

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение

высшего образования

**«Дальневосточный федеральный университет»**

**ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ**

**Департамент математического и компьютерного моделирования**

**ОТЧЕТ**

К лабораторной работе №6 по дисциплине

«Математическое и компьютерное моделирование»

Направление подготовки   
01.03.02 «Прикладная математика и информатика»

|  |  |  |
| --- | --- | --- |
|  |  | Выполнил студент гр.  Б9122-01.03.02сп  Носков Я. В. \_\_\_\_\_\_\_\_\_\_\_  (Ф.И.О.) (подпись)  Проверил профессор д.ф.-м.н.  Пермяков М. С. \_\_\_\_\_\_\_\_ |
|  |  | (Ф.И.О.) (подпись)  « » 2025г. |
|  |  |  |

**г. Владивосток**

**2025**

**Оглавление**

[Введение 3](#_Toc201158963)

[1. Построение модели 4](#_Toc201158964)

[1.1 Начальные и граничные условия 4](#_Toc201158965)

[1.2 Кинематика потока 4](#_Toc201158966)

[1.3 Метод частиц 5](#_Toc201158967)

[1.4 Конечно-разностный метод 5](#_Toc201158968)

[2. Анализ модели 6](#_Toc201158969)

[2.1 Не сжимаемость и сохранение массы 6](#_Toc201158970)

[2.2 Сохранение концентрации вдоль траектории 6](#_Toc201158971)

[2.3 Формулировка задачи Коши 7](#_Toc201158972)

[2.4 Устойчивость 7](#_Toc201158973)

[2.5 Единственность и существование 7](#_Toc201158974)

[3. Вычислительные эксперименты 8](#_Toc201158975)

[3.1 Условия эксперимента 8](#_Toc201158976)

[3.2 Сравнение 8](#_Toc201158977)

[3.3 Результат 10](#_Toc201158978)

[Заключение 12](#_Toc201158979)

[Приложения 13](#_Toc201158980)

# Введение

Уравнения переноса занимают важное место в прикладной математике и математическом моделировании, поскольку описывают процессы перемещения вещества, энергии или импульса в различных средах. В классических задачах математической физики одномерное уравнение переноса используется для моделирования движения загрязняющих веществ в реке, распространения тепла или волн в трубопроводах, но в реальных системах часто необходимо учитывать более сложную, двумерную геометрию.

Двумерное уравнение переноса позволяет описывать процессы, происходящие одновременно в двух независимых пространственных направлениях, например, перенос тепла в пластине, движение примесей в атмосфере или морской воде, распространение загрязнений в почве и другие процессы, где направление переноса не ограничивается одной осью. Такие модели особенно актуальны в инженерных задачах, экологии, метеорологии и геофизике.

В отличие от одномерных моделей, двумерные модели требуют более сложных методов анализа и численного решения, так как увеличивается количество переменных, усложняется структура граничных условий и возрастает вычислительная нагрузка. Кроме того, при численном решении особое внимание уделяется устойчивости и точности используемых методов, поскольку даже незначительные ошибки могут привести к физически некорректным результатам.

Целью данной работы является построение и исследование модели двумерного уравнения переноса. Будут рассмотрены математическая постановка задачи, численные методы её решения, а также проведены вычислительные эксперименты, позволяющие проанализировать поведение системы при различных параметрах. Особое внимание уделяется устойчивости схем и влиянию начальных и граничных условий на характер решения.

# Построение модели

Двумерное уравнение переноса описывает процесс переноса физической величины (например, концентрации вещества, температуры или плотности) в двумерной области под действием потоков в направлениях координатных осей. Обобщённая форма уравнения переноса имеет вид:

|  |  |
| --- | --- |
|  | *(1)* |

где:

* – функция, описывающая распределение вещества во времени и пространстве;
* ) – компоненты вектора скорости переноса в направлении осей и ;
* – время.

## Начальные и граничные условия

* Начальное условие:

|  |  |
| --- | --- |
|  | *(2)* |

где определяет форму и положение примеси в начальный момент времени.

* Граничные условия – либо задаются значения на границе (Дирихле), либо используются прозрачные/периодические условия для симуляции замкнутой области.

## Кинематика потока

Для задания поля скоростей используется **функция тока** , с помощью которой компоненты вектора скорости выражаются как:

|  |  |
| --- | --- |
|  | *(3)* |

Такая постановка предполагает, что перенос осуществляется **без диффузии** (чистая конвекция), и что примесь перемещается исключительно за счёт движения среды. Это типичный подход при моделировании быстро текущих, слабо диффузных процессов, таких как движение загрязнителя в жидкости или газа в атмосфере.

