МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИМ. М. В. ЛОМОНОСОВА

Физический Факультет



Практическое задание по курсу «Основы математического моделирования»:

**Решение квазилинейного уравнения переноса с использованием схемы бегущего счета и итерационных методов**

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

Москва 2019

1. Постановка задачи.

Используя схему бегущего счета и итерационные методы решить задачу:

1. Исследуем существование однозначного решения в области определения :

Для этого составим уравнение характеристик и узнаем, будут ли пересекаться их проекции. Если проекции характеристик будут пересекаться, то может возникнуть физическая неоднозначность, и решение может быть неоднозначно (может возникнуть физическая неоднозначность, которой нет в реальном физическом процессе)

а) Характеристики

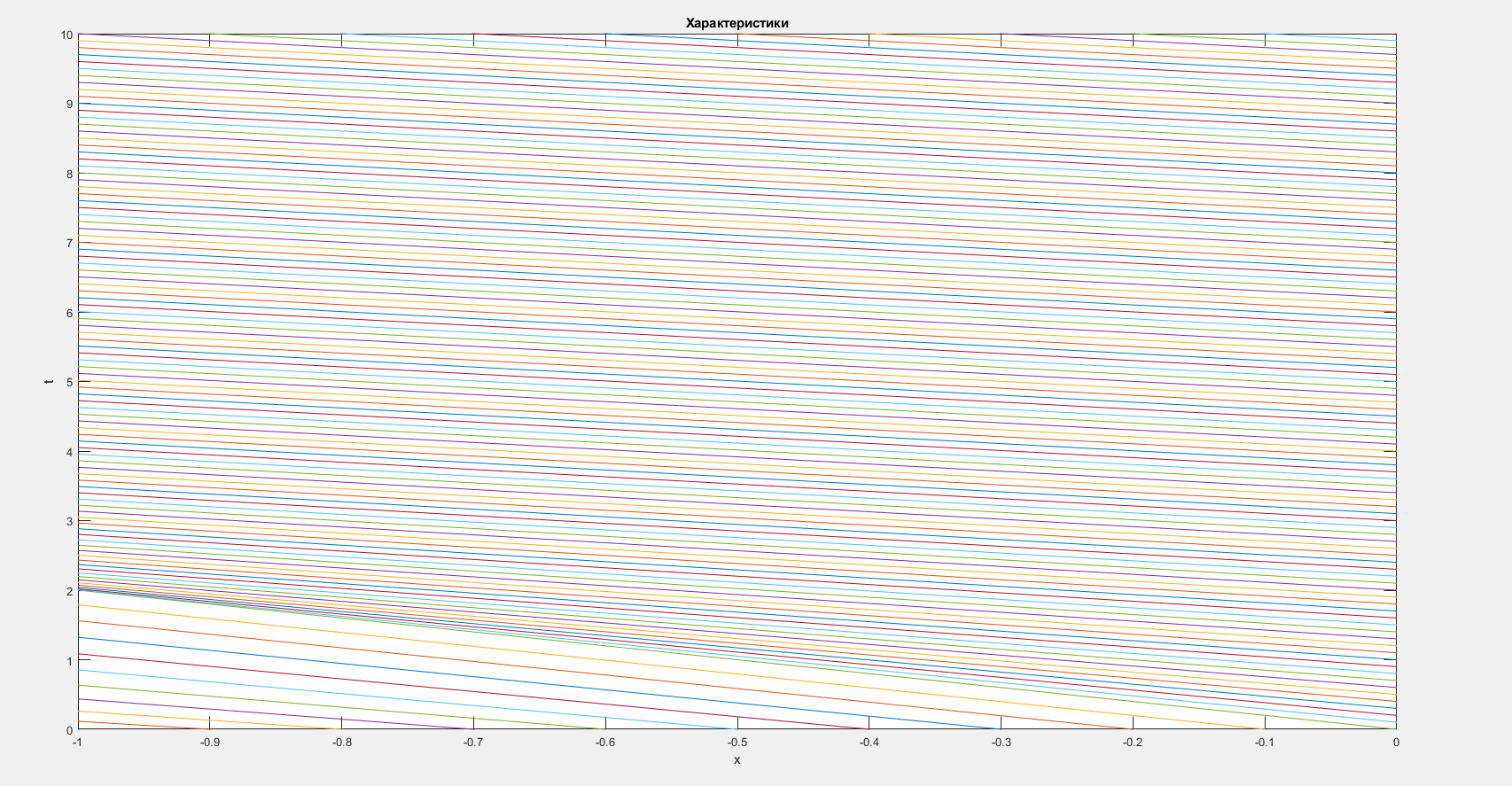
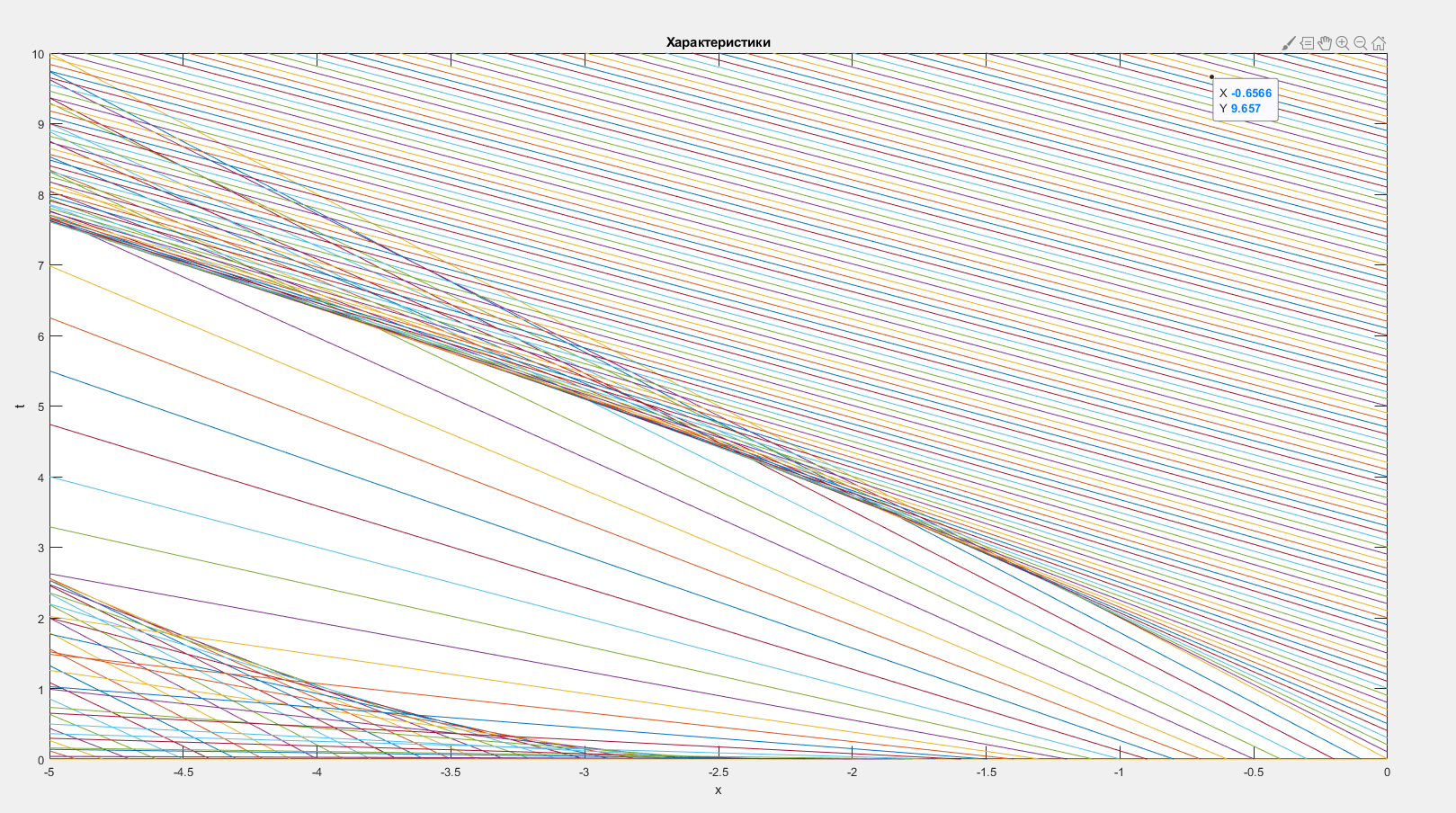
Далее:

Интегрируя (4) получим:

*=>*

*=>*

Далее построим для различных x0 и y0  характеристики:



*Рис. Показ точек пересечения характеристик за пределами рассмотрения задачи*

Как видно, на интервале характеристики не пересекаются, следовательно, нет опрокидывания волны (нет «выгибания волны»). Решение на нашем интервале определено однозначно.

1. Перейдем к построению разностной схемы.

