

Распознавание графиков решения одномерного линейного уравнения переноса

Выполнил: Климов О. Д., ФН2–61Б
под руководством Галанина М. П.

18 июня 2024 г.

Формулировка

- 1 Необходимо изучить методы численного решения линейного одномерного уравнения переноса. Составить и отладить программу для нахождения численного решения задачи Коши для указанного уравнения. Использовать шесть различных разностных схем. Для всех тестов использовать одинаковую систему тестов.
- 2 Реализовать модель нейронной сети для распознавания решения уравнения переноса на языке программирования Python.
- 3 На основе модели разработать программу, которая по неточному решению возвращает более точное известное решение.

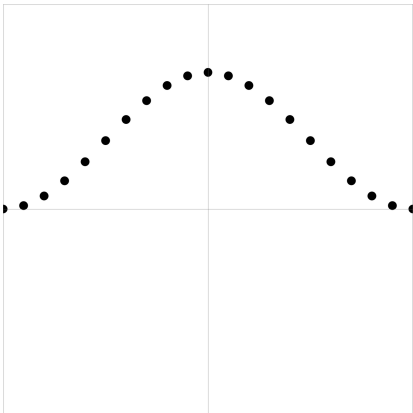


Рис. 1: Численное решение

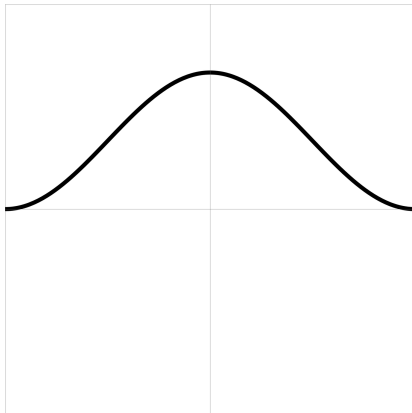


Рис. 2: Улучшенное решение

Задача Коши для уравнения переноса

$$\begin{cases} \frac{\partial u}{\partial t} + a \frac{\partial u}{\partial x} = 0, \\ u(x, 0) = u_0(x), \end{cases} \quad \text{где } u = u(x, t), \quad a = \text{const} > 0, \quad t \in (0, T), \quad x \in (-\infty, \infty) \quad (1)$$

Решение имеет вид $u = u_0(x - at)$ и заключается в сносe неизменного профиля по характеристикам. То есть начальный профиль переносится без искажений с заданной скоростью a .

Численное решение уравнения переноса

Целью построения методов численного решения данного уравнения является не собственно нахождение его решения, а исследование численного алгоритма на простейшем примере. Аналитическое решение данного уравнения известно и тривиально. При этом в чрезвычайно большое количество математических моделей уравнение переноса входит в качестве составной части.

Разностные схемы в той или иной мере искажают точное решение, например схемы первого порядка, как правило, дают решение «расплывающееся» со временем по пространству.

Для решения поставленной задачи было рассмотрено 6 схем.

Явная схема с левой разностью по 2-м точкам

$$\hat{y} = (1 - \gamma)y + \gamma y_{-1}.$$

Погрешность аппроксимации схемы: $\psi_h = O(\tau + h)$.

Данная схема является безусловно устойчивой.

Неявная схема с левой разностью по 2-м точкам

$$\hat{y} = \frac{\gamma}{1 + \gamma} \hat{y}_{-1} + \frac{1}{1 + \gamma} y.$$

Погрешность аппроксимации схемы: $\psi_h = O(\tau + h)$.

Данная схема является безусловно устойчивой.

Явная схема с левой разностью по 3-м точкам

$$\hat{y} = (1 - \frac{3}{2}\gamma)y + \gamma(2y_{-1} - \frac{1}{2}y_{-2}).$$

Погрешность аппроксимации схемы: $\psi_h = O(\tau + h^2)$.

Схема является безусловно неустойчивой.

