Федеральное государственное автономное образовательное учреждение
высшего образования
«Московский физико-технический институт (Национальный исследовательский университет)»
Физтех-школа аэрокосмических технологий
Кафедра вычислительной физики

Направление подготовки: 03.03.01 Прикладные математика и физика (бакалавриат)
Направленность(профиль) подготовки: Вычислительные технологии математического моделирования

Аппроксимация высокого порядка на нерегулярной расчётной сетке без использования вспомогательных узлов на рёбрах и гранях

(бакалаврская работа)

Студент:
Смирнов Иван

____

Научный руководитель:
Васюков Алексей Викторович

____

Москва 2022

1 Аннотация

smth

Содержание

2 Введение

2.1 Цель работы

Главной задачей данной работы является создание подхода для решения двумерного уравнения переноса на нерегулярных расчётных двумерных сетках с повышенным порядком аппроксимации.

2.2 Объект исследования

В данной работе рассматривается численное решение двумерного уравнения переноса на двумерных нерегулярных расчётных сетках. Решаемое уравнение для функции $u(x, y, t)$ в квадрате $[-1, 1] \times [-1, 1]$ с периодическими граничными условиями имеет следующий вид:

$$\begin{cases} u_t + \lambda_x u_x + \lambda_y u_y = 0 \\ u(x, y, t)\Big|_{t=0} = F(x, y) \\ u(a \cdot T_x + x, b \cdot T_y + y, t) = u(x, y, t) \\ T_x = T_y = 2;\ a, b \in Z \\ x \in [-1, 1],\ y \in [-1, 1] \end{cases} \tag{1}$$

Здесь $F(x, y)$ - функция начальных условий; $T_x,\ T_y$ - период функции $u(x, y, t)$; $\lambda_x,\ \lambda_y$ - скорости по соответствующим направлениям.

2.3 Сетка в расчётной области

Для решения данной задачи обычно используется сетка из треугольников в области.

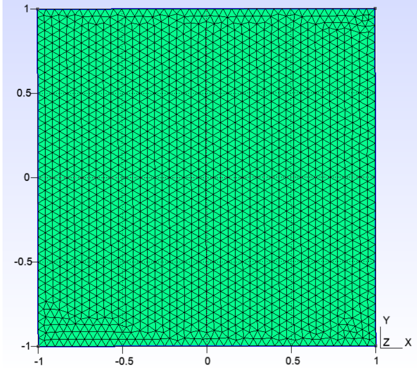

Рис. 1: Пример сгенерированной сетки, scale = 0.05

3 Теоретические сведения

3.1 Дискретизация

Для дискритизации уравнения по времени используется равномерная сетка по времени с шагом τ, такая что $T = N \cdot \tau$. Где число N - количество слоёв по времени.
На каждом слое по времени используется неравномерная сетка из треугольников, генерируемая при помощи таких алгоритмов как *Delaunay, MeshAdapt, Frontal-Delanay*.

3.2 Расщепление по пространству

Сначала задача расщепляется по пространственным переменным на два независимых уравнения, решаемых последовательно на каждой временной итерации.

$$\begin{cases} \frac{\partial u}{\partial t} + \lambda_x \cdot \frac{\partial u}{\partial x} = 0 \\ \frac{\partial u}{\partial t} + \lambda_y \cdot \frac{\partial u}{\partial y} = 0 \end{cases}$$

Для решения каждого из получившихся одномерных уравнений переноса с постоянными коэффициентами, используется классический подход - *сеточно-характеристический метод*.

3.3 Сеточно - характеристический метод

Рассмотрим данных метод на примере уравнения для x координаты относительно $w(x, y, t)$:

$$\frac{\partial w}{\partial t} + \lambda_x \cdot \frac{\partial w}{\partial x} = 0$$

Он заключается в сведении дифференциального уравнения первого порядка в частных производных к обыкновенному дифференциальному уравнению вдоль характеристики:

$$\frac{dx}{dt} = \lambda_x$$

Тогда:

$$\frac{\partial w}{\partial t} + \lambda_x \cdot \frac{\partial w}{\partial x} = 0 \to \frac{\partial w}{\partial t} + \frac{dx}{dt} \cdot \frac{\partial w}{\partial x} = 0 \to \frac{dw}{dt} = 0$$

То есть вдоль характеристики $dx/dt = \lambda_x$ решение не зависит от времени:

$$w(x, y, t)\Big|_{dx/dt=\lambda_x} = w(x, y)$$

Для нахождения значения функции в момент времени t^{n+1}, опускается характеристика(прямая, задаваемая уравнением $x = \lambda_x \cdot t$) на предыдущий слой по времени t^n. В точке пересечения $(x_0,\ y_0,\ t^n)$ этой прямой с плоскостью $t = t^n = \tau \cdot n = const$ аппроксимируется значение функции, с использованием известных значений в точках на данном слое по времени. Далее это значение переносится в точку $(x_0 + \lambda_x \cdot \tau,\ y_0,\ t^{n+1})$.

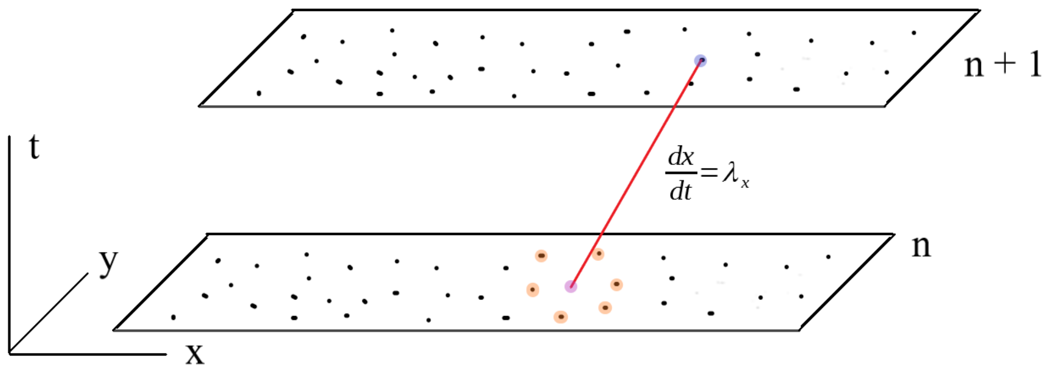

Рис. 2: Пример действия сеточно-характеристического метода

3.4 Аппроксимация значений функции

Для аппроксимации значения функции на предыдущем слое по времени в точке, полученной при помощи сеточно-характеристического метода можно использовать несколько подходов.