Приведем к более удобному виду:

Тогда:

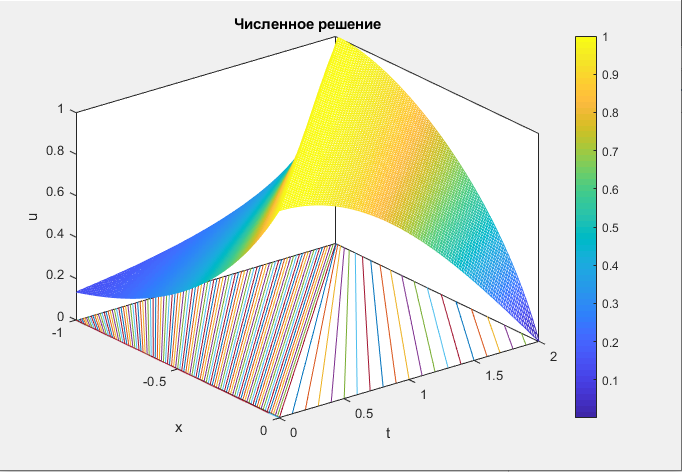
Далее:

Введем разностную сетку **w** **= {xi = h\*i , i = от 0 до N, h = -1/N , tj = τ\*j}** , где N – число узлов вдоль оси x, τ – шаг по времени. Введем также сеточные функции  **= u(xi , tj)** и **= *F*( ).** Разностная схема поставленной задачи (1 или 8) в точке (xi + h/2, tj + τ/2) будет выглядеть следующим образом:

(9)

1. Полученные результаты:

|  |  |
| --- | --- |
| **Величина** | **Значения в работе** |
| **x** | [-1;0) |
| **t** | [0;2] |
| **Nx** | 100 |
| **Nt** | 100 |
| **τ** | 0,02 |
| **|h|** | 1/100 |
|  | 0.01 |



1. Код программы:

Файл Main.m:

%Зададим сетку по x и t и ее параметры

%Не забываем, что f-разностная схема, F-функция в pdf файле

%Так как задача «однородная», то f=0

%Зададим пределы изменения коордтинаты x

x0=0;

x1=-1;

%Зададим пределы изменения коордтинаты t

t0=0;

t1=2;

%Определим узлы сетки по x и t (размерность матрицы)

Nx = 100;

Nt = 100;

point=zeros(Nt,Nx);

%Зададим шаг сетки по x и t

global h tau

h=(x1-x0)/Nx;

tau=(t1-t0)/Nt;

%Заданная точность

global e

e=0.01;

%Начальный момент времени на границе

%point(1,1)=internal\_t(0);

%point(координата{ось Y},время{ось X})

%Распределение начальных значений по t

for i=0:1:Nx

point(i+1,1)=internal\_t(i\*h);

end

%Распределение начальных значений по x

for j=0:1:Nt

point(1,j+1)=internal\_x(j\*tau);

end

%Схема бегущего счета:

%u12 u11 u21 - это узлы, которые мы знаем

%u1 - это приближение y(i+1)(j+1)

%u2 - это сумма первого приближения u1 плюс delta

%delta - это обычно будет delta = -f(u1)/f'(u1), в pdf F(u)=-arctg(exp(u))

%f=y../2h + F../2tau-из разностной схемы, это как бы вся разностная схема

%а f'=1/(2\*tau) + F'/2h, это для разложения функции в ряд:

% 0 = f + f' \* delta

for j=1:1:Nt

for i=1:1:Nx

delta=1;

u1=point(i+1,j+1);

u12=point(i,j+1);

u11=point(i,j);

u21=point(i+1,j);

%Итерационный метод Ньютона:

%delta - это обычно будет delta = -f(u1)/f'(u1)

%derivFunF(u1) = f'=1/(2\*tau) + F'/2h

%F'=-exp(u)/1+exp(2u)

%FunF1(u12, u1, u11, u21)= f=y../2h + F../2tau

while (delta>e)

u2=u1-FunF1(u12, u1, u11, u21)/derivFunF(u1);

delta=abs(u2-u1);

u1=u2;

end

point(i+1,j+1)=u2;

end

end

%Параметры графика

x\_grid = linspace(x0, x1, Nx+1); %для построения сетки

y\_grid = linspace(t0, t1, Nt+1);

% t\_grid = linspace(t0, tN, Nt);

x\_mesh = zeros(Nt+1, Nx+1);

for p = 1:Nt+1

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

end

y\_mesh = zeros(Nt+1, Nx+1);

for p = 1:Nx+1

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

end

%Параметры графика

mesh(x\_mesh, y\_mesh, point);

title('Численное решение');

xlabel('x');

ylabel('t');

zlabel('u');