Неявная схема с левой разностью по 3-м точкам

$$\hat{y} = \frac{2}{2+3\gamma}y + \frac{\gamma}{2+3\gamma}(4\hat{y}_{-1} - \hat{y}_{-2}).$$

Погрешность аппроксимации схемы: $\psi_h = O(\tau + h)$.

Схема является безусловно неустойчивой.

Схема Лакса

$$\hat{y} = \frac{(y_{+1} + y_{-1}) - \gamma(y_{+1} - y_{-1})}{2}.$$

Погрешность аппроксимации схемы: $\psi_h = O(\tau + h + \frac{h^2}{\tau})$.

Схема устойчива при $\gamma \leq 1$ и сходится $h^2 \rightarrow 0$ быстрее, чем τ .

Схема Лакса-Вендрофа

$$\hat{y} = y - \gamma \left(\frac{(y_{+1} + y) - \gamma(y_{+1} - y)}{2} - \frac{(y + y_{-1}) - \gamma(y - y_{-1})}{2} \right).$$

Погрешность аппроксимации схемы: $\psi_h = O(\tau^2 + h^2)$.

Схема устойчива при $\gamma \leq 1$, а при $\gamma = 1$ схема является точной.

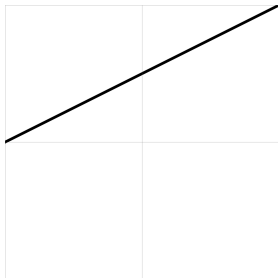


Рис. 3:
Левый треугольник,
 $u_0(x) = \frac{x-l_1}{l_2-l_1}$

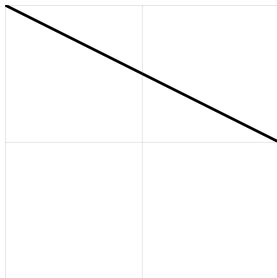


Рис. 4:
Правый треугольник,
 $u_0(x) = \frac{l_2-x}{l_2-l_1}$



Рис. 5:
Прямоугольник,
 $u_0(x) = \frac{2}{3}$

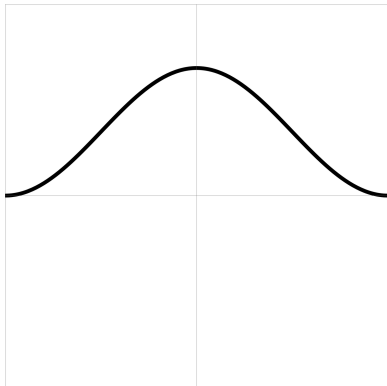


Рис. 6:
Косинус

$$u_0(x) = \frac{1}{3} \left(1 - \cos\left(\frac{2\pi(x-l_1)}{l_2-l_1}\right) \right)$$

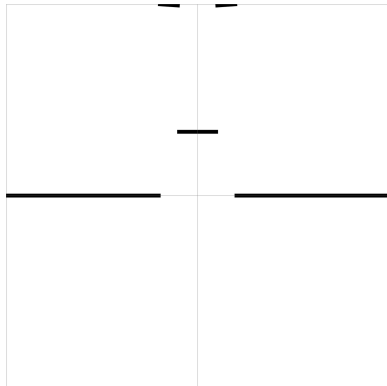


Рис. 7:
«Зуб»

$$u_0(x) = \begin{cases} -\frac{2}{3}(l_{11} - l_1)(x - l_1) + 1, & l_1 \leq x < l_{11} \\ \frac{1}{3}, & l_{11} \leq x \leq l_{22} \\ \frac{2}{3}(l_2 - l_{22})(x - l_2) + 1, & l_{22} < x \leq l_2 \end{cases}$$