3.4.1 Использование производных в вершинах

- Определяется треугольник, в который попадает характеристика, опущенная на предыдущий слой по времени.
- Необходимо определить коэффициенты интерполяционного полинома
  $P(x, y) = \alpha \cdot x^2 + \beta \cdot y^2 + \gamma \cdot xy + \delta \cdot x + \epsilon \cdot y + \zeta$
  $\alpha, \beta, \gamma, \delta, \epsilon, \zeta \in \mathbb{R}$
  Известны значения функции u в точках $u_1 = u(x_1, y_1), u_2 = u(x_2, y_2), u_3 = u(x_3, y_3)$
- Составляется система уравнений:

$$\begin{cases} P(x_1, y_1) = u_1 \\ P(x_2, y_2) = u_2 \\ P(x_3, y_3) = u_3 \\ \frac{\partial P(x,y)}{\partial x}\Big|_{(x_1,y_1)} = (u_1)'_x \\ \frac{\partial P(x,y)}{\partial x}\Big|_{(x_2,y_2)} = (u_2)'_x \\ \frac{\partial P(x,y)}{\partial x}\Big|_{(x_3,y_3)} = (u_3)'_x \\ \frac{\partial P(x,y)}{\partial y}\Big|_{(x_1,y_1)} = (u_1)'_y \\ \frac{\partial P(x,y)}{\partial y}\Big|_{(x_2,y_2)} = (u_2)'_y \\ \frac{\partial P(x,y)}{\partial y}\Big|_{(x_3,y_3)} = (u_3)'_y \end{cases}$$

- Видно, что получается переопределённая система (9 уравнений, 6 неизвестных).

3.4.2 Обратно взвешенные расстояния(IDW)

- Выбирается какая-то область вокруг точки, в которой(точке) необходимо определить значение функции.

  Область - окружность определённого радиуса, либо иная геометрия по характеру задачи.

- Точки с данными внутри области участвуют в расчётах. Значение функции в точке вычисляется по следующей формуле:

$$f = \sum_i u_i \cdot \omega_i,\ \omega_i = d^{-p}$$

  Где ω_i - весовой коэффициент для каждой используемой точки с данными, d - расстояние от точки с неизвестым значением до данной, $p > 1$ - степень - подбирается экспериментально.

3.4.3 Безье-поверхности

- Параметризуем область $(x, y) \longrightarrow (u, v)$ так, чтобы $(u, v) = [0, 1] \times [0, 1]$.

- Определим поверхность Безье порядка (n, m) (задаётся $(n+1)\cdot(m+1)$ контрольными точками $P_{i,j}$):

$$P(u, v) = \sum_{i=0}^{n} \sum_{j=0}^{m} B_i^n(u) \cdot B_j^m(v) \cdot P_{i,j}$$

- B - многочлены Берштейна:

$$B_i^n(u) = \frac{n!}{i!(n-i)!} \cdot u^i \cdot (1-u)^{n-1}$$

- Есть готовые реализации данного подхода, например алгоритм Клафа-Точера

  `scipy.interpolate.CloughTocher2DInterpolator`

3.4.4 Построение интерполяционного полиному по k ближайшим точкам

- Необходимо определить коэффициенты интерполяционного полинома
  $P(x, y) = \alpha \cdot x^2 + \beta \cdot y^2 + \gamma \cdot xy + \delta \cdot x + \epsilon \cdot y + \zeta$
  $\alpha, \beta, \gamma, \delta, \epsilon, \zeta \in \mathbb{R}$
  Видно, что полином 2 порядкаа требует 6 коэфициентов, для лучшей точности можно использовать полином 3 порядка, для которого понадобиться 10 коэффициентов.

- Находятся ближайшие 6 точек к данной и по значением в них строится следующая система:

$$\begin{cases} P(x_1, y_1) = u_1 \\ P(x_2, y_2) = u_2 \\ P(x_3, y_3) = u_3 \\ P(x_4, y_4) = u_4 \\ P(x_5, y_5) = u_5 \\ P(x_6, y_6) = u_6 \end{cases} \tag{2}$$

  Это также можно записать в виде: $A \cdot \vec{x} = \vec{u}$, где $\vec{x}$ - вектор, составленный из выбранных ближайших точек, A - матрица из коэффициентов интерполяционного полинома, $\vec{u}$ - вектор, составленный из значений функции в выбранных точках.

- Решая эту СЛАУ , находятся необходимые коэффициенты, и определяется значение функции в требуемой точке.

3.4.5 Используемый метод аппроксимации

В работе используется именно последний подход. Он хорошо действует на функциях начальных условий из класса C^1, но хуже на разрывных - если рвётся производная функции или сама функция.

Для решения этой проблемы применялся подход с размытием места разрыва. А именно на первых итерациях по времени определялись места разрыва, и на них проводилась аппроксимация первым порядком. Места разрыва оценивались по первым и вторым производным функции(полученным из формального дифференцирования интерполяционного полинома вторй степени). То есть алгоритм был следующим:

Algorithm 1 Размытие разрывного решения

```
while n < N do
    for all Points do
        P2 = Make2orderPolinom(n, Point)
        Der = DetermineDerivatives(P2)
        if RoughPoint(Der) and n < N_0 then
            P1 = Make1orderPolinom(n, Point)
            CalculateValue(P1, Point)
        else
            CalculateValue(P2, Point)
```

- Количество первых шагов по времени N_0, на которых производилось размытие решений, определялось экспериментально, но не превышало 20% от общего времени расчёта.
- Условия разрывности RoughPoint() также подбирались самостоятельно. Сначала на разрывных решениях строилась картина производных, затем визуально определялись места разрыва, и из значений производных в этих точках составлялся критерий разрыва.

К сожалению размытие начальных условий не давало существенных улучшений на разрывных решениях, но портило гладкие задачи. Поэтому было решено отказаться от этого и использовать второй порядок точности на всех решениях.

Главная особенность подхода «Построение интерполяционного полиному по k ближайшим точкам» - не используются вспомагательные точки на рёбрах или гранях сетки в расчётной области. В задаче используются только узлы данной сетки.