Такой подход гарантирует не сжимаемость потока (), где:

* ;

что отражает физическую модель движущейся несжимаемой среды (например, воды).

## Метод частиц

Метод частиц моделирует перенос вещества за счёт **отслеживания движения большого числа отдельных точек (частиц),** которые перемещаются по траекториям потока. Каждая частица «несёт» с собой свою концентрацию, которая со временем **не изменяется**

А движение частиц подчиняется системе ОДУ:

|  |  |
| --- | --- |
|  | *(4)* |

где – текущие координаты частицы.

Для интегрирования этой системы используется **явный метод Эйлера**:

|  |  |
| --- | --- |
|  | *(5)* |

После расчёта траекторий множества частиц можно восстановить распределение концентрации в любой момент времени, используя интерполяцию.

## Конечно-разностный метод

В отличие от метода частиц, конечно-разностный метод (КРМ) решает уравнение переноса в фиксированной сетке Эйлерова типа.

Пусть и – разностные аппроксимации производных по x и y, выбор которых зависит от направления скорости:

|  |  |
| --- | --- |
|  | *(6)* |

Тогда – значение концентрации в узле сетки в момент времени ​. Тогда конечно-разностная схема имеет вид:

|  |  |
| --- | --- |
|  | *(7)* |

# Анализ модели

Математическая модель, использованная для описания переноса пассивной примеси в двумерном несжимаемом потоке, обладает рядом важных физических и математических свойств, которые определяют характер поведения решения, устойчивость схемы и возможность интерпретации результатов с точки зрения реального течения. Проанализируем эти свойства.

## Не сжимаемость и сохранение массы

Использование функции тока автоматически гарантирует не сжимаемость потока, поскольку выполняется тождество:

|  |  |
| --- | --- |
|  | *(6)* |

Это означает, что поток не создаёт локального накопления или исчезновения вещества, и в силу отсутствия источников и стоков, концентрация пассивной примеси может изменяться только за счёт перемещения.

## Сохранение концентрации вдоль траектории

Рассмотрим систему характеристик — уравнения движения частицы в лагранжевой системе координат:

|  |  |
| --- | --- |
|  | *(7)* |

Вдоль каждой такой траектории можно показать, что производная концентрации равна нулю:

|  |  |
| --- | --- |
|  | *(8)* |

Отсюда следует, что **значение концентрации вдоль траектории сохраняется**:

|  |  |
| --- | --- |
|  | *(9)* |

Это ключевое свойство переноса пассивной примеси — отсутствие изменения концентрации на частице.

## Формулировка задачи Коши

Пусть начальное условие задано:

|  |  |
| --- | --- |
|  | *(10)* |

Ищется решение , удовлетворяющее уравнению переноса и начальным условиям.

Решение может быть записано в виде:

|  |  |
| --- | --- |
|  | *(11)* |

где – координаты начальной точки, из которой частица пришла в точку в момент .

## Устойчивость

Уравнение переноса – первого порядка, линейное, гиперболическое. Гиперболический характер уравнения означает, что возмущения распространяются вдоль характеристик (траекторий движения).

Рассмотрим два решения ​ и ​, с близкими начальными условиями ​ и ​. Их разность:

удовлетворяет тому же уравнению переноса, и её максимум по сохраняется:

.

Таким образом, задача устойчива к малым возмущениям начальных условий.

## Единственность и существование

Поскольку уравнение линейное, и характеристики однозначно определяются полем (при условии, что поле непрерывно), решение задачи Коши **существует и единственно** при заданной начальной функции .

# Вычислительные эксперименты

Целью численного эксперимента является исследование поведения решения двумерного уравнения переноса в различных подходах дискретизации: метод частиц и конечно-разностный метод. Основное внимание уделяется качеству аппроксимации поля концентрации и динамике его эволюции во времени.

## Условия эксперимента

* Область моделирования: квадрат ;
* Сетка: ;
* Количество частиц: ;
* Временной отрезок:;

Граничное условие:

где – граница области .

Начальное распределение концентрации:

Функция тока:

Тогда стационарное поле скорости :

* .

## Сравнение

В рамках сравнения были построены временные срезы для метода частиц и конечно-разностного метода. Будем сравнивать данные срезы. Левый график – конечно-разностный метод, а правый – метод частиц.

