

Министерство образования и науки Российской Федерации  
Федеральное государственное автономное образовательное учреждение  
высшего образования

«Самарский национальный исследовательский университет  
имени академика С.П.Королёва»

Факультет информатики  
Кафедра технической кибернетики

### **Отчет по курсовой работе**

Дисциплина «Уравнения математической физики»

Тема: **«АНАЛИТИЧЕСКОЕ РЕШЕНИЕ КРАЕВЫХ ЗАДАЧ  
МАТЕМАТИЧЕСКОЙ ФИЗИКИ»**

Вариант № 25

Выполнили студенты:

Наумов М. Е.

Родин И. А.

Группа:

6409

Проверил:

Дегтярев А. А.

Самара 2018

## ЗАДАНИЕ К КУРСОВОЙ РАБОТЕ

1. Осуществить математическую постановку краевой задачи для физического процесса, описанного в предложенном варианте курсовой работы.
2. Используя метод разделения переменных (метод Фурье), получить решение краевой задачи в виде разложения в ряд Фурье по собственным функциям оператора Лапласа, соответствующим краевым условиям задачи.
3. Исследовать сходимость ряда Фурье, получить оценку остатка ряда.
4. Разработать компьютерную программу расчета функции-решения краевой задачи (суммирования ряда Фурье). При расчете коэффициентов ряда использовать метод численного интегрирования, если это необходимо. Если необходимо, то разработать специальный программный модуль для вычисления используемых собственных чисел оператора Лапласа. Компьютерная программа должна обеспечивать возможность диалогового режима ввода физических, геометрических параметров задачи, числа суммируемых элементов ряда, графическую визуализацию рассчитанного решения задачи.
5. Используя разработанную программу, провести экспериментальное исследование качества полученной аналитической оценки остатка ряда Фурье.
6. Оформить отчет о выполненной курсовой работе.

## **ВАРИАНТ 25**

Трубка длиной  $l$ , заполненная пористым поглощающим материалом, была погружена в дистиллированную воду. После того, как пористый материал оказался полностью пропитан водой, трубка была перенесена в водный раствор некоторого вещества с концентрацией  $\gamma(t)$ . В процессе диффузии вещества внутрь трубки происходит его частичное поглощение пористым материалом, имеющим коэффициент пористости  $c$ .

Скорость поглощения пропорциональна концентрации растворённого вещества (коэффициент пропорциональности равен  $D$ ).

Коэффициент диффузии  $\alpha$  известен.

Для решения описанной задачи математической физики применить метод разделения переменных.

Разработать программу расчёта динамического поля концентрации исследуемого вещества на временном промежутке  $0 < t \leq T$ . В расчётах использовать представление решения задачи в виде ряда Фурье по собственным функциям оператора Лапласа, удовлетворяющим соответствующим краевым условиям.

При проведении расчётов использовать значения параметров  $l$ ,  $D$ ,  $c$ ,  $\alpha$ ,  $T$  и выражение функции  $\gamma(t)$ , указанные преподавателем.

Указание. Концентрацию раствора по сечению трубки считать одинаковой в любой момент времени.

Значения параметров, указанные преподавателем:

$$l = 30 \text{ [см]},$$

$$T = 100 \text{ [с]},$$

$$D = 0.001,$$

$$C = 50\%,$$

$$\alpha = 0.3 \left[ \frac{\text{см}^2}{\text{с}} \right],$$

$$\phi(t) = 0.1 * \sin^2\left(\frac{\pi t}{T}\right) \left[ \frac{\text{г}}{\text{см}^3} \right].$$

## РЕФЕРАТ

Отчёт: 30 страниц, 5 рисунков, 1 таблица, 4 источника, 1 приложение.

*УРАВНЕНИЯ МАТЕМАТИЧЕСКОЙ ФИЗИКИ, КРАЕВАЯ ЗАДАЧА, УРАВНЕНИЕ ДИФФУЗИИ, МЕТОД РАЗДЕЛЕНИЯ ПЕРЕМЕННЫХ, СОБСТВЕННЫЕ ФУНКЦИИ ОПЕРАТОРА ЛАПЛАСА, РЯД ФУРЬЕ, ОЦЕНКА ОСТАТКА РЯДА.*

Целью курсовой работы является получение решения краевой задачи диффузии вещества в трубке с пористым поглощающим материалом в виде разложения в ряд Фурье по собственным функциям оператора Лапласа и создание компьютерной программы для расчета функции-решения с заданной точностью.