3.5 Расчёт порядка аппроксимации

3.5.1 Аналитическое решение

Для рассматриеваемой задачи существует аналитическое решение:

$$u_{analitic}(x, y, t) = F(x - \lambda_x \cdot t, y - \lambda_y \cdot t)$$

3.5.2 Невязка

Численное решение связано с точным аналитическим решение следующим образом:

$$u_{numeric} = u_{analytic} + R(h)$$

Где $R(h)$ - невязка - функция от мелкости пространственной сетки h.
В качестве h используется:

- Обратный масштаб сетки $1/\text{scale}$ - (scale - наибольший линейный размер ячеек).
- Обратный корень количества точек сетки $1/\sqrt{\text{dots num}}$.

Предполагается, что:

$$R(h) = h^p + o(h^p)$$

Таким образом, порядок аппроксимации p может быть определён следующим образом:

$$\Delta = |u_{numeric} - u_{analytic}| = h^p + o(h^p) \rightarrow \ln \Delta / \ln h \simeq p$$

3.5.3 Нормы ошибки

Для определения Δ используются несколько норм векторов(предполагается что значения в точках на каждом слое по времени можно занумеровать):

- Максимум модуля ошибки:
  $\Delta_1 = \Delta_\infty = \max(|u_{numeric}[i] - u_{analytic}[i]|)$.
- Среднее арифмитическое модулей ошибок:
  $\Delta_2 = \sum_{i=0}^{N} |u_{numeric}[i] - u_{analytic}[i]|/N$
- Евклидова норма:
  $\Delta_3 = \sqrt{\sum_{i=0}^{N} (u_{numeric}[i] - u_{analytic}[i])^2/N}$

3.5.4 Алгоритм расчёта порядка аппроксимации

То есть для определения порядка сходимости решения строилось аналитическое решение в узлах сетки, затем вычислялись численные значения в этих же узлах. После этого рассчитывался вектор ошибки(невязки) - предполагается что все точки уникальные(их можно занумеровать и поместить по порядку в вектор). Определяются нормы этого вектора, как написанно в предыдущем пункте. Такие действия проделываются на каждом слое по времени, и в итоге по результатам норм векторов ошибок на нескольких сетках(с известными величинами шага по времени h) строится прямая в координатах $(\ln\Delta, \ln h)$ - по её наклону определяется порядок аппроксимации p.

Данную прямую можно построить с помощью метода наименьших квадратов. Пусть прямая имеет вид: $y = a+p\cdot h$. Тогда наилучшая прямая(в смысле $\sum_i |y_i - \ln\Delta_i| \to min$) будет при:

- $$p = \frac{\langle \ln h_i \cdot \ln\Delta_i\rangle - \langle \ln h_i\rangle \cdot \langle \ln\Delta_i\rangle}{\left\langle \ln^2 h_i\right\rangle - \langle \ln h_i\rangle^2}$$

- $$a = \langle \ln\Delta_i\rangle - p\,\langle \ln h_i\rangle$$

Где скобками $\langle ...\rangle$ обозначено среднее арифметическое значение выражения. Реально порядок аппроксимации оценивался не на всех итерациях по времени, а только на последнем слое.

4 Описание алгоритма

4.1 Постановка задачи

Перед расчётом необходимо определить функцию начальных условий $F(x, y)$, числа λ_x, λ_y, шаг по времени τ, количество шагов по времени N, построить сетку в рассматриеваемой области.

На каждом слое по времени определяются значения функции в точках сетки(также предполагается, что все точки можно занумеровать - их количество не превосходит M).

Algorithm 2 Постановка задачи

```
TimeLays[N][M]
while n < N do
    t_cur = τ · n
    for all Points do
        if n == 0 then
            TimeLays[0][Point] = InitialCondition(Point)
        else
            TimeLays[n][Point]= CalculateFunctionValueInPoint(n, Point)
    n = n + 1
```

Массив TimeLays[N][M] содержит значения искомой функции в точках сетки (Point = (x, y)) на каждом слое по времени, при этом используются:

- функции начальных и граничных условий(в данной работе граничные условия для всех задач одни, а начальные уловия различаются)
- функция вычисления значения функции на *n-ом* слое по времени

4.2 Начальные и граничные условия

Для решения поставленной задачи с заданным начальным условием $F(x, y)$ и периодическими граничными условиями с периодом $T = T_x = T_y = 2$ в области $[-1, 1] \times [-1, 1]$, применяются следующие условия:

Algorithm 3 Начальные и граничные условия

```
function InitialCondition((x, y))
    return F(x, y)
function BoundaryConditions((x, y))
    T = T_x = T_y = 2
    x = x − sign(x) · T_x · int(x/T_x)
    if |x| ≥ 1 then
        x = x − sign(x) · T_x
    y = y − sign(y) · T_y · int(y/T_y)
    if |y| ≥ 1 then
        y = y − sign(y) · T_y
    return (x, y)
```

4.3 Нахождение значения функции в точке

- Для нахождения значения функции в точке (x, y) используется точка (x_0, y_0) падения характеристики на слой по времени $t_{cur} - \tau$ с учётом периодических граничных условий.
- В данной работе используется структура хранения данных *K-D Tree* со встроенным поиском ближайших соседей.
- Так как найденные *k-nearest* точки располагаются близко к (x_0, y_0), применяется переход к локальной системе координат(отображение на единичный квадрат) вблизи этой точки.
- Число k выбирается несколько больше, чем необходимое количество точек для выбранного порядка аппроксимации, для возможность перебрать их комбинации и выбрать ту, которая даёт наибольший детерминант матрицы системы.

Algorithm 4 Определение значения функции в точке

```
function CalculateFunctionValueInPoint(n, (x, y))
    x_0 = x − λ_x · τ
    y_0 = y − λ_y · τ
    (x_0, y_0) = BoundaryConditions((x_0, y_0))
    dots = FindKNearest((x_0, y_0))
    values = GetValuesFromTimeLay(n, dots)
    w = InterpolateFunctionValue(dots, values, (x_0, y_0))
    return w
```

5 Результаты численного эксперимента

Расчётная область квадрат $[-1, 1] \times [-1, 1]$. Количество слоёв по времени $N = 51$, время моделирования $T = 1$, постоянный шаг по времени $\tau = T/(N-1) = 0.02$.
Скорости распространения возмущений по осям OX и OY соответственно равны: $\lambda_x = -2, \lambda_y = 5$.

5.1 Гладкая шапочка

Начальное условие

$$F(x, y) = \cos^4(x \cdot \pi/2) \cdot \cos^4(y \cdot \pi/2) \tag{3}$$

Визуализация решения

Ниже приведено решение в разные моменты времени, указан номер шага по времени $n \in [0, N-1]$ и фактическое время $t = n \cdot \tau$.