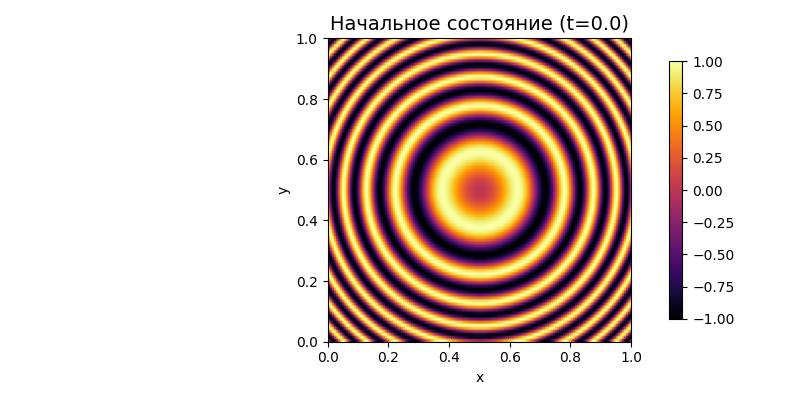


Рисунок 1 – Начальное распределение

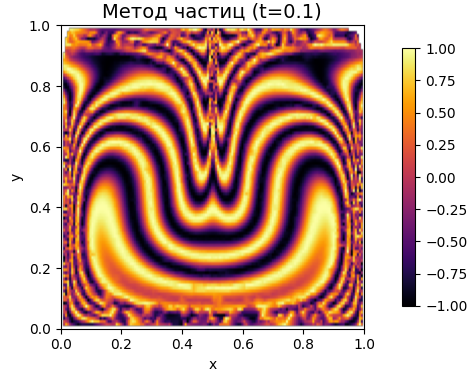
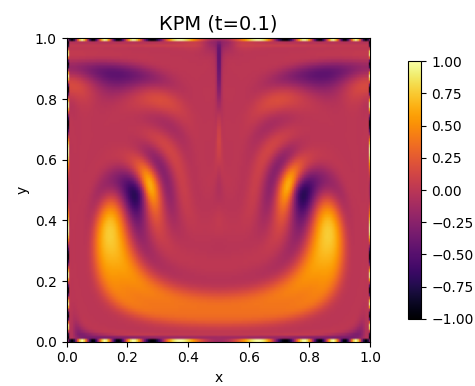


Рисунок 2 – Распределение в момент времени

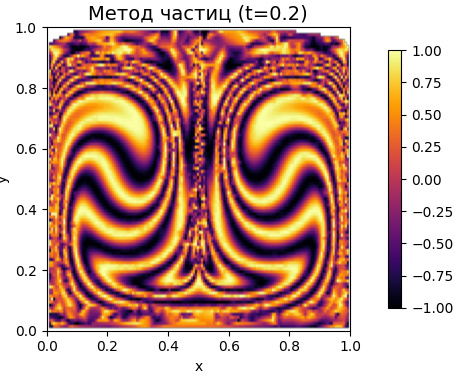
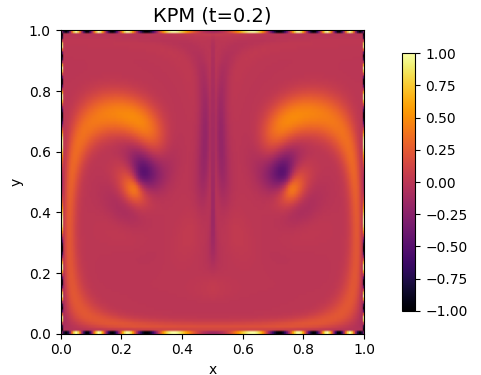


Рисунок 3 – Распределение в момент времени

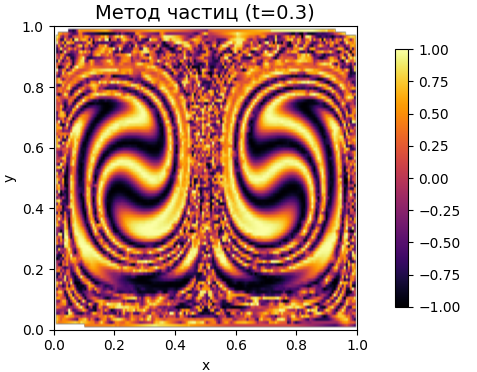
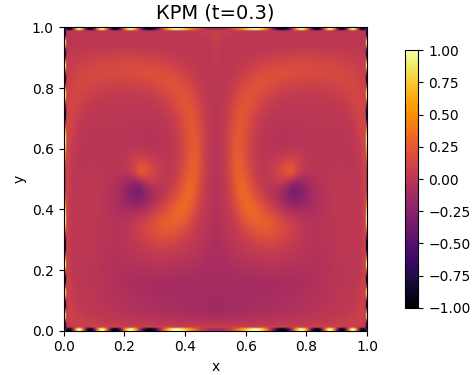