Для получения аналитического решения краевой задачи использован метод разделения переменных. Решение задачи получено в виде бесконечного ряда Фурье. Получена оценка остатка ряда.

Разработана компьютерная программа, обеспечивающая расчет и графическую визуализацию концентрации вещества в трубке с пористым поглощающим материалом. Для контроля погрешности усечения бесконечного ряда использована оценка остатка ряда. Приведены графические результаты численного решения задачи диффузии, а также результаты экспериментального исследования практической пригодности полученной оценки остатка ряда.

Программа написана на языке Python в текстовом редакторе Vim, в операционной системе Linux.

## СОДЕРЖАНИЕ

Введение . . . . .	6
1    Постановка краевой задачи . . . . .	7
2    Аналитическое решение краевой задачи и получение решения задачи в виде ряда Фурье . . . . .	8
3    Оценка остатка ряда (Родин И. А.) . . . . .	12
4    Результаты расчётов . . . . .	15
5    Экспериментальное исследование качества оценки остатка ряда (Наумов М. Е.)	20
Заключение . . . . .	22
Список использованных источников . . . . .	23
Приложение А Код программы . . . . .	24

## ВВЕДЕНИЕ

Метод разделения переменных относится к классу аналитических методов решения краевых задач математической физики. Характеризуя этот метод, необходимо выделить его достоинства и недостатки в сравнении с другими методами.

К достоинствам метода разделения переменных следует отнести возможность получения точного решения краевой задачи в виде ряда Фурье. Такая форма решения задачи часто и весьма успешно используется для теоретического исследования свойств этого решения. В случае достаточно быстрой сходимости ряда Фурье она может с успехом использоваться для численного моделирования физического процесса (явления).

К числу недостатков метода следует отнести его невысокую универсальность. Этот метод весьма проблематично использовать для решения нелинейных уравнений математической физики, уравнений с переменными операторными коэффициентами, а также для решения краевых задач в областях со сложными границами.

Суть метода разделения переменных состоит в факторизации по каждой независимой переменной функции, определяющей решение уравнения математической физики. Далее осуществляется переход к так называемой задаче Штурма-Лиувилля, решение которой приводит к получению собственных функций и соответствующих им собственных чисел оператора Лапласа. Затем решение исходной задачи ищется в виде ряда Фурье по этим собственным функциям.

В настоящей работе метод разделения переменных применен для получения аналитической формы решения первой краевой задачи диффузии. На основе этого результата разработан алгоритм и компьютерная программа численного моделирования процесса диффузии в тонкой трубке.

## 1 Постановка краевой задачи

Явление диффузии вещества в среде описывается известным дифференциальным уравнением в частных производных с тремя пространственными переменными. Поскольку по условию задачи концентрация вещества по сечению трубки считается одинаковой, будем использовать следующее уравнение с одной пространственной переменной:

$$C \frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2} - Du, \quad 0 \leq x \leq l, \quad 0 \leq t \leq T, \quad (1)$$

где  $u = u(x, t)$  - это концентрация вещества в трубке,  $x$  - пространственная координата (считаем, что ось  $x$  направлена вдоль трубки),  $t$  - координата времени.

Последнее слагаемое в уравнении (1) отражает поглощение вещества пористым материалом.