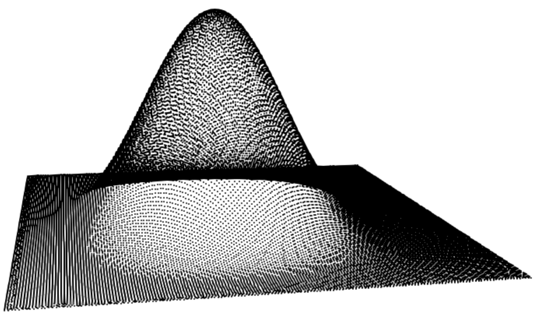

Рис. 3: $t = 0.0, n = 0$

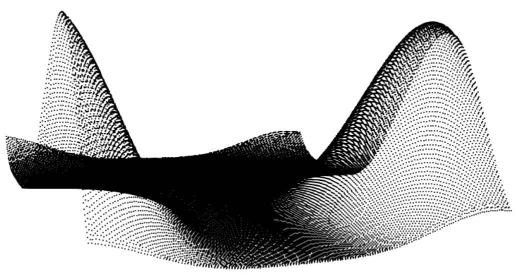

Рис. 4: $t = 0.22, n = 11$

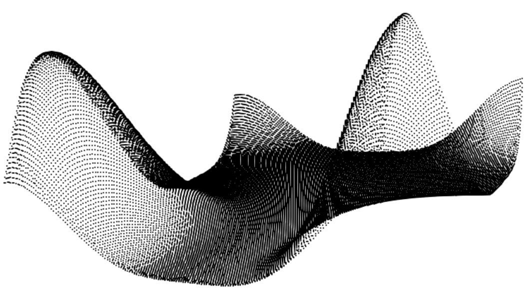

Рис. 5: $t = 0.52, n = 26$

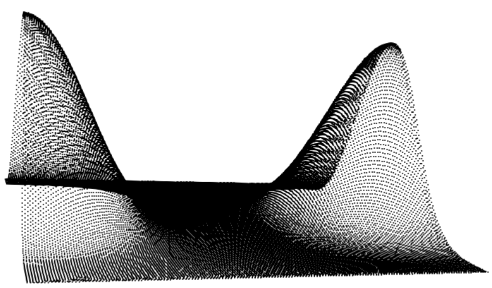

Рис. 6: $t = 1.0, n = 50$

Порядок сходимости

При использовании в качестве аналога шага по пространству $1/\sqrt{\text{dots number}}$ получены наиболие репрезентативные результаты порядка сходимости, ниже приведён график для второй и третьей норм.

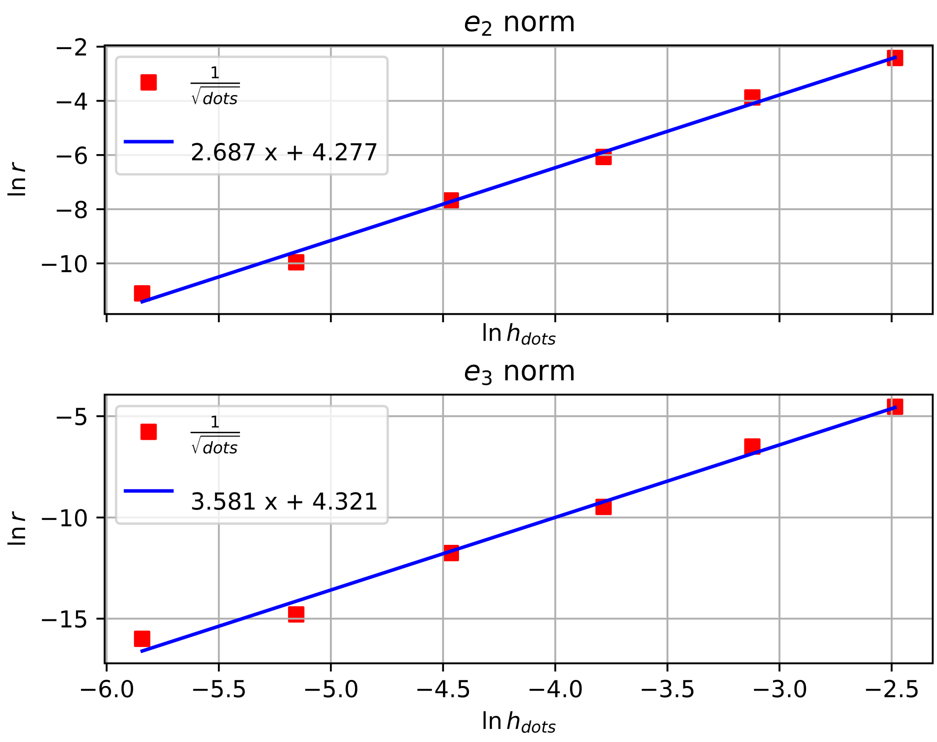

Рис. 7: График порядка сходимости численного решения для начальных условий типа «гладкая шапочка». Вторая норма - порядок сходимости 2.687, третья 3.581.

Результаты всех вычисляемых норм проиллюстрированны в таблице ниже. В первом столбце в качестве шага по пространству используется $h_{dots} = 1/\sqrt{\text{dots number}}$, во втором $h_{scale} = 1/\text{scale}$

	$1/\sqrt{\text{dots number}}$	$1/\text{scale}$
e_1	1.914	1.864
e_2	2.687	2.610
e_3	3.581	3.479

Таблица 1: Таблица с результатами расчётов порядка сходимости с помощью различных типов норм и аналогов шагов по пространству для начальных условий типа «гладкая шапочка».