Рисунок 4 – Распределение в момент времени

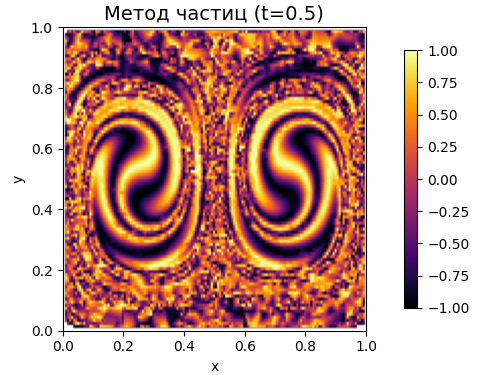
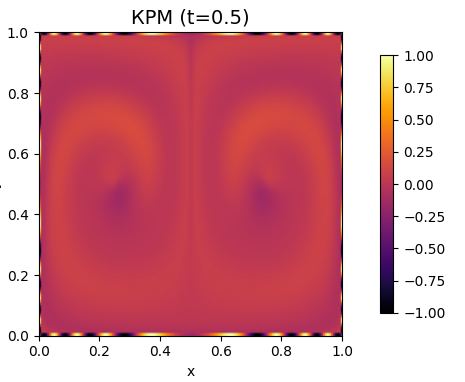


Рисунок 5 – Распределение в момент времени

## Результат

Анализ графиков, представленных в вычислительных экспериментах, позволяет сделать следующие выводы о сравнении метода частиц и конечно-разностного метода:

1. Четкость границ распределения:

* Метод частиц демонстрирует высокую четкость границ распределения концентрации на протяжении всего времени моделирования. Это связано с тем, что каждая частица сохраняет свою начальную концентрацию, а траектории их движения точно следуют характеристикам уравнения переноса.
* В конечно-разностном методе наблюдается размытие фронта концентрации уже на ранних этапах моделирования. Это обусловлено численной диффузией, которая неизбежно возникает при дискретизации уравнения на фиксированной сетке.

1. Сохранение структуры:

* Метод частиц лучше сохраняет начальную структуру распределения, включая мелкие детали и резкие градиенты. Это особенно важно для задач, где требуется точное воспроизведение локализованных особенностей (например, пятен загрязнения).
* Конечно-разностный метод, несмотря на схожесть общей структуры, теряет детализацию из-за аппроксимации производных и влияния шага сетки. Размытие становится более выраженным с увеличением времени.

1. Устойчивость к численным артефактам:

* Метод частиц менее подвержен таким артефактам, как осцилляции или ложные максимумы, которые могут возникать в конечно-разностных схемах из-за неустойчивости или недостаточной точности.
* В конечно-разностном методе численная диффузия играет роль "фильтра", сглаживающего резкие изменения, что может быть как недостатком (потеря точности), так и преимуществом (устойчивость схемы).

# Заключение

В данной работе была рассмотрена математическая модель двумерного уравнения переноса пассивной примеси в несжимаемом потоке. Модель основана на уравнении конвекции без диффузии, где перенос вещества определяется векторным полем скорости, заданным через функцию тока. Такой подход гарантирует не сжимаемость потока и сохранение массы вещества, что отражает физически корректное описание процесса переноса в несжимаемой среде.

Анализ модели показал, что решение уравнения переноса обладает важными свойствами:

* Концентрация вещества сохраняется вдоль траекторий частиц, что соответствует физическому принципу отсутствия источников и стоков вещества;
* Задача Коши с гладкими начальными условиями имеет единственное и устойчивое решение, чувствительность к малым возмущениям начального распределения отсутствует;
* Гиперболический характер уравнения обусловливает распространение возмущений вдоль характеристик, что требует аккуратного выбора численных методов для обеспечения стабильности и точности.

Для численного решения модели реализованы и сравнены два подхода: метод частиц и конечно-разностный метод. Метод частиц моделирует движение большого числа частиц, каждое из которых сохраняет исходную концентрацию, что обеспечивает высокую точность и сохранение чёткости фронтов распределения. В свою очередь, конечно-разностный метод работает с фиксированной сеткой и приближает производные с учётом направления потока, что повышает устойчивость схемы, но приводит к некоторому численному размазыванию фронтов из-за диффузионных артефактов.