В начальный момент времени вещество в трубке отсутствует, следовательно начальное условие имеет вид:

$$u|_{t=0} = 0, \quad 0 \leq x \leq l. \quad (2)$$

Так как концентрация вещества в растворе равна  $\gamma(t)$ , то концентрация вещества на концах трубки в любой момент времени также будет равна  $\gamma(t)$ . Таким образом, граничные условия имеют вид:

$$u|_{x=0} = \gamma(t), \quad 0 < t \leq T, \quad (3)$$

$$u|_{x=l} = \gamma(t), \quad 0 < t \leq T. \quad (4)$$

Объединив соотношения (1)-(4) в одну систему, получим математическую модель процесса, описывающую концентрацию вещества в трубке на промежутке времени  $0 < t \leq T$  в виде:

$$\begin{cases} C \frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2} - Du, \quad 0 \leq x \leq l, \quad 0 \leq t \leq T, \\ u|_{t=0} = 0, \quad 0 \leq x \leq l, \\ u|_{x=0} = \gamma(t), \quad 0 < t \leq T, \\ u|_{x=l} = \gamma(t), \quad 0 < t \leq T. \end{cases} \quad (5)$$

## 2 Аналитическое решение краевой задачи и получение решения задачи в виде ряда Фурье

Поделим уравнение (1) на  $C$  и введём новую переменную  $a$ , такую что  $a^2 = \frac{\alpha}{C}$ , получим:

$$\frac{\partial u}{\partial t} = a^2 \frac{\partial^2 u}{\partial x^2} - \frac{D}{C} u, \quad 0 \leq x \leq l, \quad 0 \leq t \leq T. \quad (6)$$

Сведём граничные условия к нулевым, сделав следующую замену:

$$u(x, t) = v(x, t) + \gamma(t). \quad (7)$$

Подставив замену (7) в систему (5), где первое соотношение заменено на (6), получаем:

$$\begin{cases} \frac{\partial v}{\partial t} = a^2 \frac{\partial^2 v}{\partial x^2} - \frac{D}{C}(v + \gamma(t)) - \gamma'(t), & 0 \leq x \leq l, \quad 0 \leq t \leq T, \\ v|_{t=0} = -\gamma(t), & 0 \leq x \leq l, \\ v|_{x=0} = 0, & 0 < t \leq T, \\ v|_{x=l} = 0, & 0 < t \leq T. \end{cases} \quad (8)$$

Будем искать решение системы (8) в виде:

$$v(x, t) = \sum_{n=1}^{\infty} C_n(t) X_n(x). \quad (9)$$

Рассмотрим однородную задачу, полученную из (8):

$$\begin{cases} \frac{\partial v_n}{\partial t} = a^2 \frac{\partial^2 v_n}{\partial x^2}, & 0 \leq x \leq l, \quad 0 \leq t \leq T, \\ v_n|_{x=0} = 0, & 0 < t \leq T, \\ v_n|_{x=l} = 0, & 0 < t \leq T. \end{cases} \quad (10)$$

Решение задачи (10) можно записать в виде:

$$v_n(x, t) = e^{\lambda_n t} X_n(x), \quad (11)$$

где  $\lambda_n$  - некоторая константа.

Подставив (11) в (10) и сократив  $e^{\lambda_n t}$ , так как оно не равно 0, получаем:

$$\begin{cases} \lambda_n X_n(x) = a^2 X_n''(x), & 0 \leq x \leq l, \\ X_n(x)|_{x=0} = 0, \\ X_n(x)|_{x=l} = 0, \end{cases} \quad (12)$$



Система (12) задает задачу Штурма-Лиувилля с граничными условиями Дирихле, следовательно  $X_n(x)$  можно выразить как:

$$X_n(x) = \sin\left(\frac{\pi n}{l}x\right), \quad (13)$$

$$\lambda_n = \left(\frac{\pi n a}{l}\right)^2. \quad (14)$$

Подставив решение однородной задачи (13) в уравнение (9), с учётом начального условия, получаем:

$$\sum_{n=1}^{\infty} C_n(0) \sin\left(\frac{\pi n}{l}x\right) = -\gamma(0). \quad (15)$$

Таким образом  $C_n(0)$  - это коэффициенты разложения  $(-\gamma(0))$  по функциям  $\sin(\frac{\pi n}{l}x)$ . Такое разложение является возможным в силу того, что  $X_n(x)$  являются решением задачи Штурма-Лиувилля, а значит образуют ортогональный базис.