5.2 Быстро спадающая экспонента

Начальное условие

$$F(x,y) = \begin{cases} \exp\left(-84 \cdot (x^2+y^2)\right), (x,y) \in [-0.2, 0.2] \times [-0.2, 0.2] \\ 0, (x,y) \notin [-0.2, 0.2] \times [-0.2, 0.2] \end{cases} \tag{4}$$

Визуализация решения

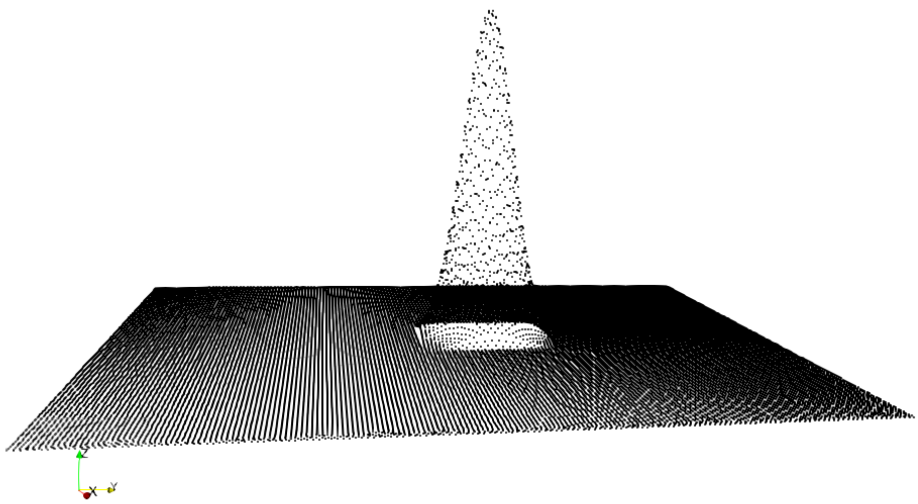

Рис. 8: Быстро спадающая экспонента в начальный момент времени

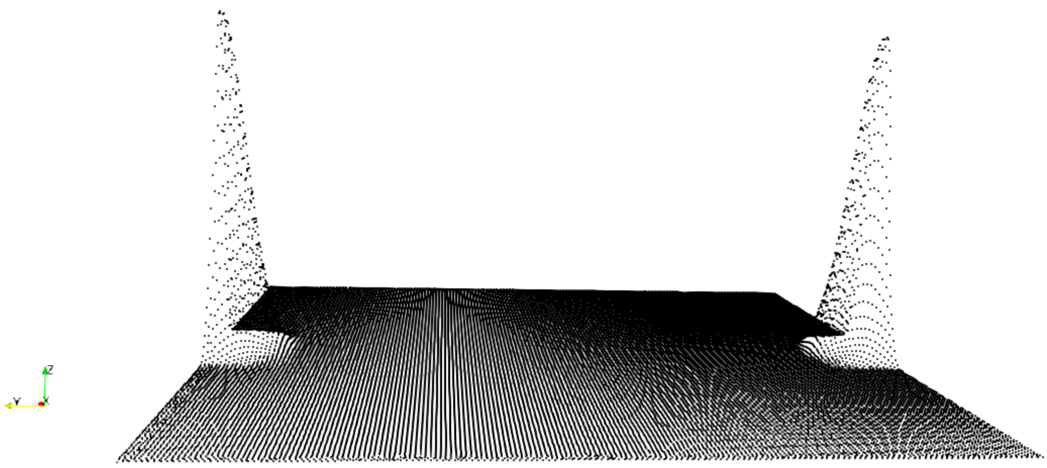

Рис. 9: Быстро спадающая экспонента в конечный момент времени

Порядок сходимости

При использовании в качестве аналога шага по пространству $1/\sqrt{\text{dots number}}$ получены наиболие репрезентативные результаты порядка сходимости, ниже приведён график для второй и третьей норм.

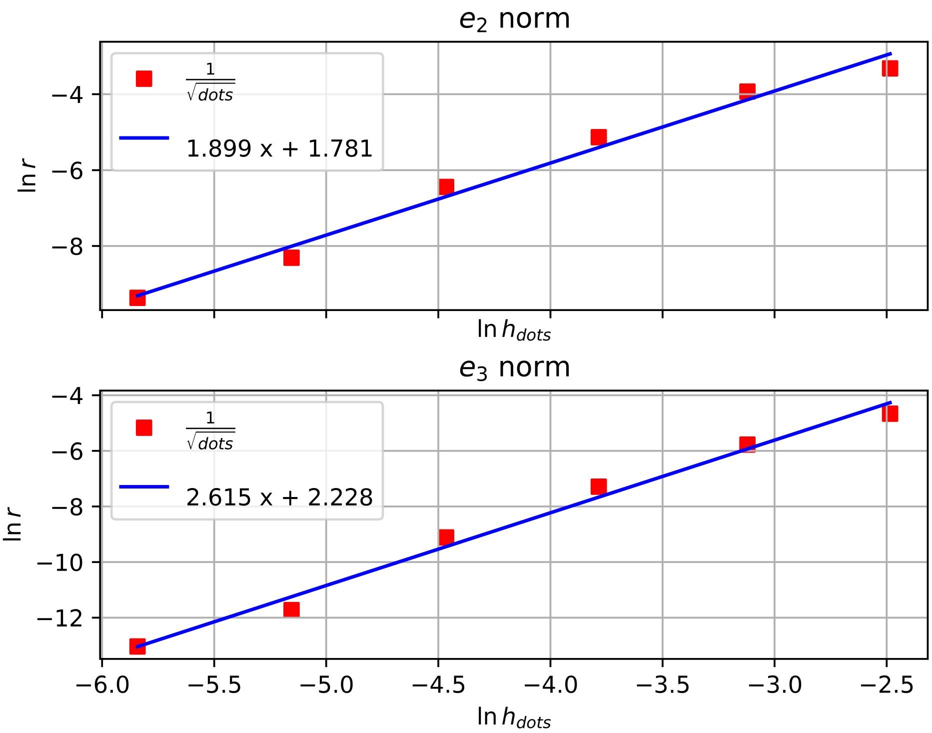

Рис. 10: График порядка сходимости численного решения для начальных условий типа «быстро спадающая экспонента». Вторая норма - порядок сходимости 1.899, третья 2.615.

Результаты всех вычисляемых норм проиллюстрированны в таблице ниже. В первом столбце в качестве шага по пространству используется $h_{dots} = 1/\sqrt{\text{dots number}}$, во втором $h_{scale} = 1/\text{scale}$

	$1/\sqrt{\text{dots number}}$	$1/\text{scale}$
e_1	1.242	1.203
e_2	1.899	1.842
e_3	2.615	2.537

Таблица 2: Таблица с результатами расчётов порядка сходимости с помощью различных типов норм и аналогов шагов по пространству для начальных условий типа «быстро спадающая экспонента».

