Московский авиационный институт
(Национальный исследовательский университет)

Факультет прикладной математики и физики

Кафедра вычислительной математики и программирования

Лабораторная работа № 5

по курсу «Численные методы»

Студент: Гаврилов М.С.

Группа: 80-406б

Вариант: 7

Москва, 2022

Постановка задачи

Используя явную и неявную конечно-разностные схемы, а также схему Кранка - Николсона, решить начально-краевую задачу для дифференциального уравнения параболического типа. 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 $U(x,t)$. Исследовать зависимость погрешности от сеточных параметров τ, h.

Вариант 7.

$\dfrac{\partial u}{\partial t} = \dfrac{\partial^2 u}{\partial x^2} + 0{,}5\exp(-0{,}5t)\cos x$,

$u_x(0,t) = \exp(-0{,}5t)$,

$u_x(\pi,t) = -\exp(-0{,}5t)$,

$u(x,0) = \sin x$,

Аналитическое решение: $U(x,t) = \exp(-0{,}5t)\sin x$.

Решение

Для отображения результата вычислений используется функция visualize, которая выводит на экран двумерную пространственно-временную сетку, цвет пикселей которой определяются значением соответствующих узлов сетки. Так как, для решения неявным методом используются крайне мелкие по времени сетки, для экономии ресурсов, предусмотрен вывод каждой n-й временной строки сетки.

```
def visualise(matr,t_skip,shape = [1,1]):
    pylab.figure(figsize = (shape[0],shape[1]))
    pylab.xlabel("x")
    pylab.ylabel("t")
    for i in range(matr.shape[0]):
        if(i % t_skip != 0):
            continue

        for j in range(matr.shape[1]):

            if(matr[i][j] > 0):
                pylab.plot(j,i,'s',color = (matr[i][j],0,0))
            if(matr[i][j] <= 0):
                pylab.plot(j,i,'s',color = (0,0,-matr[i][j]))
    pylab.show()
```

Уравнение:

```python
def f_x(crd):
    t,x = crd
    return 0.5*np.exp(-0.5*t)*np.sin(x)

def f_0(x):
    return np.sin(x)

def x_0(t): # > 0 --> энергия утекает, < 0 --> энергия притекает
    return np.exp(-0.5*t)

def x_1(t): # < 0 --> энергия утекает, > 0 --> энергия притекает
    return -np.exp(-0.5*t)
```

Обоснование смены f(x).

Видно, что $\int_0^{\pi} 0.5e^{-0.5t}\cos(x)\,dx = 0$. Следовательно, $\int_0^{\infty}\int_0^{\pi} 0.5e^{-0.5t}\cos(x)\,dx\,dt = 0$, а значит, f(x) не меняет суммарную энергию системы. Изначально в системе находится $\int_0^{\pi}\sin(x)\,dx = 2$ энергии. Через границы за все время утекает 2*$\int_0^{\infty} e^{-0.5t}dt = 2 * 2 = 4$, а значит, система, представленная в условии, должна прийти в равновесие с суммарной энергией = -2. Однако, аналитическое решение $U(x,t) = e^{-0.5t}\sin(x)$ на $t \to \infty$ стремится к 0. Также функция f(x) нарушает симметрию системы относительно $\frac{\pi}{2}$, которая присутствует в аналитическом решении. Подставим аналитическое решение в основное уравнение.

$u = e^{-0.5t}\sin(x)$

$u_x = e^{-0.5t}\cos(x)$

$u_{xx} = -e^{-0.5t}\sin(x)$
$u_t = -0.5e^{-0.5t}\sin(x)$

$u_t = u_{xx} + f$

$f = u_t - u_{xx}$

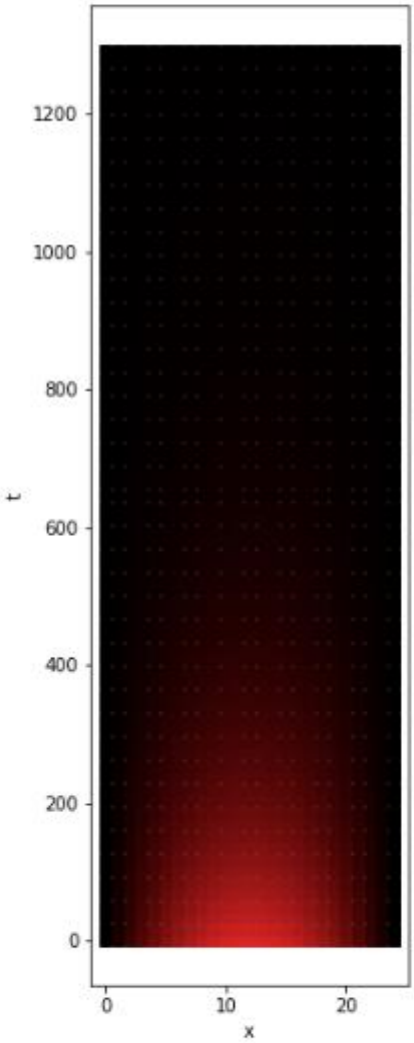
$f = -0.5e^{-0.5t}\sin(x) + e^{-0.5t}\sin(x) = 0.5e^{-0.5t}\sin(x)$