Домножим в уравнении (15) левую и правую части на  $X_k(n)$ ,  $k \neq n$ :

$$\sum_{n=1}^{\infty} C_n(0) \sin\left(\frac{\pi n}{l}x\right) \sin\left(\frac{\pi k}{l}x\right) = -\gamma(0) \sin\left(\frac{\pi k}{l}x\right). \quad (16)$$

Проинтегрируем по  $x$  левую и правую части уравнения (16), принимая во внимание то, что  $X_n(x)$  образуют ортогональный базис:

$$C_k(0) \int_0^l \sin^2\left(\frac{\pi k}{l}x\right) dx = -\gamma(0) \int_0^l \sin\left(\frac{\pi k}{l}x\right) dx. \quad (17)$$

Таким образом  $C_k(0)$  выражается как:

$$C_k(0) = -\gamma(0) \frac{\int_0^l \sin\left(\frac{\pi k}{l}x\right) dx}{\int_0^l \sin^2\left(\frac{\pi k}{l}x\right) dx}. \quad (18)$$

Вычислив интегралы, получаем:

$$C_k(0) = -2\gamma(0) \frac{(1 - (-1)^k)}{\pi k}. \quad (19)$$

Подставим (9) и (13) в первое уравнение системы (8), принимая во внимание то, что  $\lambda_n X_n(x) = a^2 X_n''(x)$ :

$$\sum_{n=1}^{\infty} C_n'(t) X_n(x) = \sum_{n=1}^{\infty} C_n(t) \lambda_n X_n(x) - \frac{D}{C} \left( \sum_{n=1}^{\infty} C_n(t) X_n(x) + \gamma(t) \right) - \gamma'(t). \quad (20)$$

Распишем  $\gamma(t)$  и  $\gamma'(t)$  в (20) в ряд по  $X_n(x)$ :

$$\begin{aligned} \sum_{n=1}^{\infty} C'_n(t) X_n(x) &= \sum_{n=1}^{\infty} C_n(t) X_n(x) \left( \lambda_n - \frac{D}{C} \right) - \\ &- 2 \sum_{n=1}^{\infty} \left( \frac{D}{C} \gamma(t) + \gamma'(t) \right) \frac{(1 - (-1)^n)}{\pi n} X_n(x). \end{aligned} \quad (21)$$

В (21) отбросим суммы и перейдём к членам рядов. Это возможно в силу того, что ряды в левой и правой частях равны, а следовательно равны и их элементы.

$$C'_n(t) X_n(x) = C_n(t) X_n(x) \left( \lambda_n - \frac{D}{C} \right) - 2 \left( \frac{D}{C} \gamma(t) + \gamma'(t) \right) \frac{(1 - (-1)^n)}{\pi n} X_n(x). \quad (22)$$

Рассмотрим два случая:  $n$  - чётное и  $n$  - нечётное.

1.  $n$  - чётное.

Подставим  $n = 2k$  в (22) и (19):

$$C'_{2k}(t) = C_{2k}(t) \left( \lambda_{2k} - \frac{D}{C} \right), \quad (23)$$

$$C_{2k}(0) = 0. \quad (24)$$

Решая уравнение (23), учитывая начальное условие (24), получаем:

$$C_{2k}(t) = 0. \quad (25)$$

2.  $n$  - нечётное.

Подставим  $n = 2k + 1$  в (22) и (19):

$$C'_{2k+1}(t) = C_{2k+1}(t) \left( \lambda_{2k+1} - \frac{D}{C} \right) - 4 \left( \frac{D}{C} \gamma(t) + \gamma'(t) \right) \frac{1}{\pi(2k+1)}, \quad (26)$$

$$C_{2k+1}(0) = -4 \frac{\gamma(0)}{\pi(2k+1)}. \quad (27)$$

Решая уравнение (26), учитывая начальное условие (27), получаем:

$$C_{2k+1}(t) = e^{g_{2k+1}t} \left( C_{2k+1}(0) - \frac{2}{\pi i(2k+1)} \left( \frac{D}{C} I_{1_{2k+1}}(t) + I_{2_{2k+1}}(t) \right) \right), \quad (28)$$

где

$$\left\{ \begin{aligned} g_{2k+1} &= - \left( \frac{(2k+1)^2 \pi^2 a^2}{l^2} + \frac{D}{C} \right), \\ I_{1_{2k+1}}(t) &= \int_0^t 0.1 \sin^2 \left( \frac{\pi}{T} \tau \right) e^{-g_{2k+1} \tau} d\tau, \\ I_{2_{2k+1}}(t) &= \int_0^t 0.2 \sin \left( \frac{\pi}{T} \tau \right) \cos \left( \frac{\pi}{T} \tau \right) \frac{\pi}{T} e^{-g_{2k+1} \tau} d\tau. \end{aligned} \right. \quad (29)$$

Очевидно, что в суммировании будут участвовать только нечётные члены ряда.