5.3 Конус

Начальное условие

$$F(x,y) = \begin{cases} (1-5\cdot|x|)\cdot(1-5\cdot|y|), (x,y)\in[-0.2,0.2]\times[-0.2,0.2] \\ 0, (x,y)\notin[-0.2,0.2]\times[-0.2,0.2] \end{cases} \tag{5}$$

Визуализация решения

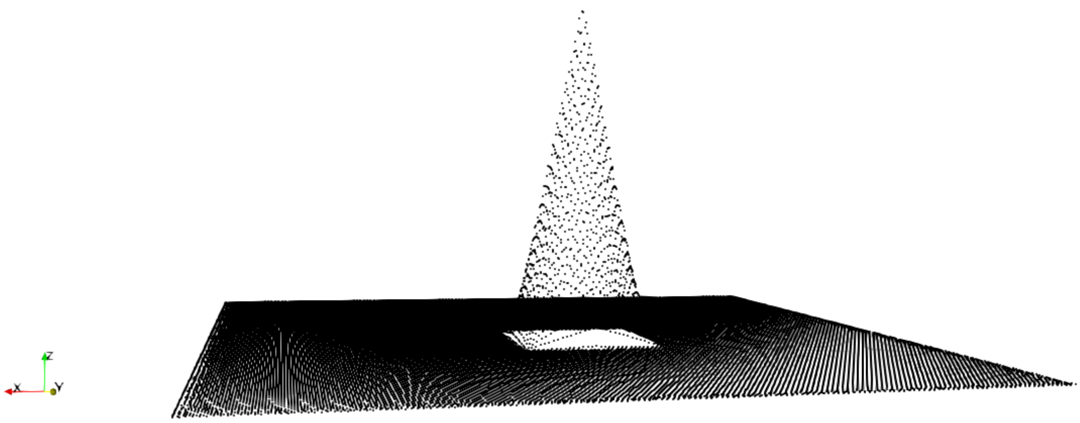

Рис. 11: Конус в начальный момент времени

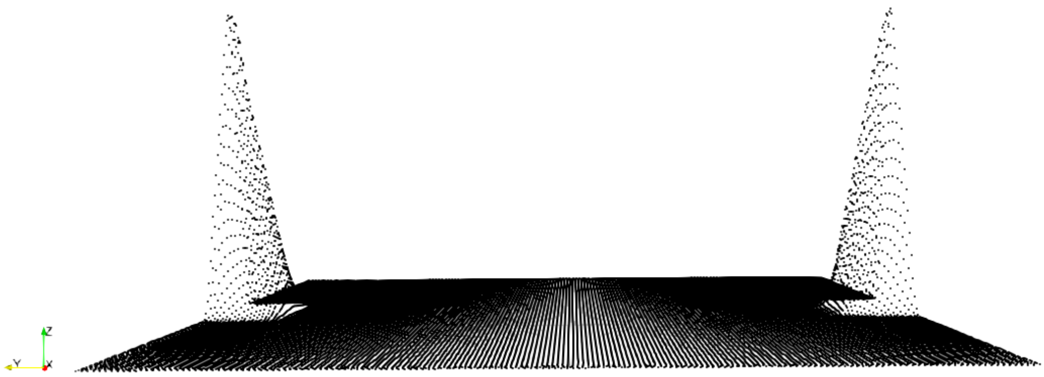

Рис. 12: Конус в конечный момент времени

Порядок сходимости

При использовании в качестве аналога шага по пространству $1/\sqrt{\text{dots number}}$ получены наиболие репрезентативные результаты порядка сходимости, ниже приведён график для второй и третьей норм.

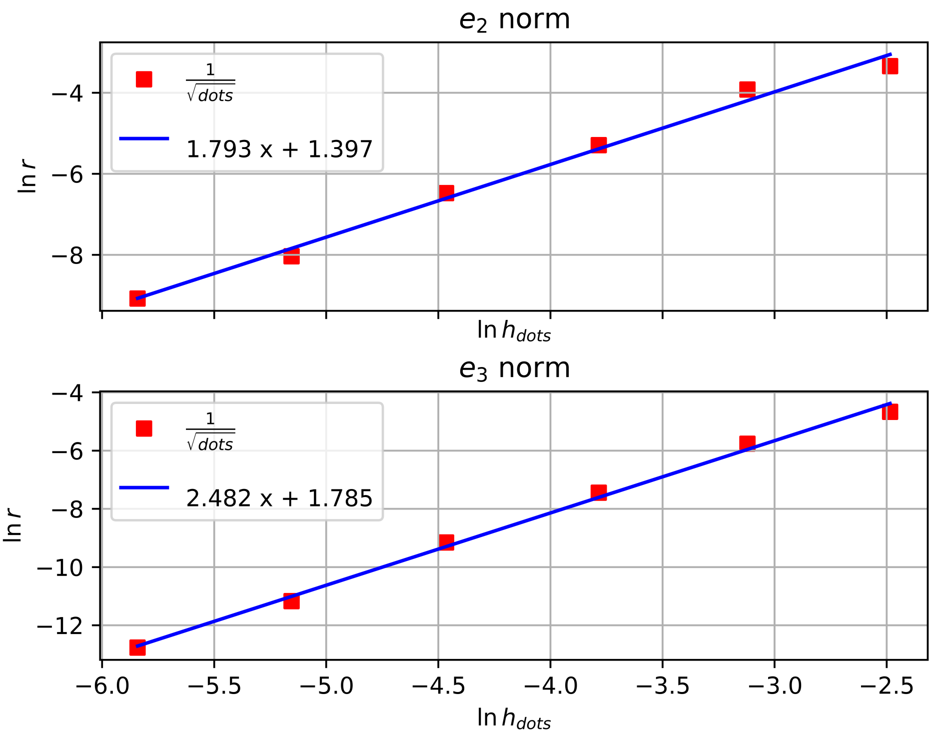

Рис. 13: График порядка сходимости численного решения для начальных условий типа «конус». Вторая норма - порядок сходимости 1.793, третья 2.482.

Результаты всех вычисляемых норм проиллюстрированны в таблице ниже. В первом столбце в качестве шага по пространству используется $h_{dots} = 1/\sqrt{\text{dots number}}$, во втором $h_{scale} = 1/\text{scale}$

	$1/\sqrt{\text{dots number}}$	$1/\text{scale}$
e_1	0.888	0.861
e_2	1.793	1.739
e_3	2.482	2.408

Таблица 3: Таблица с результатами расчётов порядка сходимости с помощью различных типов норм и аналогов шагов по пространству для начальных условий типа «конус».