Таким образом, выбор численного метода должен основываться на требованиях задачи: если важна высокая детализация и точное отслеживание переноса вещества – предпочтителен метод частиц, если же акцент ставится на стабильность и сглаженность решения – более подходящим будет конечно-разностный метод.

# Приложения

import numpy as np

import matplotlib.pyplot as plt

import os

from scipy.interpolate import griddata

from tqdm import tqdm

# =============================================================================

# 1. ОБЩИЕ ПАРАМЕТРЫ И ФУНКЦИИ

# =============================================================================

T\_MAX = 0.5

COMPARE\_TIMES = [0.1, 0.2, 0.3, 0.5]

NX, NY = 150, 100

NUM\_PARTICLES = 15000

def u(x, y):

    return -2\*np.pi \* np.sin(2\*np.pi \* x) \* np.cos(np.pi \* y)

def v(x, y):

    return 2\*np.pi \* np.cos(2\*np.pi \* x) \* np.sin(np.pi \* y)

def c\_0(x, y):

    return np.sin(100\*((x - 0.5)\*\*2 + (y - 0.5)\*\*2))

# =============================================================================

# 2. РЕАЛИЗАЦИЯ МЕТОДА ЧАСТИЦ (ЛАГРАНЖЕВ ПОДХОД)

# =============================================================================

def run\_particle\_method(target\_times, c\_0=c\_0):

    print("Запуск симуляции: Метод частиц...")

    points = np.random.rand(NUM\_PARTICLES, 2)

    concentrations = c\_0(points[:, 0], points[:, 1])

    dt = 0.001

    time = 0.0

    results = {}

    def velocity\_field(p):

        return np.array([u(p[:, 0], p[:, 1]), v(p[:, 0], p[:, 1])]).T

    max\_time\_to\_run = max(target\_times)

    times\_to\_save = sorted(target\_times)

    steps = int((max\_time\_to\_run + dt / 2) / dt)

    with tqdm(total=steps, desc="Метод частиц") as pbar:

        while time <= max\_time\_to\_run + dt/2: # Adjusted condition to ensure last time step is included

            if times\_to\_save and time >= times\_to\_save[0]:

                current\_save\_time = times\_to\_save.pop(0)

                print(f" Метод частиц: сохраняем срез на t={current\_save\_time:.1f}...")

                grid\_x, grid\_y = np.mgrid[0:1:complex(0, NX), 0:1:complex(0, NY)]

                interpolated\_grid = griddata(points, concentrations, (grid\_x, grid\_y), method='linear')

                results[current\_save\_time] = interpolated\_grid

            k1 = velocity\_field(points)

            k2 = velocity\_field(points + 0.5 \* dt \* k1)

            k3 = velocity\_field(points + 0.5 \* dt \* k2)

            k4 = velocity\_field(points + dt \* k3)

            points += (dt / 6.0) \* (k1 + 2 \* k2 + 2 \* k3 + k4)

            time += dt

            pbar.update(1)

    # Ensure the last time point is saved if it wasn't exactly matched by the loop

    if max(target\_times) not in results:

         print(f" Метод частиц: сохраняем срез на t={max(target\_times):.1f} (финальный)...")

         grid\_x, grid\_y = np.mgrid[0:1:complex(0, NX), 0:1:complex(0, NY)]

         interpolated\_grid = griddata(points, concentrations, (grid\_x, grid\_y), method='linear')

         results[max(target\_times)] = interpolated\_grid

    print("Метод частиц: симуляция завершена.")

    return results

# =============================================================================

# 3. РЕАЛИЗАЦИЯ КОНЕЧНО-РАЗНОСТНОГО МЕТИОДА (ЭЙЛЕРОВ ПОДХОД)

# =============================================================================

def run\_finite\_difference\_method(target\_times, initial\_C, U\_grid, V\_grid):

    print("Запуск симуляции: Конечно-разностный метод...")