Упростим формулу (28), подставив (27) в (28), заметив при этом, что из условия следует следующее равенство:  $\gamma(0) = 0$ .

$$C_{2k+1}(t) = -e^{g_{2n+1}t} \frac{2}{\pi(2k+1)} \left( \frac{D}{C} I_{1_{2k+1}}(t) + I_{2_{2k+1}}(t) \right). \quad (30)$$

Таким образом  $v(x, t)$  можно записать следующим образом:

$$v(x, t) = \sum_{n=1}^{\infty} -e^{g_{2n+1}t} \frac{2}{\pi(2n+1)} \left( \frac{D}{C} I_{1_{2n+1}}(t) + I_{2_{2n+1}}(t) \right) \sin\left(\frac{\pi(2n+1)}{l}x\right). \quad (31)$$

Или, если перейти к  $u(x, t)$ :

$$u(x, t) = \gamma(t) + \sum_{n=1}^{\infty} -e^{g_{2n+1}t} \frac{2}{\pi(2n+1)} \left( \frac{D}{C} I_{1_{2n+1}}(t) + I_{2_{2n+1}}(t) \right) \sin\left(\frac{\pi(2n+1)}{l}x\right). \quad (32)$$

Формула (32) является аналитическим решением задачи (5).

### 3 Оценка остатка ряда (Родин И. А.)

Рассмотрим ряд из равенства (32):

$$\sum_{n=1}^{\infty} -e^{g_{2n+1}t} \frac{2}{\pi(2n+1)} \left( \frac{D}{C} I_{1_{2n+1}}(t) + I_{2_{2n+1}}(t) \right) \sin\left(\frac{\pi(2n+1)}{l}x\right), \quad (33)$$

его остаток соответственно будет равен:

$$R = \sum_{n=N+1}^{\infty} -e^{g_{2n+1}t} \frac{2}{\pi(2n+1)} \left( \frac{D}{C} I_{1_{2n+1}}(t) + I_{2_{2n+1}}(t) \right) \sin\left(\frac{\pi(2n+1)}{l}x\right), \quad (34)$$

для некоторого натурального  $N$ , равного количеству суммируемых членов ряда (33).

Так как нас интересует модуль погрешности, запишем оцениваемый ряд:

$$|R| \leq \left| \sum_{n=N+1}^{\infty} e^{g_{2n+1}t} \frac{2}{\pi(2n+1)} \left( \frac{D}{C} I_{1_{2n+1}}(t) + I_{2_{2n+1}}(t) \right) \sin\left(\frac{\pi(2n+1)}{l}x\right) \right|. \quad (35)$$

Для первоначальной оценки воспользуемся следующими двумя соотношениями:

$$\left| \sum_{n=N+1}^{\infty} f(n) \right| \leq \sum_{n=N+1}^{\infty} |f(n)|, \quad (36)$$

$$|f(n) g(n)| = |f(n)| |g(n)|. \quad (37)$$

Применив (36) и (37) к (35), получим следующую оценку:

$$|R| \leq \sum_{n=N+1}^{\infty} \left| e^{g_{2n+1}t} \frac{2}{\pi(2n+1)} \left( \frac{D}{C} I_{1_{2n+1}}(t) + I_{2_{2n+1}}(t) \right) \right| \left| \sin\left(\frac{\pi(2n+1)}{l}x\right) \right|. \quad (38)$$

Заметим, что  $|\sin(x)| \leq 1$  для любого  $x \in \mathbb{R}$ :

$$|R| \leq \sum_{n=N+1}^{\infty} \left| e^{g_{2n+1}t} \frac{2}{\pi(2n+1)} \left( \frac{D}{C} I_{1_{2n+1}}(t) + I_{2_{2n+1}}(t) \right) \right|. \quad (39)$$