%axis(gca, 'vis3d');

axis([-1 0 t0 t1]);

rotate3d;

colorbar;

%ПЕРЕД ЗАПУСКОМ ПРОГНАТЬ internal\_t

Отдельные файл-функции:

%Коэффициент это у нас F'

function y = coeff(x)

y=-1/(1+x);

end

%Производная f' для итерационного метода Ньютона

function f = derivFunF(u)

global tau h

f=1/(2\*tau) + coeff(u)/(2\*h);

end

%Зададим функцию F(u)в расчете

function y = funf(x)

y=-log(abs(1+x));

end

%Задаем f для итерационного метода Ньютона FunF1 (f) через funf (F)

function f = FunF1( a1, a2, b1, b2 )

global tau h

f=(a1-b1+a2-b2)/(2\*tau) + (funf(b2)-funf(b1)+funf(a2)-funf(a1))/(2\*h);

end

%Начальные условия по t

function y = internal\_t(x)

y=cos(pi\*x/2);

end

%Граничные условия по x

function y = internal\_x(t)

y=exp(-t);

end

Построение характеристик

xm=linspace(-5, 0, 100);

for xx=-5:0.1:0

tm=(xx-xm)\*(1+cos(pi\*xx/2));

plot(xm, tm);

title('Характеристики');

hold on

xlabel('x');

ylabel('t');

axis([-5 0 0 2]);

end

for tt=0:0.1:20

tm=tt-xm\*(1+exp(-tt));

plot(xm, tm);

hold on

xlabel('x');

ylabel('t');

axis([-5 0 0 10]);

end

1. Теоретическое приложение:

*Устойчивость разностной схемы:*

Для исследования устойчивости нашей разностной схемы проверим выполнение критерия Неймана (необходимое условие устойчивости), критерий Куранта (достаточное условие устойчивости) и геометрический критерий устойчивости.

***Критерий Неймана***

Применим метод «замороженных» коэффициентов и «заморозим» коэффициент перед Возьмём произвольную точку рассматриваемой области.

Введём обозначение:

Тогда наша разностная схема будет иметь следующий вид:

Чтобы не путать мнимую единицу с индексом i сделаем следующее переобозначение в этом параграфе: i → n, j → m.

Получаем:

Будем искать решение уравнения в виде: , где i – мнимая единица. Тогда подставляя это решение в уравнение, получаем:

Поделим на и получим:

Выражая из этого уравнения находим:

Тогда:

Таким образом, получаем, что необходимое условие устойчивости выполняется при любом соотношении шагов по координате и времени.

***Критерий Куранта***

Проверим выполнение достаточного условия устойчивости для нашей разностной схемы. Для этого поставим для неё следующую задачу:

Преобразуем это уравнение и получим:

или

- возмущение исходной задачи

Учитывая, что получаем

Тогда:

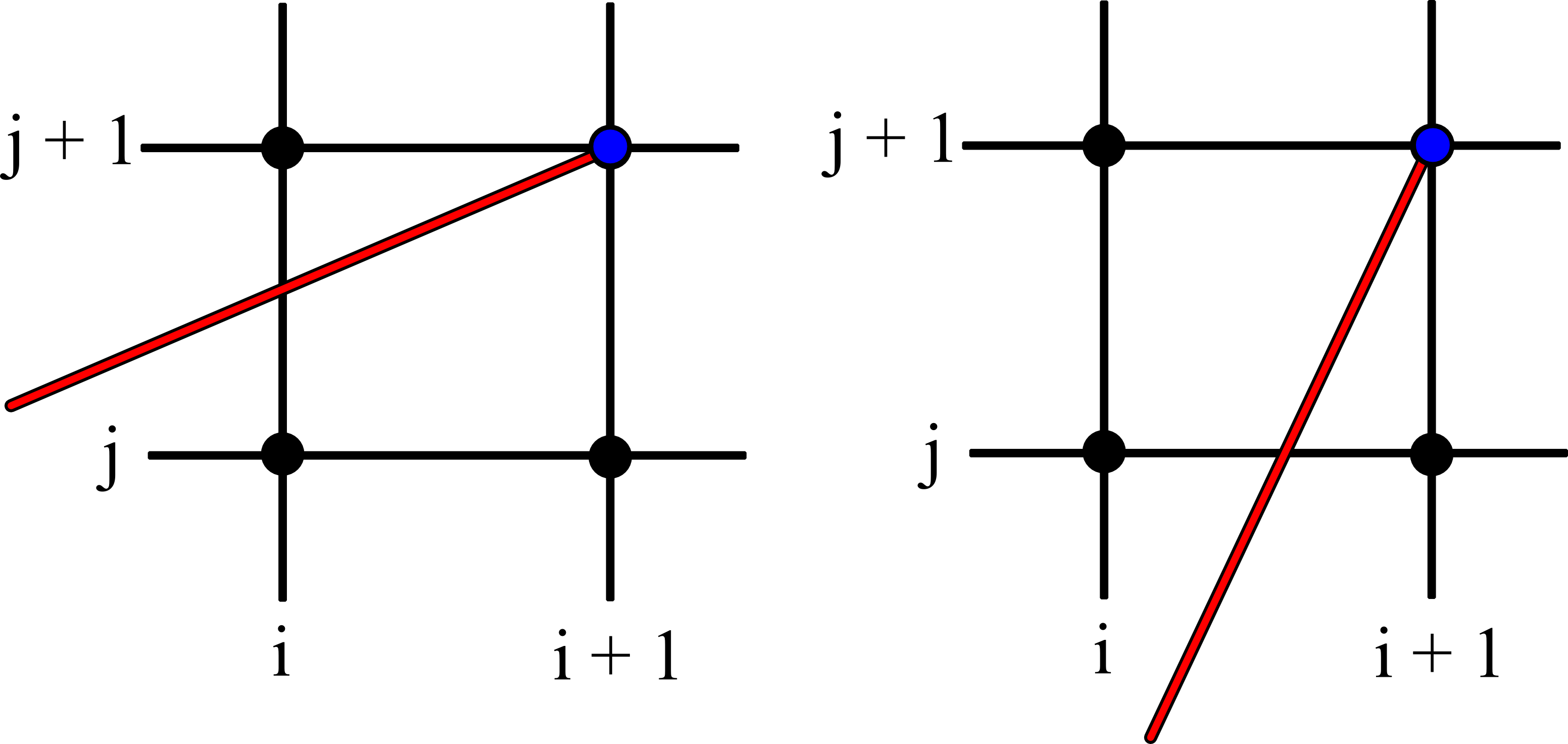
Поскольку

то получаем, что

где T – величина интервала времени, на котором мы ищем решение. Получаем, что достаточное условие устойчивости выполнено для нашей схемы для любых соотношений шагов по времени и координате. Это и означает, что разностная схема, которой мы пользуемся, является безусловно устойчивой.

***Геометрический критерий устойчивости***

При решении задачи мы используем следующий шаблон:



В точке ищется значение исследуемой функции. Пусть через эту точку проходит одна из характеристик. Поскольку в остальных точках шаблона значение исследуемой функции известно, то характеристика пересечёт отрезок, соединяющий точки шаблона, в которых значение исследуемой функции известно, при любых соотношениях шагов по времени и координате. Это в свою очередь и означает устойчивость схемы.

Итак, мы показали, что наша разностная схема является безусловно устойчивой.

*Порядок аппроксимации разностной схемы*

Вычислим порядок аппроксимации нашей задачи разностной схемой в точке Обозначим эту точку за А.

Разложим значение функции U в узлах сетки в ряд Тейлора до члена третьего порядка включительно в точке

Наша разностная схема имеет вид:

Для приращения по пространственному индексу имеем:

Для приращения по временному индексу имеем:

Подставляя выражения (11), (12), (13) и (14) в выражение (10), получим следующее:

или

Таким образом, порядок аппроксимации четырёхточечной схемы бегущего счёта равен

*Схема бегущего счёта и итерационные методы*

Полученную разностную задачу будем решать при помощи схемы бегущего счёта. Считаем, что значение на j-м слое известно, и нужно найти решение на (j+1)-м слое при i, пробегающем все допустимые значения, учитывая, что значение в точке (0, j+1) задано.

Используем итерационный метод Ньютона.

Пусть

Считаем, что значение известно, тогда при i = 0 имеем:

Будем считать, что известно некоторое приближение к корню , то тогда получаем, что

Разложим это выражение в ряд и линеаризуем его.

Тогда получим:

Отсюда имеем:

Пусть - некоторое новое приближение

Тогда

Зная , можно найти новое значение , а зная его найти новое, более точное приближение и так далее. Таким образом этот процесс будет повторяться и остановится по достижению заданной точности

Последовательно вычисляя получим значения функции для времени

*Ошибка решения*



Зависимость ошибки от размеров сетки при e=0,01 в точке (-0,5;1)