5.4 Шапочка-корень

Начальное условие

$$F(x,y) = \begin{cases} \sqrt{(1 - 25 \cdot x^2) \cdot (1 - 25 \cdot y^2)}, (x,y) \in [-0.2, 0.2] \times [-0.2, 0.2] \\ 0, (x,y) \notin [-0.2, 0.2] \times [-0.2, 0.2] \end{cases} \tag{6}$$

Визуализация решения

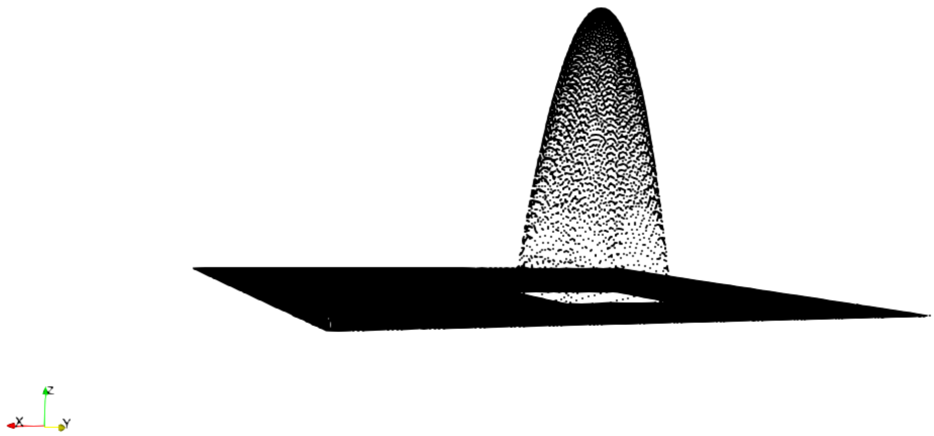

Рис. 14: Шапочка-корень в начальный момент времени

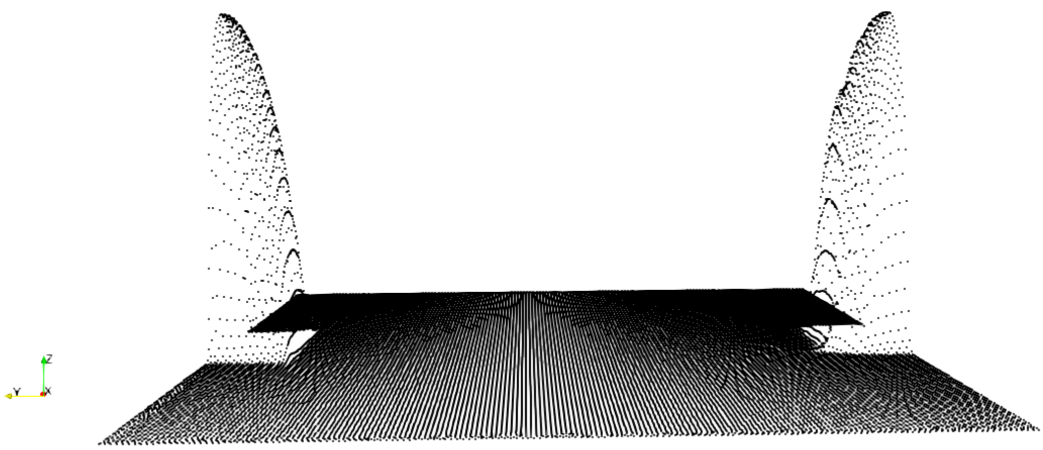

Рис. 15: Шапочка-корень в конечный момент времени

Порядок сходимости

При использовании в качестве аналога шага по пространству $1/\sqrt{\text{dots number}}$ получены наиболие репрезентативные результаты порядка сходимости, ниже приведён график для второй и третьей норм.

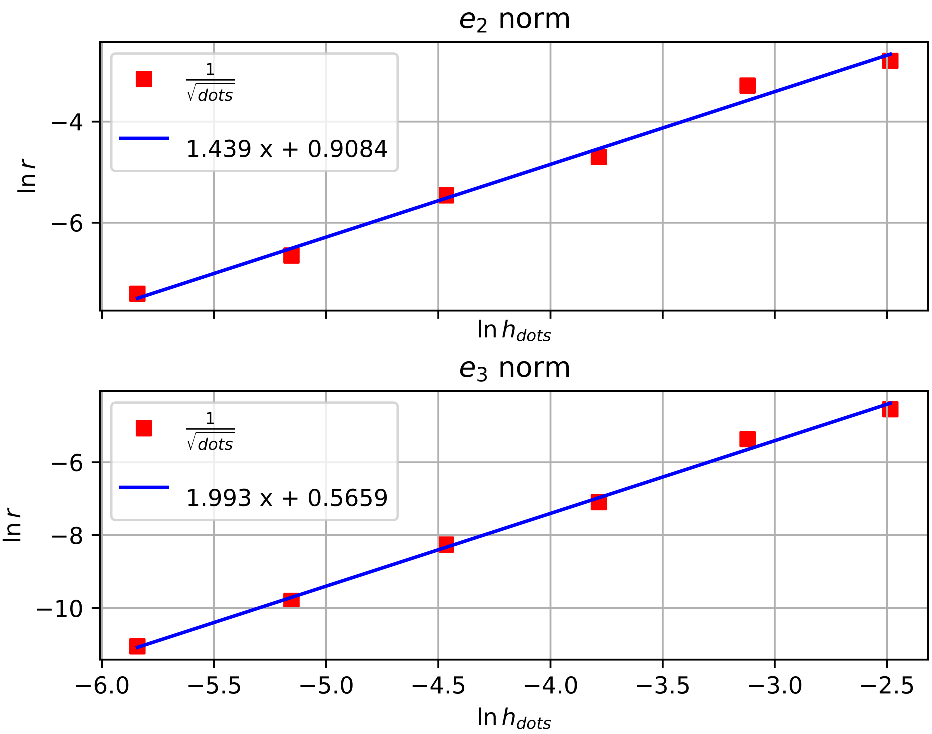

Рис. 16: График порядка сходимости численного решения для начальных условий типа «шапочка-корень». Вторая норма - порядок сходимости 1.439, третья 1.993.

Результаты всех вычисляемых норм проиллюстрированны в таблице ниже. В первом столбце в качестве шага по пространству используется $h_{dots} = 1/\sqrt{\text{dots number}}$, во втором $h_{scale} = 1/\text{scale}$

	$1/\sqrt{\text{dots number}}$	$1/\text{scale}$
e_1	0.500	0.487
e_2	1.439	1.397
e_3	1.993	1.935

Таблица 4: Таблица с результатами расчётов порядка сходимости с помощью различных типов норм и аналогов шагов по пространству для начальных условий типа «шапочка-корень».

