических операций, пропорционального числу узлов сетки, т.е.O(Nx, Ny, Nt), и на каждый узел сетки приходится число операций, не зависящее от числа узлов. Схема имеет точность аппроксимации .

1. Спектральный критерий устойчивости схемы

Проверим, выполняется ли необходимое условие Неймана для нашей разностной схемы. Сделаем переобозначение: **i** → **m**, **j** → **n**. Тогда получаем:

Будем искать решение в виде:

Тогда подставляя в уравнения (1) и (2), получаем следующее:

Сократив оба уравнения на и второе ещё на получаем:

Поскольку то

Аналогично:

Тогда:

Таким образом, необходимое условие устойчивости выполнено. Наша схема безусловно устойчива.

1. Порядок аппроксимации схемы

Определим порядок аппроксимации нашей разностной схемы:

Для этого вычтем из уравнения (1) уравнение (2):

Вводя обозначение:

Получаем:

Данное равенство связывает значение функций на промежуточном слое со значениями сеточной функции на слоях k и (k+1). Теперь подставим (3) в (1). Получаем следующее:

Если 1)

2) I – единичный оператор

3) То:

И уравнение (4) можно переписать следующим образом:

Или

Итак, мы исключили полуцелый слой (\*только f аппроксимируется на промежуточном слое по времени).

Пусть u – точное решение нашей задачи; z – численное решение нашей задачи;

Рассмотрим сеточную функцию w = z – u. В начальный момент времени и на границе области погрешность w равна 0, поскольку начальные и граничные условия аппроксимируются точно. Так как:

И то:

Во внутренних узлах w удовлетворяет разностному уравнению:

где погрешность аппроксимации

Так как:

То:

Собирая всё вместе, имеем:

Итак, схема переменных направлений имеет погрешность аппроксимации и в силу линейности и безусловной устойчивости она сходится и имеет второй порядок точности по координате и времени.

Поскольку для реализации этой схемы требуется число действий, пропорциональное числу узлов сетки, то схема является экономичной.

1. Решение задачи с помощью схемы переменных направлений

Рассмотрим алгоритм схемы переменных направлений. В этой схеме:

переход со слоя k к слою k+1 совершается в 2 этапа с шагами Вначале решается уравнение (1), а затем уравнение (2). Рассмотрим подробнее переход со слоя k на слой k+1.

Используем уравнение (1):

Оно неявное по направлению x и явное по направлению y. Преобразуем его и получим:

Умножим это уравнение на (-1) и сделаем следующие обозначения:

Тогда получаем:

Поскольку в этой задаче:

То это означает, что выполнены достаточные условия использования метода прогонки, и тогда система (3) решается этим методом при каждом при граничных условиях (4). Решение при и находится из граничных условий. Прогонка осуществляется вдоль строк. В результате получаем значение на слое

Теперь используем уравнение (2):

Оно явное по направлению x и неявное по направлению y. Преобразуем его и получим:

Умножим это уравнение на (-1) и сделаем следующие обозначения:

Тогда получаем:

Поскольку в этой задаче:

То это означает, что выполнены достаточные условия использования метода прогонки, и тогда система (5) решается этим методом при каждом при граничных условиях (6). Решение при и находится из граничных условий. Прогонка осуществляется вдоль столбцов. В результате получаем значение на слое

Получив значение функции на слое k+1, находим её значение на слое k+2 и так далее. Таким образом, используя схему переменных направлений и метод прогонки, мы можем численно решить нашу задачу для уравнения теплопроводности.

1. Метод прогонки

Рассмотрим теперь метод, который мы используем для численного решения нашей задачи, метод прогонки.

Данный метод применяется к решению системы алгебраических уравнений:

Достаточные условия применения метода прогонки для решения этой системы имеют вид:

Или

Число арифметических операций прогонки O(N).

Получим теперь формулы прямой и обратной прогонки. Для решения нашей системы уравнений будем считать, что значение искомой функции в двух любых соседних точках связаны следующим соотношением:

где прогоночные коэффициенты

Подставляя (2) в (3), а затем (2) и (3) в (1), получаем

Чтобы соотношение было верно для любых необходимо, чтобы

Соотношения (4) и (5) называются формулами прямой прогонки.

