**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РФ**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский Авиационный Институт»**

**(Национальный Исследовательский Университет)**

**Институт: №8 «Информационные технологии   
и прикладная математика»   
Кафедра: 806 «Вычислительная математика   
и программирование»**

Лабораторная работа № 6   
по курсу «Численные методы»

Группа: М8О-407Б-22

Студент: Е.С. Кострюков

Преподаватель: Ю.В. Сластушенский

Оценка:

Дата: 26.12.2025

Москва, 2025

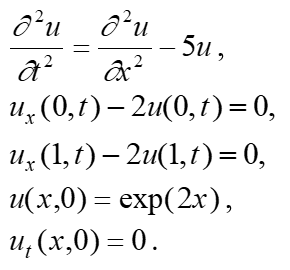
# **Тема**

Метод конечных разностей для решения уравнений гиперболического типа.

# **Задание**

Используя явную схему крест и неявную схему, решить начально-краевую задачу для дифференциального уравнения гиперболического типа. Аппроксимацию второго начального условия произвести с первым и со вторым порядком. Осуществить реализацию трех вариантов аппроксимации граничных условий, содержащих производные: двухточечная аппроксимация с первым порядком, трехточечная аппроксимация со вторым порядком, двухточечная аппроксимация со вторым порядком. В различные моменты времени вычислить погрешность численного решения путем сравнения результатов с приведенным в задании аналитическим решением  . Исследовать зависимость погрешности от сеточных параметров .

Вариант 4.





# **Листинг кода**

<https://github.com/EvgenyMAI/Study/blob/main/NumMethods/solutions/lab6/lab6.py>

Результаты:

*Максимальные погрешности*

*T = 0.5:*

*Явная схема (1 порядок): 0.014949*

*Неявная схема (1 порядок): 0.015061*

*Явная схема (2 порядок): 0.013841*

*Неявная схема (2 порядок): 0.013953*

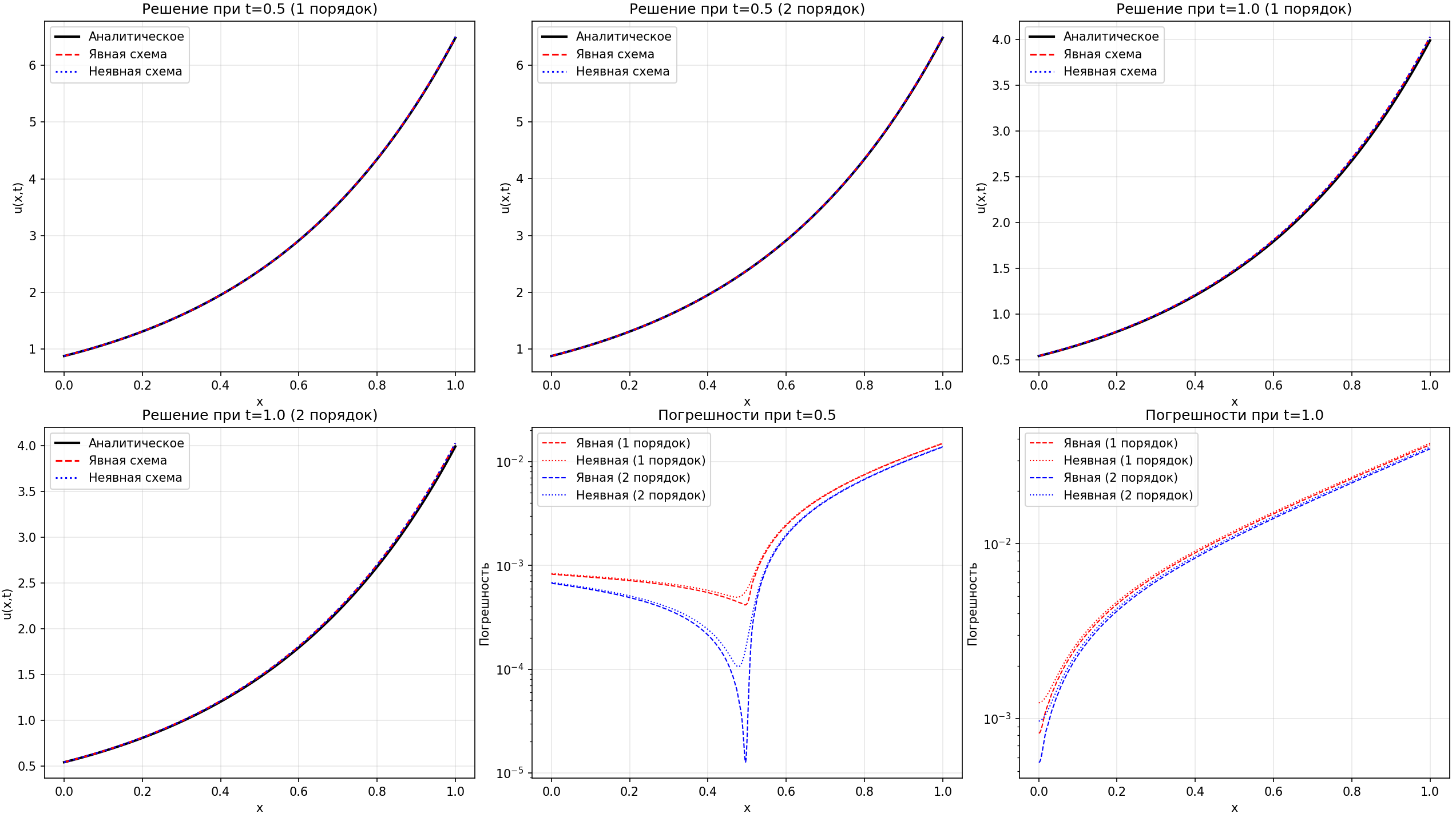
*T = 1.0:*

*Явная схема (1 порядок): 0.036862*

*Неявная схема (1 порядок): 0.037623*

*Явная схема (2 порядок): 0.034914*

*Неявная схема (2 порядок): 0.035676*



# **Метод решения**

**1. Метод решения и алгоритмическая реализация**

**1.1. Дискретизация и область моделирования**  
Задача решается в области . Вводится равномерная прямоугольная сетка:

* Пространственный шаг .
* Временной шаг .
* Критерий устойчивости (число Куранта): . Для явной схемы требуется выполнение условия .

**1.2. Разностные схемы для гиперболического уравнения**  
В работе реализованы два подхода к решению волнового уравнения (в простейшей форме, коэффициенты опущены для общности):

1. **Явная схема «крест»:**  
   Используется центральная разностная аппроксимация вторых производных по времени и пространству. Значение на новом слое вычисляется явно по трем точкам предыдущего слоя и одной точке пред-предыдущего слоя:  
   Схема имеет второй порядок точности , но условно устойчива.
2. **Неявная схема:**  
   Вторая производная по координате аппроксимируется на верхнем (новом) временном слое. Это приводит к системе уравнений относительно :  
   Схема является абсолютно устойчивой при любых значениях и .

**1.3. Аппроксимация начальных условий**  
Для гиперболического уравнения необходимо задать два начальных условия: и .

* Первый слой задается точно функцией .
* Второй слой (необходимый для старта трехслойных схем) вычисляется двумя способами:
  1. **Первый порядок точности:** (грубая аппроксимация, предполагающая нулевую начальную скорость или пренебрежение ей на малом шаге).
  2. **Второй порядок точности:** Используется разложение в ряд Тейлора:  
     Вторая производная по времени заменяется на пространственную из уравнения: .

**1.4. Учет граничных условий и формирование СЛАУ**  
Граничные условия третьего рода (вида ) аппроксимированы с помощью двухточечных разностей первого порядка точности.

* В **явной схеме** граничные значения и вычисляются после нахождения решения во внутренних узлах, используя выраженные явные формулы:
* В **неявной схеме** граничные условия встраиваются в матрицу системы:
  + Левая граница вносит вклад в первую строку матрицы (коэффициенты ).
  + Правая граница модифицирует последнюю строку матрицы (коэффициенты ).  
    Полученная трехдиагональная система уравнений решается методом **прогонки (алгоритм Томаса)** на каждом временном шаге.

**1.5. Программная реализация**  
Алгоритм реализован на Python. Функции solve\_explicit и solve\_implicit инкапсулируют логику пересчета слоев. В основном теле программы производится сравнение численных решений с аналитическим и вычисление максимальной ошибки на сетке.

# **Выводы**

# В ходе работы исследованы численные методы решения гиперболического уравнения.

1. **Влияние начальной аппроксимации:**  
   Сравнение результатов показало критическую важность корректной аппроксимации второго начального условия.
   * При использовании аппроксимации **1-го порядка** () ошибка велика даже при малых шагах по времени, так как игнорируется начальная динамика процесса.
   * Аппроксимация **2-го порядка** (через ряд Тейлора и уравнение) существенно снижает глобальную погрешность, позволяя реализовать потенциальный второй порядок точности самих разностных схем. Графики погрешностей наглядно демонстрируют разницу в несколько порядков.
2. **Сравнение схем:**
   * **Явная схема** проста в реализации и эффективна по времени выполнения (не требует решения СЛАУ), однако ограничена условием Куранта. Нарушение условия приводит к мгновенному разрушению решения.
   * **Неявная схема** требует больших вычислительных затрат (метод прогонки), но обладает запасом устойчивости.
3. **Анализ погрешности:**  
   Максимальная ошибка растет со временем ( против ), что характерно для эволюционных задач из-за накопления дисперсионной ошибки схемы. При использовании согласованных порядков аппроксимации (схема 2-го порядка + старт 2-го порядка) численное решение визуально совпадает с аналитическим, подтверждая корректность реализации алгоритмов.