



МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное образовательное учреждение высшего образования  
«Дальневосточный федеральный университет»  
(ДВФУ)

---

## ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ

Департамент математического и компьютерного  
моделирования

### ОТЧЕТ к лабораторной работе № 5

по дисциплине «Математическое и компьютерное моделирование  
(Mathematical and Computer Modeling)»

Направление подготовки  
**01.03.02 «Прикладная математика и информатика»**

Выполнила студентка  
группы Б9119-01.03.02

Пахомова Д.Е.

(ФИО)

(подпись)

Проверил д.ф.-м.н.

Пермяков М.С.

(ФИО)

(подпись)

« 17 » января 20 22 г.

г. Владивосток  
2022

# Содержание

<b>Введение</b>	<b>3</b>
<b>Задание 1: Движение вещества</b>	<b>3</b>
1.1 Формулировка задачи . . . . .	3
1.2 Постановка физической модели . . . . .	3
1.3 Постановка математической модели процесса . . . . .	4
1.4 Численная реализация методов . . . . .	5
<b>Заключение</b>	<b>8</b>
<b>ПРИЛОЖЕНИЕ</b>	<b>9</b>

# Введение

В данной лабораторной работе требуется решить и оформить задачи для нахождения концентрации вещества в конечный момент времени.

## Задание 1: Движение вещества

### 1.1 Формулировка задачи

Некоторое вещество с заданной концентрацией распределено в пространстве. С течением времени вещество переносится с постоянной скоростью. Найти концентрацию вещества в конечный момент времени.

### 1.2 Постановка физической модели

Для решения задачи используем следующие параметры:

Координата вещества:  $x$ ;

Шаг координаты:  $h$ ;

Индекс координаты:  $i$ ;

Время:  $t$ ;

Шаг времени:  $\tau[c]$ ;

Индекс времени:  $n$ ;

Скорость переноса вещества:  $u$ ;

Концентрация вещества:  $C(x, t)[\text{м}^{-3}]$ .

### 1.3 Постановка математической модели процесса

Для построения математической модели рассмотрим:

1. Формулу изменения концентрации вещества:

$$\frac{\partial C}{\partial t} + u \cdot \frac{\partial C}{\partial x} = 0;$$

2. Метод центральных разностей для концентрации вещества:

$$\frac{\partial C}{\partial x} = \frac{C_{i+1}^n - C_{i-1}^n}{2h};$$

3. Метод «чехарды» для концентрации вещества:

$$\frac{C_i^{n+1} - C_i^{n-1}}{2\tau} + u \cdot \frac{C_{i+1}^n - C_{i-1}^n}{2h} = 0;$$

4. Метод «против потока» для концентрации вещества:

$$\begin{cases} \frac{C_i^{n+1} - C_i^n}{\tau} + u \cdot \frac{C_i^n - C_{i-1}^n}{h} = 0; u > 0, \\ \frac{C_i^{n+1} - C_i^n}{\tau} + u \cdot \frac{C_{i+1}^n - C_i^n}{h} = 0; u < 0. \end{cases}$$

5. Для дальнейшей реализации метода центральных разностей подставим уравнение (2) в уравнение (1):

$$\frac{C_{i+1}^{n+1} - C_i^n}{\tau} + u \cdot \frac{C_{i+1}^n - C_{i-1}^n}{2h} = 0, \text{ откуда:}$$

$$C_{i+1}^{n+1} = C_i^n - \frac{\tau \cdot u}{2h} (C_{i+1}^n - C_{i-1}^n) -$$

нужная нам формула концентрации для вычисления по **методу центральных разностей**;

6. Для дальнейшей реализации **метода «чехарды»** выразим  $C_i^{n+1}$  из уравнение (3):

$$C_i^{m+1} = C_i^{m-1} - \frac{2\tau \cdot u}{2h} \cdot (C_{i+1}^m - C_{i-1}^m);$$

7. Для дальнейшей реализации метода «против потока» выразим  $C_i^{m+1}$  из системы уравнений (4):

$$\begin{cases} C_i^{m+1} = C_i^n - \frac{\tau \cdot u}{h}(C_i^n - C_{i-1}^n); u > 0, \\ C_i^{m+1} = C_i^n - \frac{\tau \cdot u}{h}(C_{i+1}^n - C_{i-1}^n); u < 0. \end{cases}$$

## 1.4 Численная реализация методов

Используем следующие параметры для реализации:

Координата вещества:  $x \in [0, 6]$ ;

Шаг координаты:  $h = 0.1$ ;

Шаг времени:  $\tau = 0.1$ с;

Скорость переноса вещества:  $u = \text{const} = 1$ ;

Концентрация вещества:

$$\begin{cases} C = 500; x \in [0, 1.5], \\ C = 200; x \in [1.5, 3], \\ C = 0; x \in [3, 6]. \end{cases}$$

Время:  $t \in [0, 1]$ с.

Реализацию будем производить на языке Python, для визуализации используем библиотеку matplotlib.