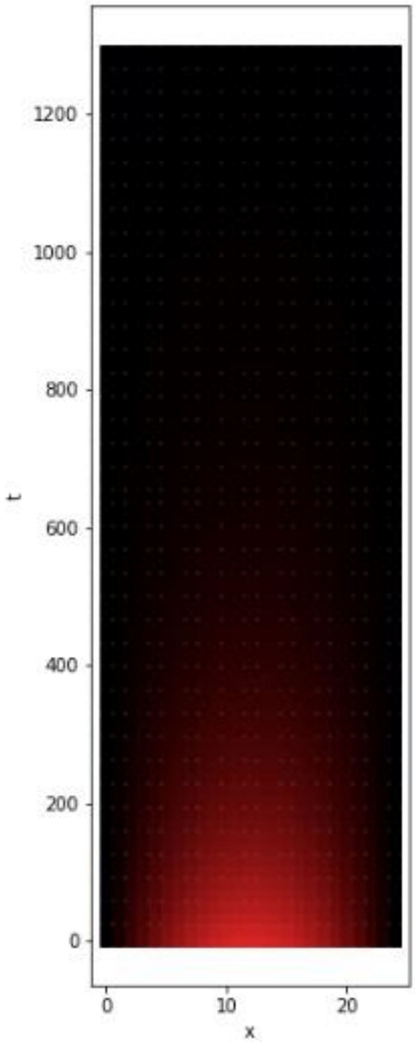
$f = 0.5e^{-0.5t}\sin(x) \neq 0.5e^{-0.5t}\cos(x)$

Можно убедиться, что при $f = 0.5e^{-0.5t}\cos(x)$ уравнение не сходится.
Чтобы $U(x,t) = e^{-0.5t}\sin(x)$ было аналитическим решением, нужно, чтобы $f = 0.5e^{-0.5t}sin(x)$.

Ожидаемо, $\int_0^{\infty}\int_0^{\pi} 0.5e^{-0.5t}\sin(x)\,dx\,dt = 2$, т.е. замена cos(x) на sin(x) балансирует энергию системы, а также делает ее симметричной относительно $\frac{\pi}{2}$.

Аналитическое решение: Решение уравнение с sin(x) явным методом:



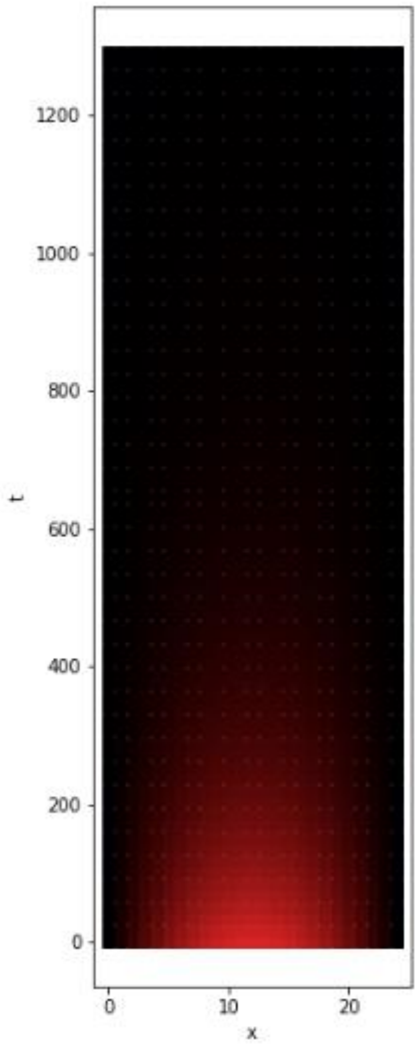


Погрешность (средняя абсолютная ошибка): `0.004600966942491745`

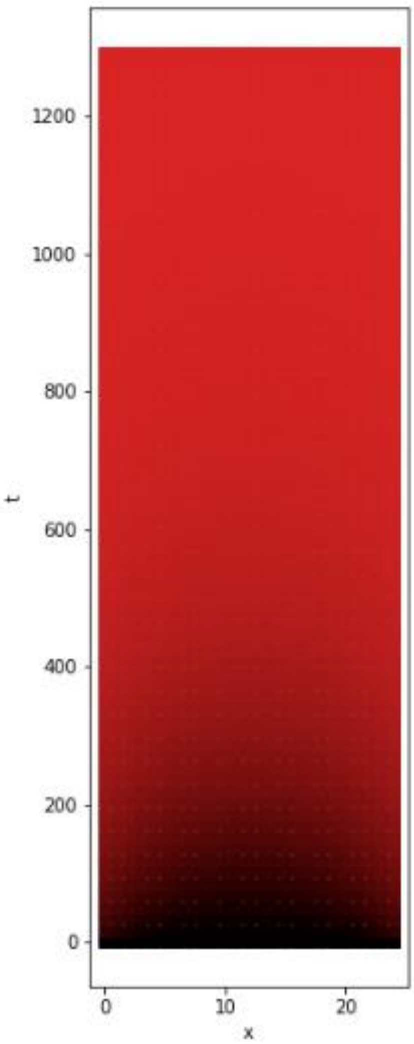
1. Двухточечная аппроксимация первого порядка точности.

Явный метод

Решение:



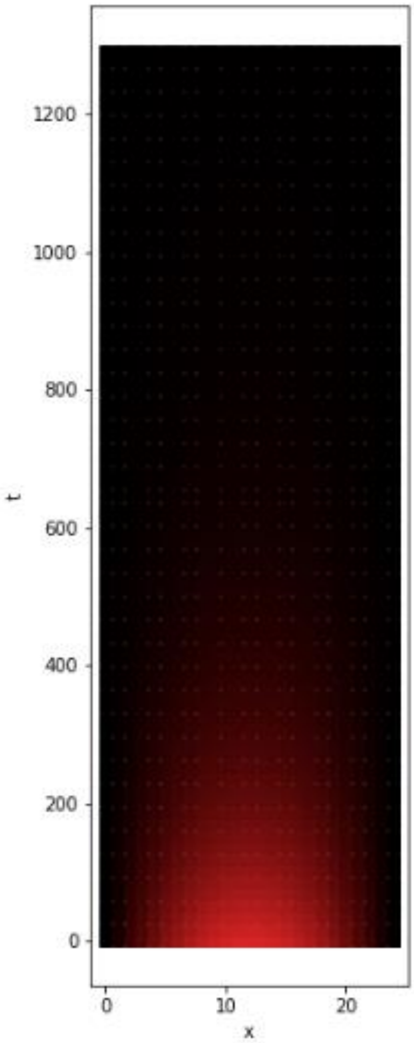
Карта погрешности, нормированная по наибольшему значению:



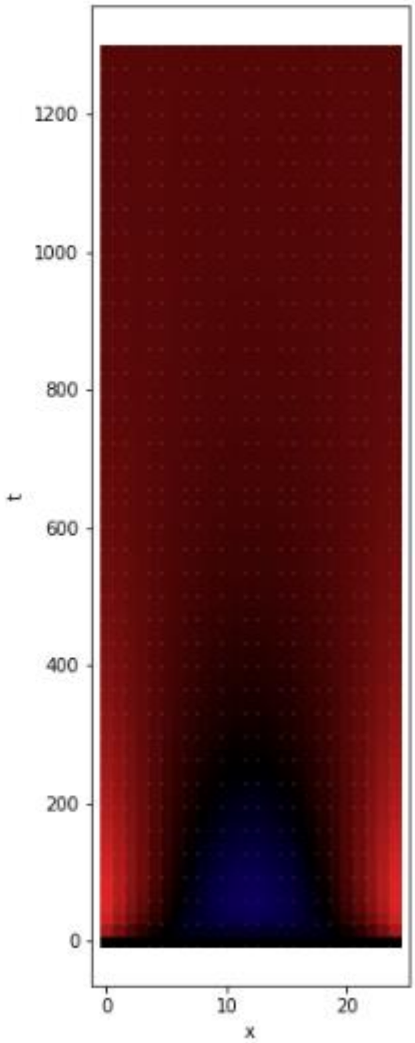
Средняя абсолютная ошибка: `0.004600966942491745`

Неявный метод

Решение:



Карта погрешности, нормированная по наибольшему значению:

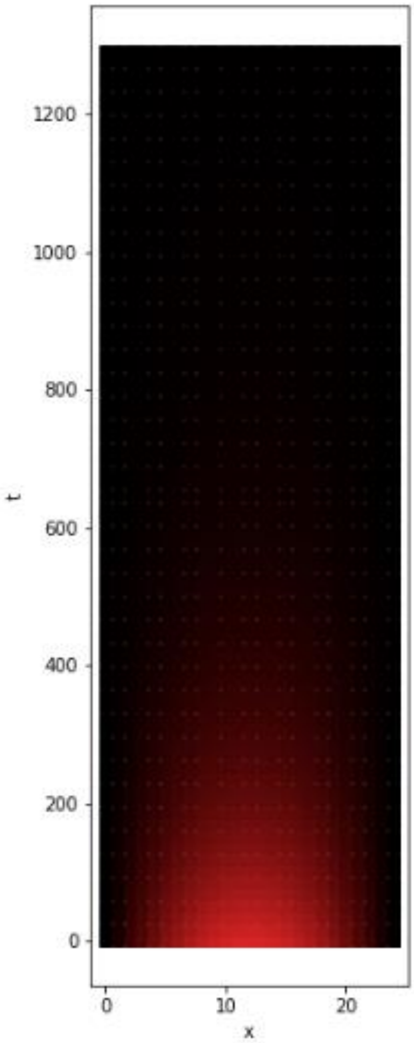


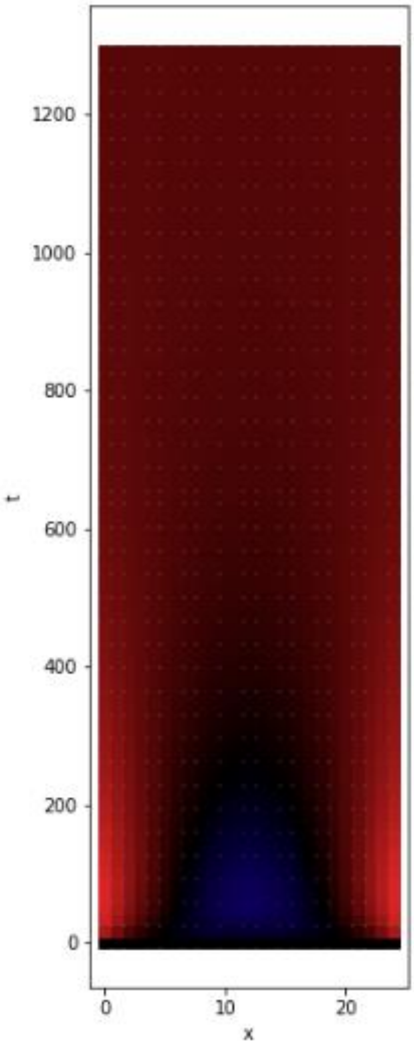
Средняя абсолютная ошибка: `0.0005593477678496438`

Комбинированный метод

Решение:

Карта погрешности, нормированная по наибольшему значению:



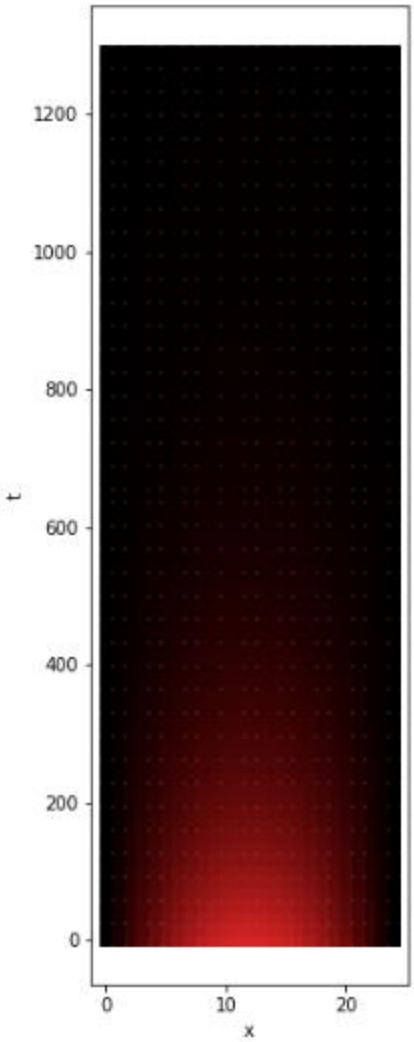


Средняя абсолютная ошибка: `0.0005559805094610557`

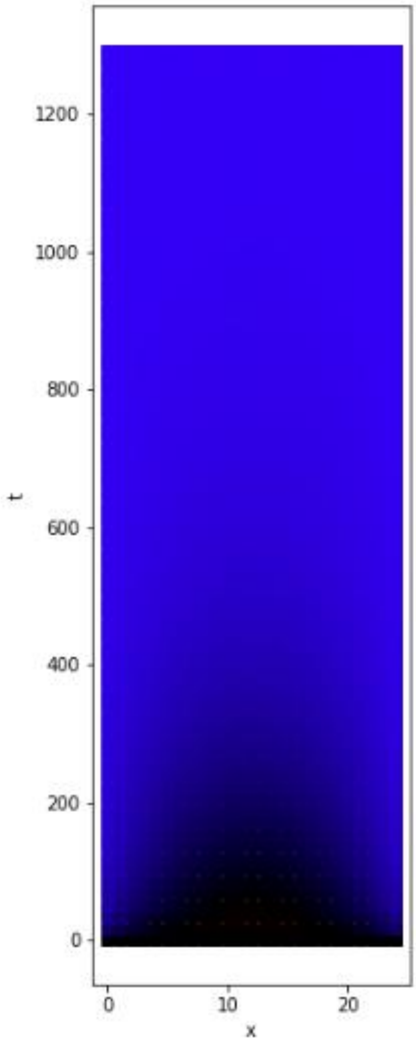
2. Трехточечная аппроксимация второго порядка точности.

Явный метод

Решение:



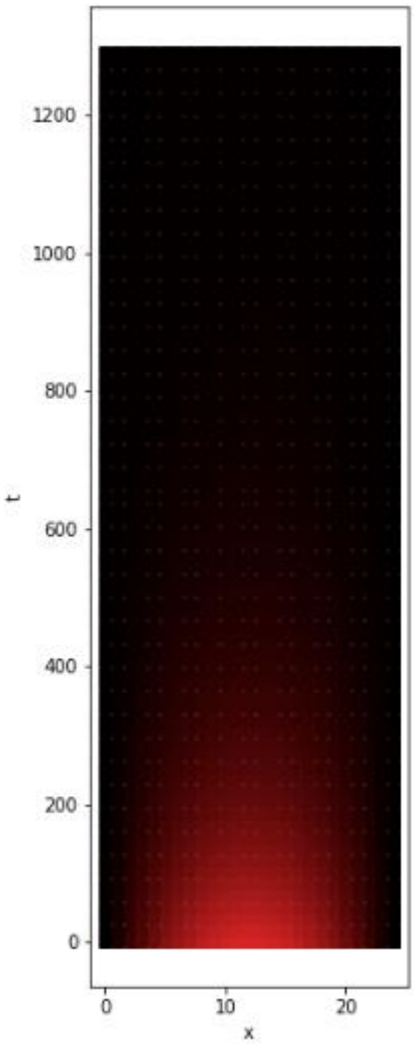
Карта погрешности, нормированная по наибольшему значению:



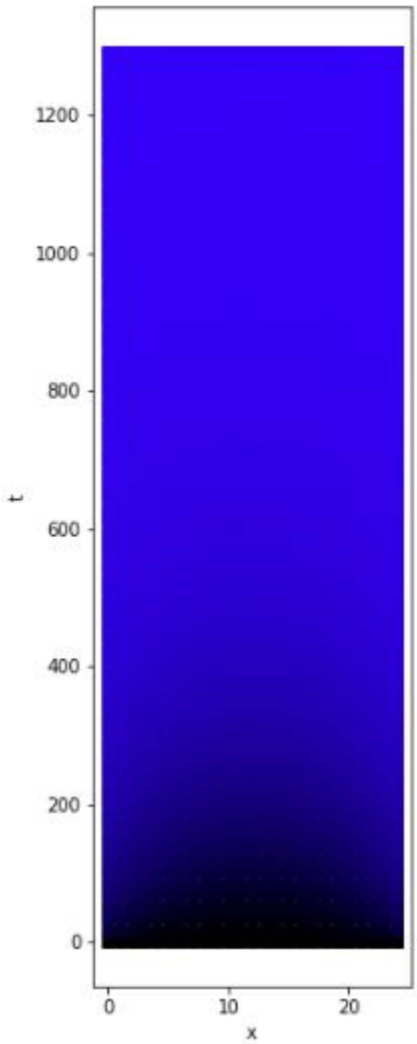
Средняя абсолютная ошибка: `0.004311436430295228`

Неявный метод

Решение:



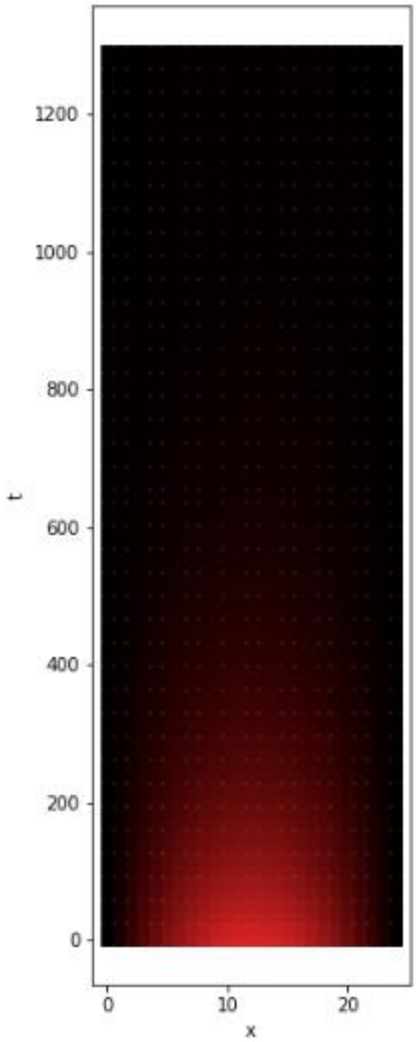
Карта погрешности, нормированная по наибольшему значению:



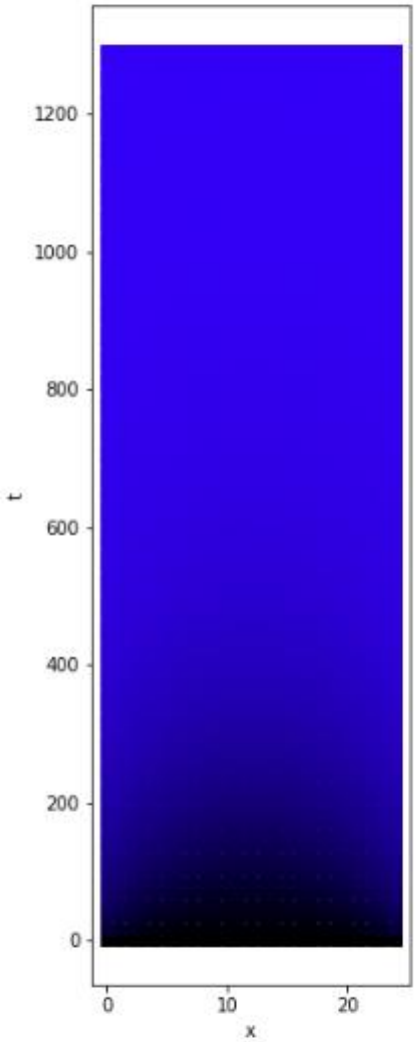
Средняя абсолютная ошибка: `0.008258104048033461`

Комбинированный метод

Решение:



Карта погрешности, нормированная по наибольшему значению:

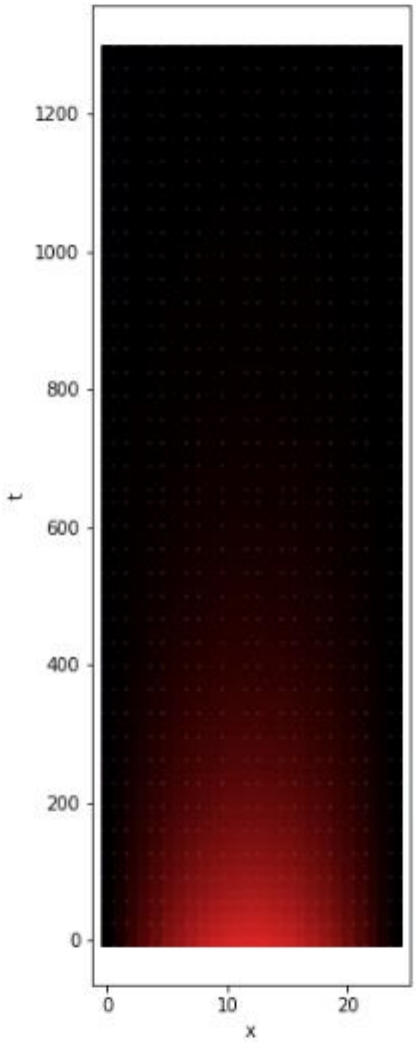


Средняя абсолютная ошибка: `0.008258166710237136`

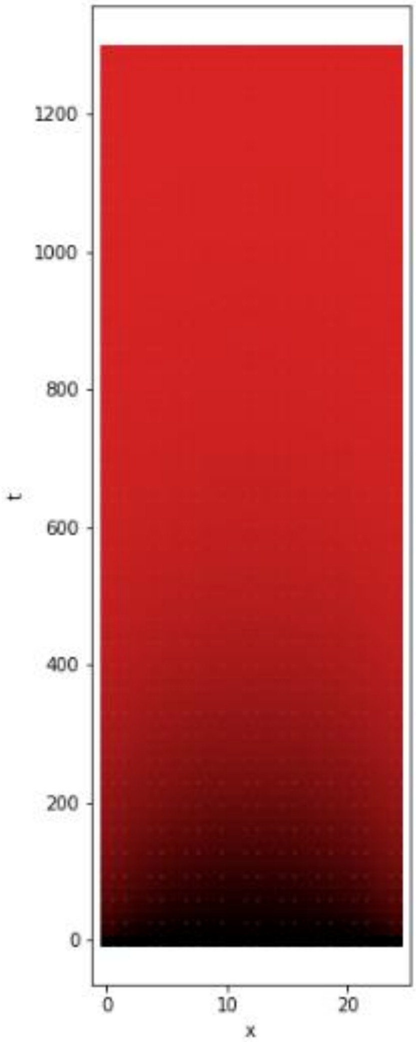
3. Двухточечная аппроксимация второго порядка точности.