Из и при n = 0 находим:

Используя эти значения, находим остальные коэффициенты из формул (4) и (5).

Из и при n = N-1 находим:

Формулы (6) и называются формулами обратной прогонки. Используя их, определяем значения .

Чтобы коэффициенты и значение были определены, достаточно, чтобы коэффициенты из уравнения (1) удовлетворяли условиям или Только в этом случае можно использовать метод прогонки и применять формулы прямой и обратной прогонки.

Из граничных условий:

Прямой ход прогонки: вычисление коэффициентов 𝛼𝑛+1 , 𝛽𝑛+1 поизвестным 𝛼1, 𝛽1

Из вторых граничных условий:

найдем 𝑦𝑁:

Нейман

Дирихле

1. Код программы:

% задача №2 ОММ вариант 4

%схема переменных направлений

% ut = (uxx + uyy)+yt^2

% x = [0; 1], y = [0;2]

% t = [0, 1]

%граничные условия

%ux(0)=u(1)=0

%uy(0)=uy(2)=0

%начальное условие u(t=0) = (x^2-1)\*cos(pi\*y)

a=1;

global hx;

global hy;

global ht;

%Число шагов по x;

Nx = 100;

%Число шагов по y;

Ny = 100;

%Число шагов по t;

Nt = 100;

%Отрезок по x;

x0=0;

x1=1;

%Отрезок по y;

y0=0;

y1=2;

%Отрезок по t;

t0=0;

t1=1;

%Шаг h по x,а q по t;

hx=(x1-x0)/(Nx);

hy=(y1-y0)/(Ny);

ht=(t1-t0)/(Nt);

x\_grid = linspace(x0, x1, Nx); %для построения сетки

y\_grid = linspace(y0, y1, Ny);

t\_grid = linspace(t0, t1, Nt);

x\_mesh = zeros(Ny, Nx);

for p = 1:Ny

x\_mesh(p, :) = x\_grid;

end

y\_mesh = zeros(Ny, Nx);

for p = 1:Nx

y\_mesh(:, p) = y\_grid'; % ' - транспонирование

end

%сетки для вывода u(y, t) при x = pi/4 = const

y\_mesh\_x = zeros(Ny, Nt);

for p = 1:Nt

y\_mesh\_x(:, p) = y\_grid';

end

t\_mesh\_x = zeros(Ny, Nt);

for p = 1:Ny

t\_mesh\_x(p, :) = t\_grid;

end

u\_x = zeros(Ny, Nt);

u\_an=zeros(Ny,Nx);

for m=0:5

for n=0:5

lambda=(pi\*m/2)^2+(pi\*(n+0.5))^2;

fi\_n\_2=(-32\*(-1)^n)/(2\*pi\*n+pi)^3;

f1\_n\_m=(-32\*(-1)^n)/(pi^3\*(2\*n+1)\*(m)^2);

f2\_n\_m=(lambda^2\*t1^2-2\*lambda\*t1-2\*exp(-lambda\*t1)+2)/lambda^3;

f\_n\_m=f1\_n\_m\*f2\_n\_m;

f\_n\_0=4\*(-1)^n/(pi\*(2\*n+1));

u\_m=cos(pi\*m\*y\_mesh/2);

u\_n=cos(pi\*(n+0.5)\*x\_mesh);

if (mod(m,2)==1)

u\_an=u\_an+f\_n\_m.\*u\_m.\*u\_n;

end

if (m==2)

u\_an=u\_an+fi\_n\_2.\*u\_n.\*cos(pi.\*y\_mesh).\*exp(-lambda.\*t1);

end

if (m==0)

u\_an=u\_an+f\_n\_0.\*f2\_n\_m.\*u\_m.\*u\_n;

end

end

m

end

f2 = figure(2);

surf(x\_mesh, y\_mesh, u\_an, 'LineStyle', 'none');

title('Аналитическое решение');

xlabel('x');

ylabel('y');

zlabel('u');

colorbar;

pointx=zeros(Nx,Ny); %матрица для численного решения

for i=1:1:Nx

for j=1:1:Ny

pointx(i,j)=((i\*hx)^2-1)\*cos(pi\*hy\*j);

end

end

for k=0:1:Nt-1

%сохраняем решение

u\_x(:, k+1) = pointx(Nx/2, :)';

f1 = figure(1);

mesh(x\_mesh, y\_mesh, pointx');

xlabel('x');

ylabel('y');

zlabel('u');

ax=ht/(2\*hx^2);

bx=ht/(2\*hx^2);

cx=1+ht/(hx^2);