1. Метод центральных разностей

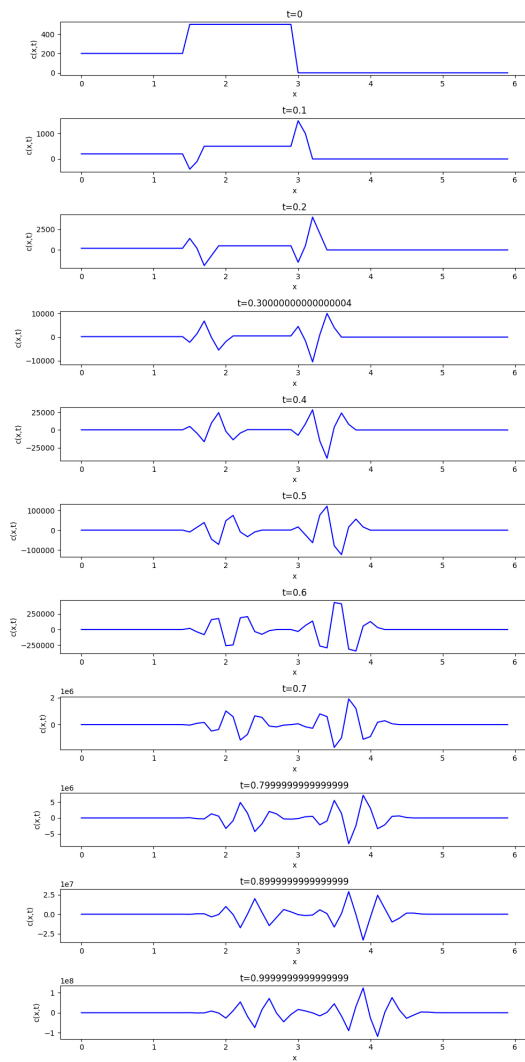


Рис. 1: Метод конечных разностей

## 2. Метод «чехарда»

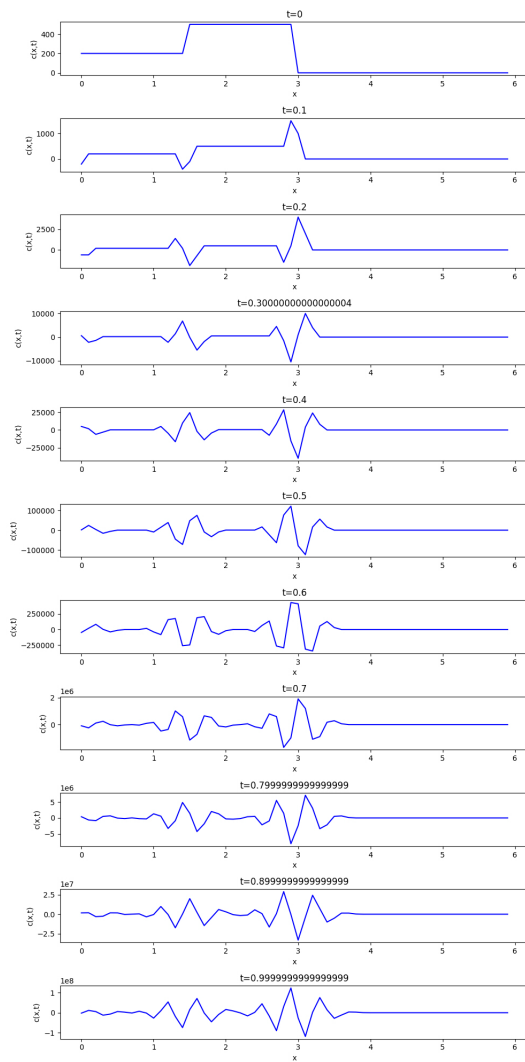


Рис. 2: Метод «чехарда»

### 3. Метод «против потока»

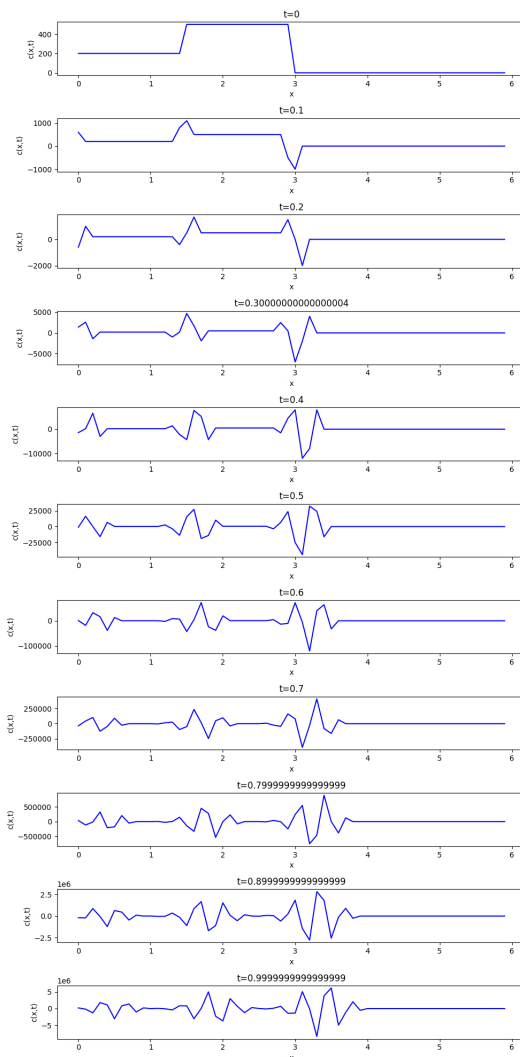


Рис. 3: Метод конечных разностей

## Заключение

В данной лабораторной работе мною были решены и оформлены в среде компьютерной верстки «TeX» поставленные задачи: построены математическая и компьютерная модели одномерного переноса вещества, проанализированы процессы и сам объект.