5.5 Ступенька

Начальное условие

$$F(x,y) = \begin{cases} 1, max(|x|,|y|) \leq 0.5 \\ 0, max(|x|,|y|) > 0.5 \end{cases} \tag{7}$$

Визуализация решения

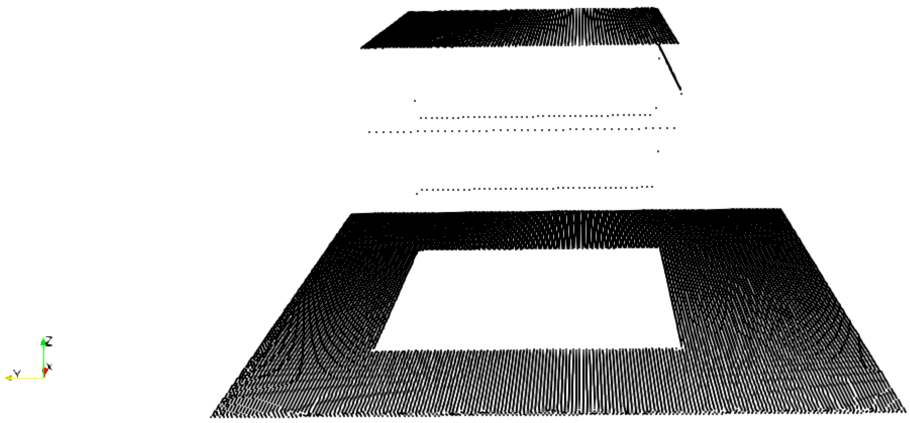

Рис. 17: Ступенька после первого шага по времени

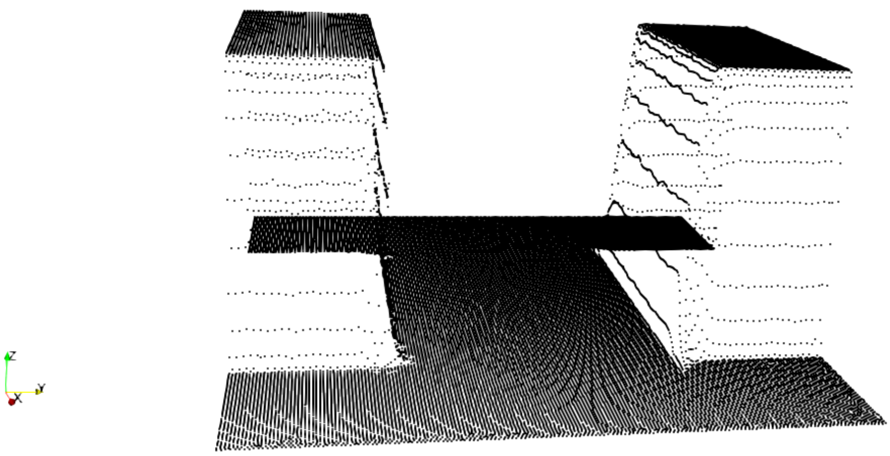

Рис. 18: Ступенька в конечный момент времени

Порядок сходимости

При использовании в качестве аналога шага по пространству $1/\sqrt{\text{dots number}}$ получены наиболие репрезентативные результаты порядка сходимости, ниже приведён график для второй и третьей норм.

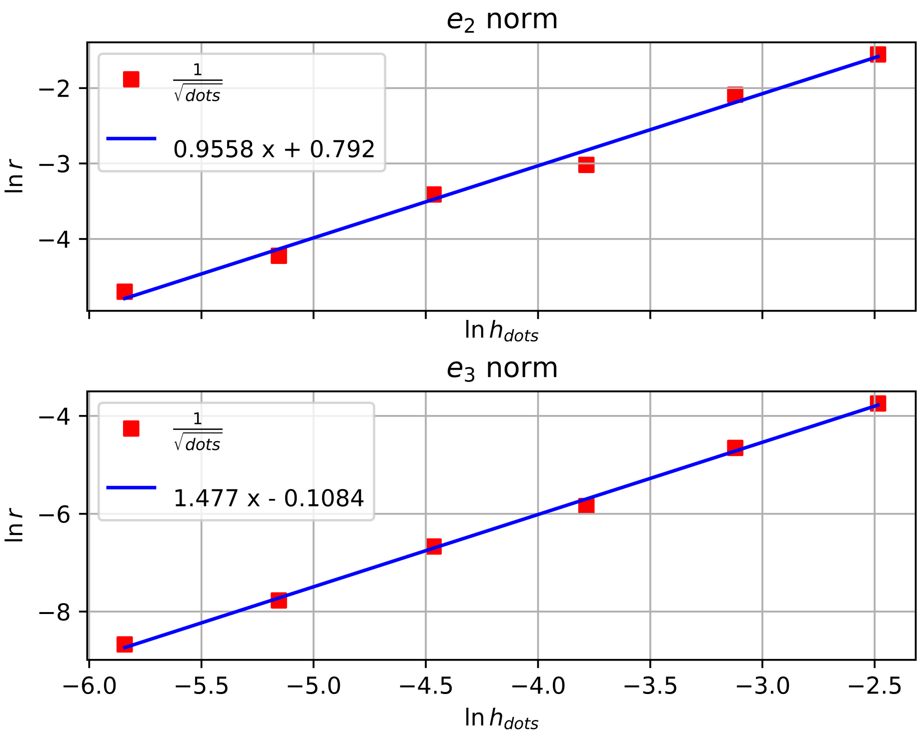

Рис. 19: График порядка сходимости численного решения для начальных условий типа «ступенька». Вторая норма - порядок сходимости 0.956, третья 1.477.

Результаты всех вычисляемых норм проиллюстрированны в таблице ниже. В первом столбце в качестве шага по пространству используется $h_{dots} = 1/\sqrt{\text{dots number}}$, во втором $h_{scale} = 1/\text{scale}$

	$1/\sqrt{\text{dots number}}$	$1/\text{scale}$
e_1	-0.003	-0.002
e_2	0.956	0.929
e_3	1.477	1.435

Таблица 5: Таблица с результатами расчётов порядка сходимости с помощью различных типов норм и аналогов шагов по пространству для начальных условий типа «ступенька».

6 Анализ результатов и вывод

smth

7 Список литературы

smth