Результаты реализации схем

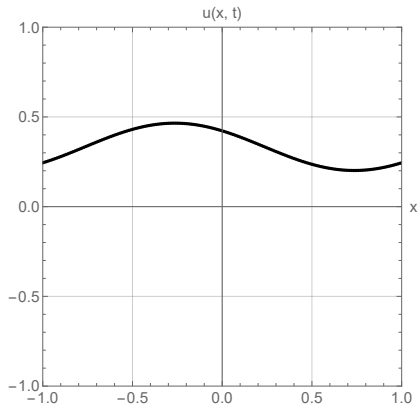


Рис. 8: Численное решение

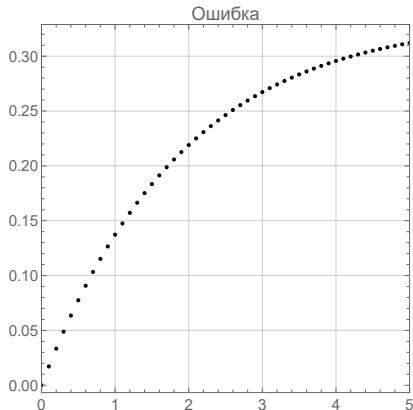


Рис. 9: Изменение нормы ошибки

Тест косинус по схеме (2), $t = 2$, при $h = 0.01, \tau = 0.1$

Результаты реализации схем

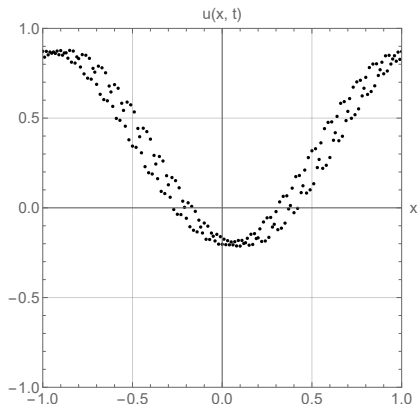


Рис. 10: Численное решение

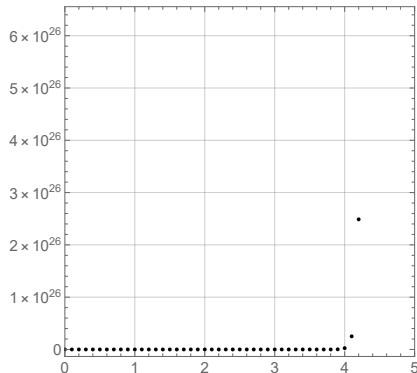


Рис. 11: Изменение нормы ошибки

Иллюстрация расходимости схемы (1) на тесте косинус, $t = 1.3$, при $h = 0.01, \tau = 0.1$