# ПРИЛОЖЕНИЕ

## 1. Код для метода центральных разностей:

```
import math
import copy
import numpy as np
from scipy.integrate import odeint
from matplotlib import pyplot as plt

h = 0.1
tau = 0.1
u = 4
x = np.arange(0, 6, h)

concentration = [0 for i in range(60)]

def central_difference(tau, h, u, concentration):
    temp = copy.deepcopy(concentration)
    for i in range(2, len(concentration)):
        concentration[i] = temp[i-1] - ((tau * u * (temp[i] - temp[i-2])) / (2 * h))

for i in range(30):
    if i < 15:
        concentration[i] = 200
    else:
        concentration[i] = 500
    print(concentration)

fig, axs = plt.subplots(11)
fig.set_size_inches(10, 20)
t = np.arange(0, 1, tau)
print(len(t))
r = 0
axs[0].set_xlabel("x")
axs[0].set_ylabel("c(x,t)")
axs[0].plot(x, concentration, 'b')
axs[0].set_title('t=%(time)s' % {'time': r})
for i in range(1, len(t) + 1):
    r += tau
    round(r)
    central_difference(tau, h, u, concentration)
    axs[i].set_xlabel("x")
    axs[i].set_ylabel("c(x,t)")
    axs[i].plot(x, concentration, 'b')
    axs[i].set_title('t=%(time)s' % {'time': r})
fig.tight_layout()
plt.show()
```

## 2. Код для метода «чехарда»:

```
import math
import copy
import numpy as np
from scipy.integrate import odeint
from matplotlib import pyplot as plt

h = 0.1
tau = 0.1
u = 4
x = np.arange(0, 6, h)

concentration = [0 for i in range(60)]

def central_difference(tau, h, u, concentration):
    temp = copy.deepcopy(concentration)
    for i in range(len(concentration) - 1):
        concentration[i] = concentration[i] - ((tau * u * (temp[i+1] - temp[i-1])) /
        (2 * h))

for i in range(30):
    if i < 15:
        concentration[i] = 200
    else:
        concentration[i] = 500
    print(concentration)

fig, axs = plt.subplots(11)
fig.set_size_inches(10, 20)
t = np.arange(0, 1, tau)
print(len(t))
r = 0
axs[0].set_xlabel("x")
axs[0].set_ylabel("c(x,t)")
axs[0].plot(x, concentration, 'b')
axs[0].set_title('t=%(time)s' % {'time': r})
for i in range(1, len(t) + 1):
    r += tau
    round(r)
    central_difference(tau, h, u, concentration)
    print(concentration)
    axs[i].set_xlabel("x")
    axs[i].set_ylabel("c(x,t)")
    axs[i].plot(x, concentration, 'b')
    axs[i].set_title('t=%(time)s' % {'time': r})
fig.tight_layout()
plt.show()
```

### 3. Код для метода «против потока»:

```
import math
import copy
import numpy as np
from scipy.integrate import odeint
from matplotlib import pyplot as plt

h = 0.1
tau = 0.1
u = 4
x = np.arange(0, 6, h)

concentration = [0 for i in range(60)]

def central_difference(tau, h, u, concentration):
    temp = copy.deepcopy(concentration)
    for i in range(len(concentration) - 1):
        if u > 0:
            concentration[i] = temp[i] - ((tau * u * (temp[i] - temp[i-1])) / (2 * h))
        elif u < 0:
            concentration[i] = temp[i] - ((tau * u * (temp[i+1] - temp[i - 1])) / (2 * h)
            )

for i in range(30):
    if i < 15:
        concentration[i] = 200
    else:
        concentration[i] = 500
    print(concentration)

fig, axs = plt.subplots(11)
fig.set_size_inches(10, 20)
t = np.arange(0, 1, tau)
print(len(t))
r = 0
axs[0].set_xlabel("x")
axs[0].set_ylabel("c(x,t)")
axs[0].plot(x, concentration, 'b')
axs[0].set_title('t=%(time)s' % {'time': r})
for i in range(1, len(t) + 1):
    r += tau
    round(r)
    if i % 2 == 0:
        central_difference(tau, h, u, concentration)
    else:
        central_difference(tau, h, -u, concentration)
    print(concentration)
    axs[i].set_xlabel("x")
    axs[i].set_ylabel("c(x,t)")
    axs[i].plot(x, concentration, 'b')
    axs[i].set_title('t=%(time)s' % {'time': r})
fig.tight_layout()
plt.show()
```