%граничные условия для x

%for j=1:1:Ny

%pointx(1,j)=0;

%pointx(Nx,j)=0;

%end

%%%%%%%%%%%%%%%%%%%%%%%

% Прямая прогонка %

%%%%%%%%%%%%%%%%%%%%%%%

%Формируем правый столбец

for i=2:1:Nx-1

for j=2:1:Ny-1

fx(i,j)= pointx(i,j)+ht\*(pointx(i,j+1)-2\*pointx(i,j)+pointx(i,j-1))/(2\*hy^2)\*a+0.5\*ht\*fun\_konovalov(i,j,k);

end

end

%Формируем левую часть

for j=2:1:Ny-1

alphax(2,j)=1;

betax(2,j)=0;

for i=2:1:Nx-1

alphax(i+1,j)=bx/(cx-alphax(i,j)\*ax);

betax(i+1,j)=(ax\*betax(i,j)+fx(i,j))/(cx-alphax(i,j)\*ax);

end

end

%%%%%%%%%%%%%%%%%%%%%%%

% Обратная прогонка %

%%%%%%%%%%%%%%%%%%%%%%%

for j=2:1:Ny-1

pointx(Nx,j)=0; % граничное условие дирихле

for i=Nx-1:-1:1

pointx(i,j)=alphax(i+1,j)\*pointx(i+1,j)+betax(i+1,j);

end

end

%-----------------------------------------------------------

ay=ht/(2\*hy^2);

by=ht/(2\*hy^2);

cy=1+ht/(hy^2);

%%%%%%%%%%%%%%%%%%%%%%%

% Прямая прогонка %

%%%%%%%%%%%%%%%%%%%%%%%

%Формируем правый столбец

for j=2:1:Ny-1

for i=2:1:Nx-1

fy(i,j)=pointx(i,j)+ht\*(pointx(i+1,j)-2\*pointx(i,j)+pointx(i-1,j))/(2\*hx^2)\*a+0.5\*ht\*fun\_konovalov(i,j,k);

end

end

%Формируем левую часть

for i=2:1:Nx-1

alphay(i,2)=1;

betay(i,2)=0;

for j=2:1:Ny-1

alphay(i,j+1)=by/(cy-alphay(i,j)\*ay);

betay(i,j+1)=(ay\*betay(i,j)+fy(i,j))/(cy-alphay(i,j)\*ay);

end

end

%%%%%%%%%%%%%%%%%%%%%%%

% Обратная прогонка %

%%%%%%%%%%%%%%%%%%%%%%%

for i=2:1:Nx-1

pointx(i,Ny)=betay(i,Ny)/(1-alphay(i,Ny));%betay(i,Ny)/(1-alphay(i,Ny));

for j=Ny-1:-1:1

pointx(i,j)=alphay(i,j+1)\*pointx(i,j+1)+betay(i,j+1);

end

end

%-----------------------------------------------------------

%Граничные условия для y

for i=1:1:Nx-1 %из граничных условий неймана

pointx(i,1)=pointx(i,2);

pointx(i,Ny)=pointx(i,Ny-1);

end

end

f5 = figure(5);

surf(x\_mesh, y\_mesh, abs((pointx' - u\_an)), 'LineStyle', 'none');

title('Ошибка');

xlabel('x');

ylabel('y');

zlabel('u');

colorbar;

f6 = figure(6);

surf(t\_mesh\_x, y\_mesh\_x, u\_x, 'LineStyle', 'none');

title('Зависимость u(y,t) при x = 1/2');

xlabel('t');

ylabel('y');

zlabel('u');

colorbar;

% axis([0, tN, 0, 2, -1, 1]);

Файл функция:

function [y] = fun\_konovalov(i,j,k)

global hx;

global hy;

global ht;

y=(j\*hy)\*(ht\*(k+(1/2)))^2;

end

1. Результаты работы:

|  |  |
| --- | --- |
| Nx | 200 |
| Ny | 200 |
| Nt | 200 |
| t | 1 |