Результаты реализации схем

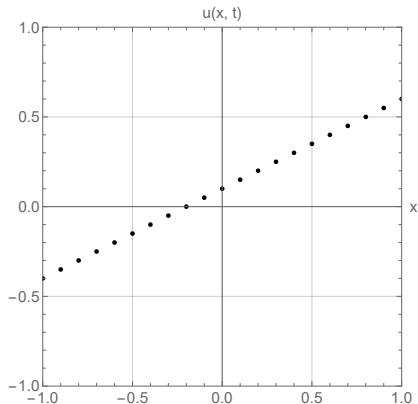


Рис. 12: Численное решение

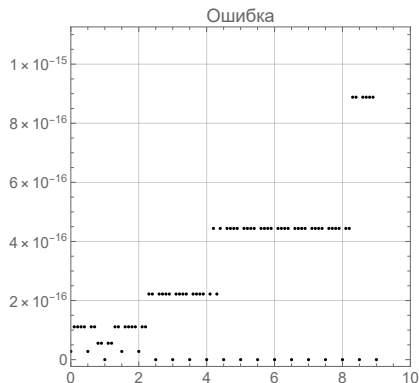


Рис. 13: Изменение нормы ошибки

Тест левый треугольник по схеме (1), $t = 1.8$ при $h = 0.1, \tau = 0.1$

Результаты реализации схем

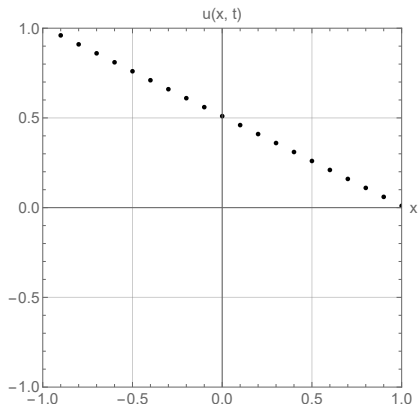


Рис. 14: Численное решение

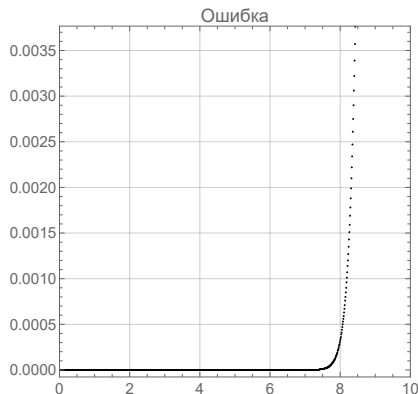


Рис. 15: Улучшенное решение

Тест правый треугольник по схеме (6), $t = 0.5$, при $h = 0.1, \tau = 0.01$

Таблица 1: Нормы ошибки разностных схем

Схема h, τ	Явная ЛДР 1-го пор.	Неявная ЛДР 1-го пор.	Явная ЛДР 2-го пор.
$h = 0.1, \tau = 0.1$	$4.44 \cdot 10^{-16}$	$4.2 \cdot 10^{-2}$	1632
$h = 0.1, \tau = 0.01$	$5.2 \cdot 10^{-2}$	$6.7 \cdot 10^{-2}$	1229

Схема h, τ	Неявная ЛДР 2-го пор.	Лакса	Лакса-Вендрофа
$h = 0.1, \tau = 0.1$	10^{18}	$9.1 \cdot 10^{-2}$	$4.46 \cdot 10^{-16}$
$h = 0.01, \tau = 0.1$	10^{54}	$3.3 \cdot 10^{32}$	$5.4 \cdot 10^{34}$

Алгоритм сверточной нейронной сети

Достоинства модели

Сверточная нейронная сеть обладает способностью улавливать пространственные взаимосвязи на изображении, то есть распознавать объекты независимо от их положения и размера на изображении. Весами(настраиваемыми параметрами) являются матрицы(фильтры), которые используется целиком для всего изображения, благодаря чему модель требует меньшее количество времени для обучения.

Алгоритм сверточной нейронной сети состоит из следующих этапов:

Этап 1: Входные данные (входной слой)

Пусть дано черно-белое изображение размером (n, n) пикселей. Изображение представляется в виде матрицы A той же размерности, а каждый элемент принимает значение яркости соответствующего пикселя.

Алгоритм сверточной нейронной сети

Этап 2: Извлечение признаков (Слой свертки)

Выбирается матрица F , являющаяся по своей сути фильтром.
Производится операция свертки для входной матрицы A .
Получаем новую матрицу $M = A * F$

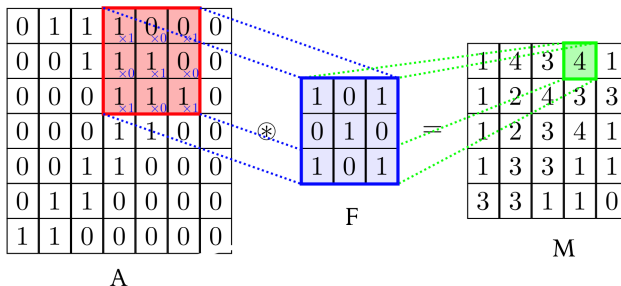


Рис. 16: Иллюстрация операции свертки

Алгоритм сверточной нейронной сети

Этап 3: Отбор признаков (Слой пуллинга)

Разбиваем матрицу M на подматрицы равной размерности и заменяем ее наибольшими элементами соответствующих подматриц. Данный этап позволяет уменьшить размерность последующих матриц, получаемых после операции свертки. Это приводит к уменьшению количества параметров в сети и ускорению обучения, а также уменьшает эффект переобучаемости нейросети.