    C = initial\_C.copy()

    dx = 1.0 / (NX - 1)

    dy = 1.0 / (NY - 1)

    u\_max, v\_max = np.max(np.abs(U\_grid)), np.max(np.abs(V\_grid))

    dt = 0.5 / (u\_max / dx + v\_max / dy)

    time = 0.0

    results = {}

    times\_to\_save = sorted(target\_times)

    max\_time\_to\_run = max(target\_times)

    steps = int((max\_time\_to\_run + dt / 2) / dt)

    with tqdm(total=steps, desc="К-Р метод") as pbar:

        while time <= max\_time\_to\_run + dt/2: # Adjusted condition to ensure last time step is included

            if times\_to\_save and time >= times\_to\_save[0]:

                current\_save\_time = times\_to\_save.pop(0)

                print(f" К-Р метод: сохраняем срез на t={current\_save\_time:.1f}...")

                results[current\_save\_time] = C.copy()

            C\_old = C.copy()

            # Apply boundary conditions before calculating derivatives

            C\_old[0, :] = C\_old[-2, :]

            C\_old[-1, :] = C\_old[1, :]

            C\_old[:, 0] = C\_old[:, -2]

            C\_old[:, -1] = C\_old[:, 1]

            for i in range(1, NX - 1):

                for j in range(1, NY - 1):

                    # Противопотоковая схема (upwind)

                    if U\_grid[j, i] >= 0:

                        dCdx = (C\_old[j, i] - C\_old[j, i - 1]) / dx

                    else:

                        dCdx = (C\_old[j, i + 1] - C\_old[j, i]) / dx

                    if V\_grid[j, i] >= 0:

                        dCdy = (C\_old[j, i] - C\_old[j - 1, i]) / dy

                    else:

                        dCdy = (C\_old[j + 1, i] - C\_old[j, i]) / dy

                    C[j, i] = C\_old[j, i] - dt \* (U\_grid[j, i] \* dCdx + V\_grid[j, i] \* dCdy)

            time += dt

            pbar.update(1)

    # Ensure the last time point is saved if it wasn't exactly matched by the loop

    if max(target\_times) not in results:

         print(f" К-Р метод: сохраняем срез на t={max(target\_times):.1f} (финальный)...")

         results[max(target\_times)] = C.copy()

    print("Конечно-разностный метод: симуляция завершена.")

    return results

# =============================================================================

# 4. ФУНКЦИЯ ДЛЯ СОХРАНЕНИЯ ОДИНОЧНОГО ГРАФИКА

# =============================================================================

def save\_single\_plot(data, title, filename):

    """Создает, настраивает и сохраняет один график."""

    fig, ax = plt.subplots(figsize=(8, 4))

    vmin, vmax = -1.0, 1.0 # Adjusted vmin/vmax based on initial condition range

    cmap = 'inferno'

    im = ax.imshow(data.T, extent=(0, 1, 0, 1), origin='lower', cmap=cmap, vmin=vmin, vmax=vmax)

    ax.set\_title(title, fontsize=14)

    ax.set\_xlabel('x')

    ax.set\_ylabel('y')

    fig.colorbar(im, ax=ax, shrink=0.85)

    plt.tight\_layout()

    plt.savefig(filename)

    plt.close(fig)

    print(f"Сохранено: {filename}")

# =============================================================================

# 5. ЗАПУСК И ВИЗУАЛИЗАЦИЯ

# =============================================================================

if \_\_name\_\_ == '\_\_main\_\_':

    x = np.linspace(0, 1, NX)

    y = np.linspace(0, 1, NY)

    X, Y = np.meshgrid(x, y)

    C\_initial = c\_0(X, Y)

    U\_grid = u(X, Y)

    V\_grid = v(X, Y)

    # --- Запуск симуляций ---

    particle\_results = run\_particle\_method(COMPARE\_TIMES)

    fdm\_results = run\_finite\_difference\_method(COMPARE\_TIMES, C\_initial, U\_grid, V\_grid)

    os.makedirs('lab7/comparison\_results', exist\_ok=True)

    print("\nСохранение начального состояния...")

    save\_single\_plot(C\_initial.T, 'Начальное состояние (t=0.0)', 'lab7/comparison\_results/initial\_state.png')

    print("\nСохранение результатов по временным срезам...")

    for t in COMPARE\_TIMES:

        p\_data = particle\_results[t]

        p\_title = f'Метод частиц (t={t:.1f})'

        p\_filename = f'lab7/comparison\_results/particle\_method\_t\_{t:.1f}.png'

        save\_single\_plot(p\_data, p\_title, p\_filename)

        fdm\_data = fdm\_results[t]

        fdm\_title = f'КРМ (t={t:.1f})'

        fdm\_filename = f'lab7/comparison\_results/fdm\_method\_t\_{t:.1f}.png'

        save\_single\_plot(fdm\_data.T, fdm\_title, fdm\_filename)

    print("\nВсе файлы успешно созданы.")