Явный метод

Решение:



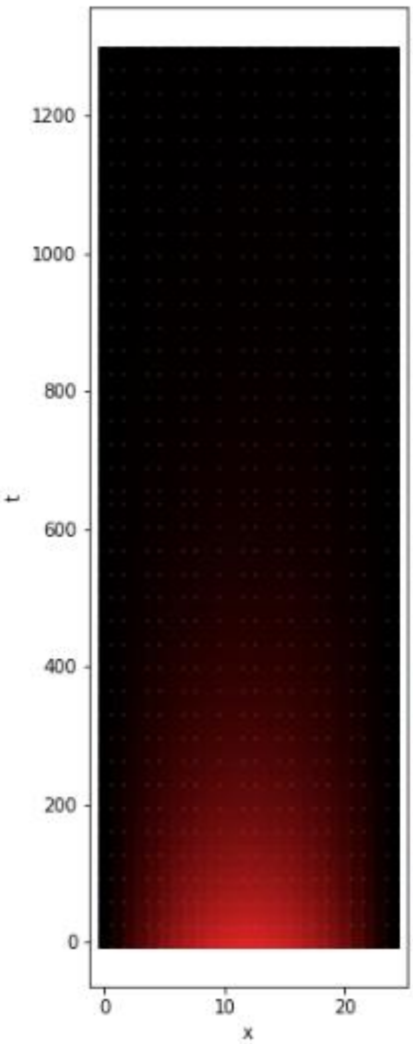
Карта погрешности, нормированная по наибольшему значению:



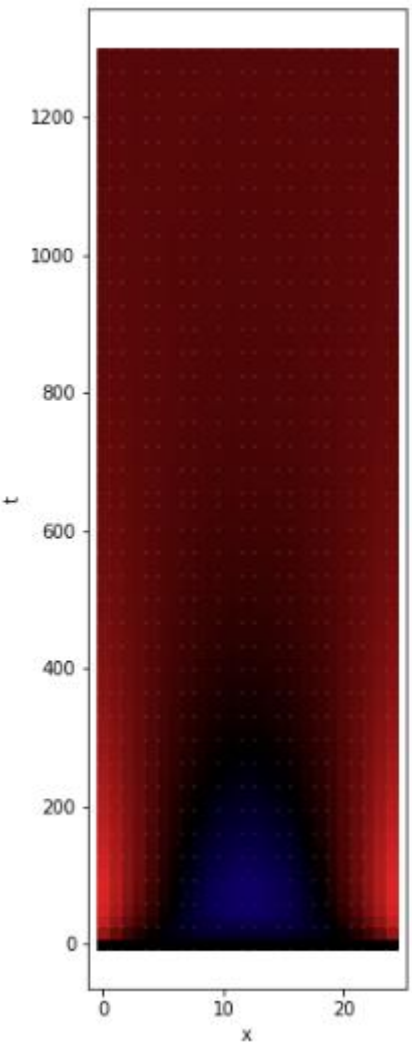
Средняя абсолютная ошибка: `0.004397394125112496`

Неявный метод

Решение:



Карта погрешности, нормированная по наибольшему значению:

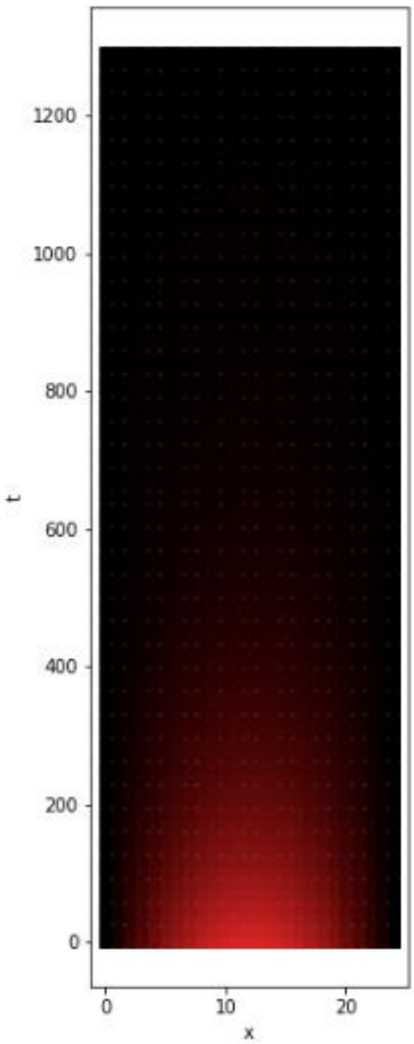


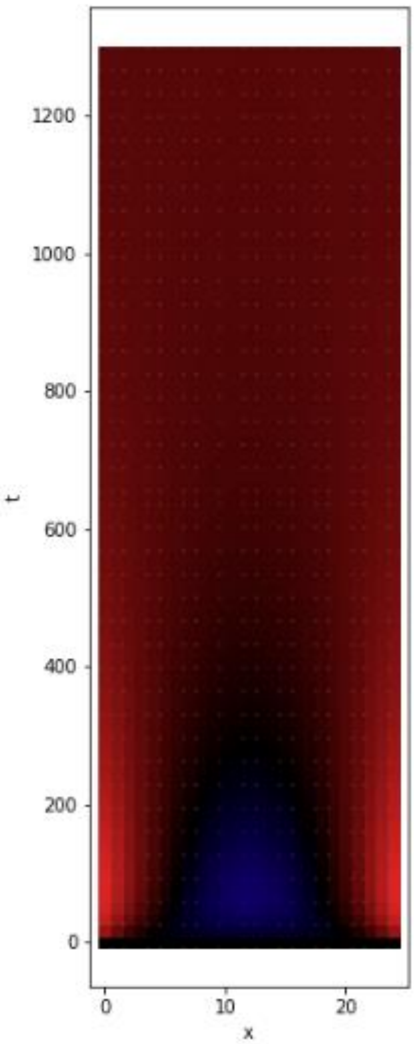
Средняя абсолютная ошибка: `0.0005307140118146265`