Этап 4: Классификация (полносвязный слой)

Матрица M преобразовывается в вектор, путем последовательного соединения строк, после чего осуществляем алгоритм перцептрона.

Задача распознавания решения

Формулировка

Пусть дано изображение графика решения $u(x, t)$ в определенной области. Пусть это изображение имеет искажения, в нашем случае — погрешность решения. Основываясь на нем, необходимо с определенной точностью определить вид функции $u(x, t)$.

Ограничения

В силу конечности множества вариантов ответа нейронной сети, введем ограничения задачи:

- 1 Рассмотрим решения, относящиеся исключительно к рассмотренным тестам
- 2 Будем рассматривать решения на определенной сетке до некоторого T
- 3 Будем рассматривать изображения определенной размерности

Краткий обзор библиотек машинного обучения

Библиотека TensorFlow

Достоинства

- 1 Высокая производительность
- 2 Статический граф вычислений
- 3 Наиболее полный набор инструментов и функций
- 4 Представление данных в «тензорах», что обеспечивает высокую степень абстракции для операций.

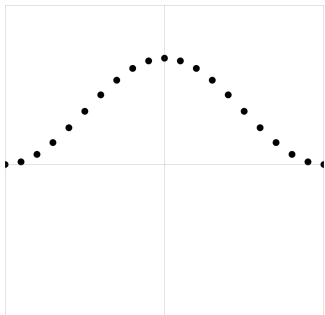
Недостатки

- 1 Сложность изучения всех возможностей библиотеки, в связи отсутствием русскоязычной литературы
- 2 Большой объем файлов и их зависимостей, из-за чего библиотека часто «ломается»

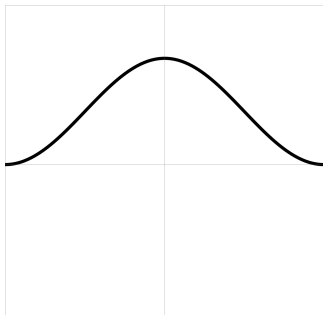
Программа улучшения решения уравнения переноса

Идея алгоритма

Чтобы распознать на изображении определенное решение уравнения, необходимо произвести расчет модели нейронной сети и получить ее ответ. Этим ответом будет метка, которая была назначена во время обучения каждому решению. Зная метку, можно поставить в соответствие исходное для обучения изображение, которое и будет являться улучшенным решением.



→ (Label №1) →



Результаты работы программы улучшения решения

Таблица 2: Результаты точности работы программы

№	Шаг по времени, τ	Шаг по пространству, h	Точность, %
1	0.1	0.1	82
2	0.05	0.1	84
3	0.05	0.05	89
4	0.01	0.01	93

Стоит отметить, что результаты могут отличаться в зависимости от множества факторов, к которым относятся выбор коэффициентов сети, количество эпох обучения и особенности набора данных для обучения.

Таким образом в ходе работы

- 1 были изучены численные методы решения уравнения переноса, проведено тестирование численных методов на 5 тестах, а также созданы несколько программ.
- 2 Была создана программа, которая по неточному решению уравнения переноса возвращает точное. Была реализована программа на языке программирования C++17 для численного решения уравнения переноса различными схемами, а также реализованы 5 тестов.
- 3 Реализована модель сверточной нейронной сети для распознавания решений уравнения на языке программирования Python 3.9.2 с помощью библиотеки машинного обучения TensorFlow 2.15.0.

Как итог, реализована программа, которая с помощью модели сети улучшает неточное решение данного уравнения до точного.