Комбинированный метод

Решение:

Карта погрешности, нормированная по наибольшему значению:

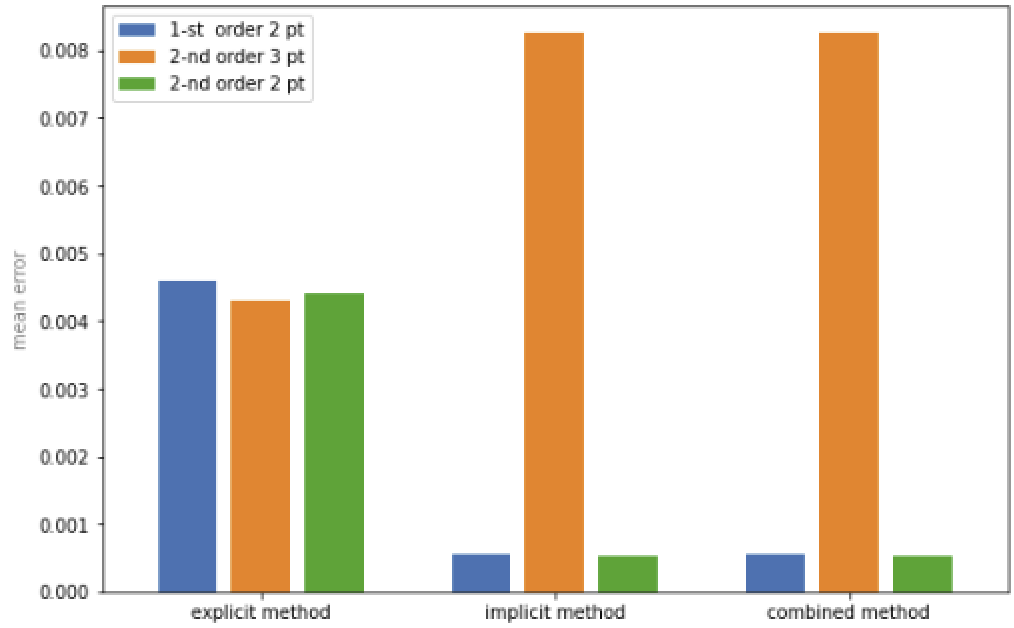




Средняя абсолютная ошибка: `0.000527593497740356`

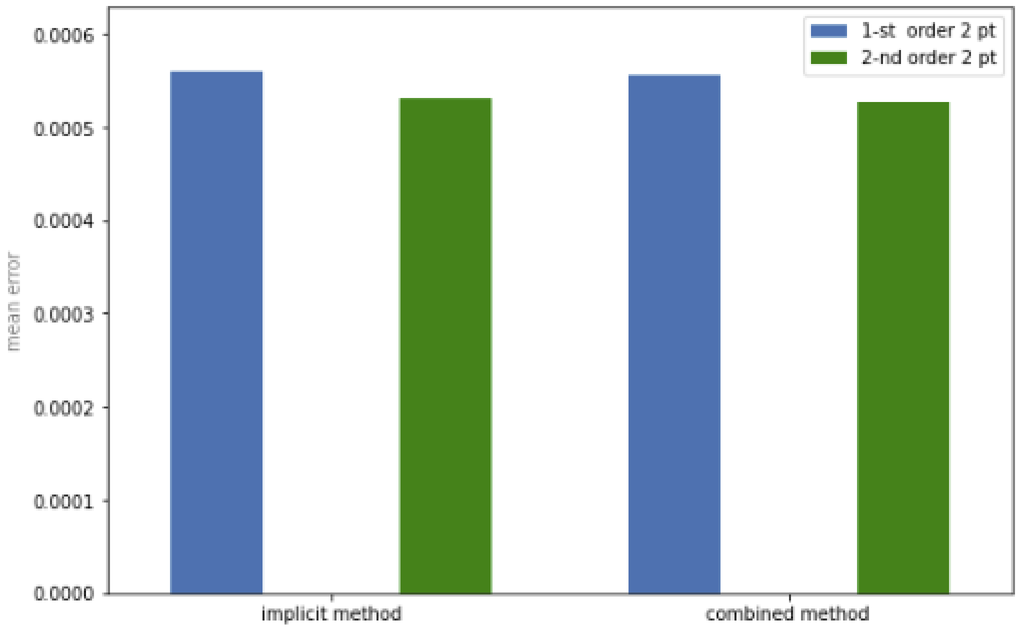
Анализ результатов

Сравнение погрешности различных методов на одинаковой сетке:



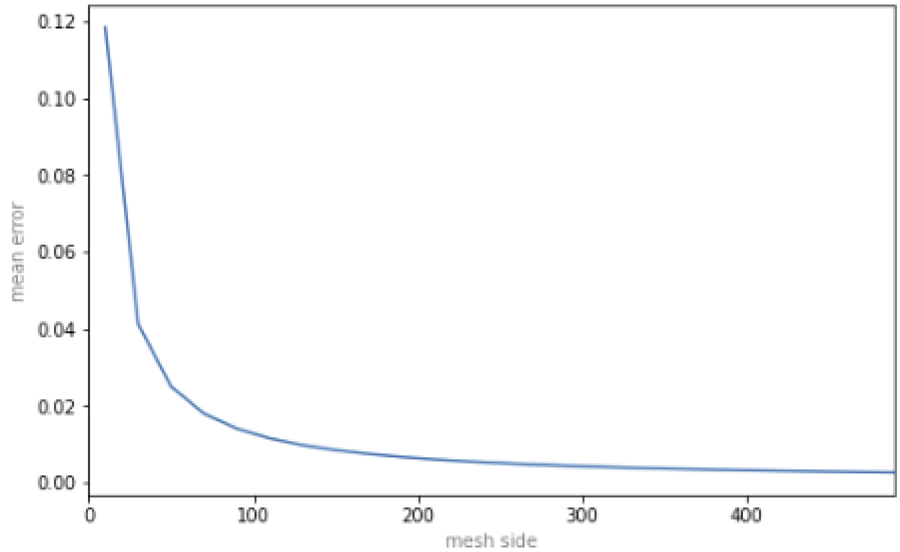
Видна аномально высокая погрешность у неявных методов при трехточечной аппроксимации. Скорее всего это ошибка в коде, но найти я ее не смог, так что, может это и действительно имеющий место эффект.

Проведем сравнение методов с наименьшей погрешностью:

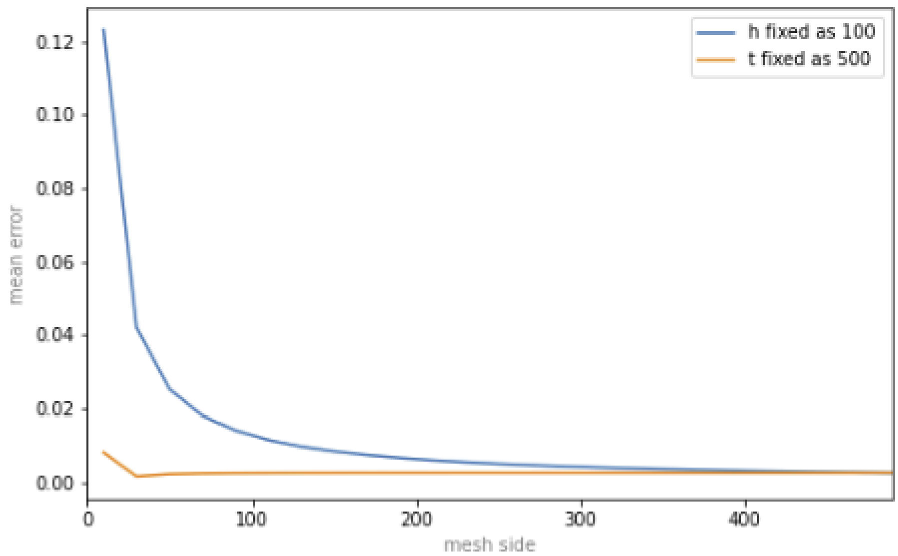


Видно, что двухточечная аппроксимация второго порядка дает прирост точности, но небольшой.

Проведем исследование зависимости погрешности от шага сетки.

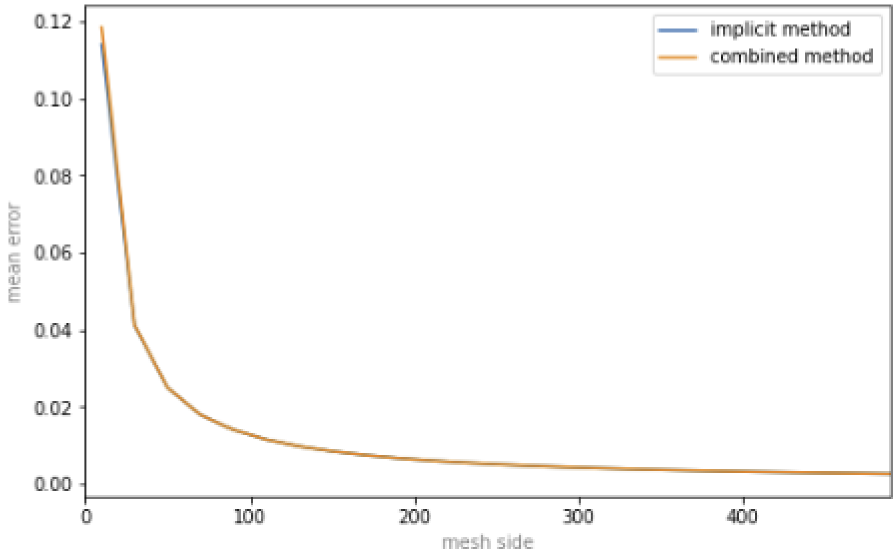


В данном случае применяется квадратная сетка со стороной mesh size и комби нированный метод с двухточечной аппроксимацией второго порядка.
Зафиксируем одну из размерностей сетки и проведем аналогичное исследован ие.



Можно сделать вывод, что, за исключением, крайне крупных сеток, высокая т очность решения достигается установлением значения t << h, а также, большо е значение t наносит куда больший вред точности решения, чем большое знач ение h.

Сравним графики зависимости погрешности от размера сетки для разных методов.



Видно, что графики имеют почти идентичную форму. Так как явный метод нельзя применить на квадратной сетке, его в сравнение добавлено не было.

Выводы

В ходе выполнения этой лабораторной работы, я ознакомился с различными схемами решения дифференциальных уравнений параб олического типа. Также я реализовал различные способы аппроксим ации производной. Созданный мной фреймворк для решения диффе ренциальных уравнений параболического типа можно улучшить, до бавив возможность решения уравнений с b и $c \neq 0$, а также упростив работу с $f = 0$, которая в нынешнем варианте реализации требует бо льшого количества конструкций if / else. При анализе результатов к ажется интересной идея построения двумерного графика зависимос ти погрешности от размерностей сетки, однако, для этого надо выпо лнить огромное количество замеров и потратить крайне много врем ени.