Для дальнейшей оценки распишем  $I_{1_{2n+1}}(t)$  и  $I_{2_{2n+1}}(t)$ :

$$I_{1_{2n+1}}(t) = e^{-gt} \frac{-g^2 + g^2 \cos(2\xi t) - 2\xi g \sin(2\xi t) + 4\xi^2 e^{gt} - 4\xi^2}{20(g^3 + 4\xi^2 g)}, \quad (40)$$

$$I_{2_{2n+1}}(t) = \xi e^{-gt} \frac{-g \sin(2\xi t) - 2\xi \cos(2\xi t) + 2\xi e^{gt}}{10(g^2 + 4\xi^2)}, \quad (41)$$

где  $\xi = \frac{\pi}{T}$ ,  $g = g_{2n+1}$ .

Разобьём (39) на две суммы, используя неравенство (36) и (37), получив следующую оценку:

$$|R| \leq \sum_{n=N+1}^{\infty} |\mathrm{e}^{g_{2n+1}t} \frac{2}{\pi(2n+1)} \frac{D}{C} I_{1_{2n+1}}(t)| + \sum_{n=N+1}^{\infty} |\mathrm{e}^{g_{2n+1}t} \frac{2}{\pi(2n+1)} I_{2_{2n+1}}(t)|. \quad (42)$$

Подставим  $I_{1_{2n+1}}(t)$  и  $I_{2_{2n+1}}(t)$  в (42), заметив что  $|\sin(x)| \leq 1$  и  $|\cos(x)| \leq 1$  для любого  $x \in \mathbb{R}$ . После подстановки, произведём дальнейшую оценку с помощью соотношений (36) и (37):

$$|R| \leq \sum_{n=N+1}^{\infty} \frac{2}{\pi(2n+1)} \frac{D}{C} \frac{|g^2| + |g^2| + |2\xi g| + |4\xi^2 \mathrm{e}^{gt}| + |4\xi^2|}{20|g^3 + 4\xi^2 g|} + \sum_{n=N+1}^{\infty} \frac{2}{\pi(2n+1)} \xi \frac{|g| + |2\xi| + |2\xi \mathrm{e}^{gt}|}{10|g^2 + 4\xi^2|} \quad (43)$$

где  $\xi = \frac{\pi}{T}$ ,  $g = g_{2n+1}$ .

По условию задачи  $T = 100$ , следовательно  $|\xi| \leq 1$ . В числителе заменим  $\xi$  на 1, увеличив его; в знаменателе заменим  $\xi$  на 0, уменьшив его. Так же учтём что  $g \leq 0$ , раскрыв модули и заменив  $\mathrm{e}^{gt}$  в числителе на 1:

$$|R| \leq \sum_{n=N+1}^{\infty} \frac{2}{\pi(2n+1)} \frac{D}{C} \frac{2g^2 - 2g + 8}{-20g^3} + \sum_{n=N+1}^{\infty} \frac{2}{\pi(2n+1)} \frac{-g + 4}{10g^2}, \quad (44)$$

где  $g = g_{2n+1}$ .

Сделаем из двух сумм одну:

$$|R| \leq \sum_{n=N+1}^{\infty} \frac{2}{\pi(2n+1)} \left( \frac{D}{C} \frac{2g^2 - 2g + 8}{-20g^3} + \frac{-g + 4}{10g^2} \right), \quad (45)$$

где  $g = g_{2n+1}$ .

Оценим  $g_{2n+1}$  как  $(-\frac{(2n+1)^2 \pi^2 a^2}{l^2})$ :

$$|R| \leq \frac{2}{\pi} \sum_{n=N+1}^{\infty} \left( \frac{D}{C} \frac{2(2n+1)^4 \pi^4 a^4 l^2 + 2(2n+1)^2 \pi^2 a^2 l^4 + 8l^6}{20(2n+1)^7 \pi^6 a^6} + \frac{(2n+1)^2 \pi^2 a^2 l^2 + 4l^4}{10(2n+1)^5 \pi^4 a^4} \right). \quad (46)$$

Оставим слагаемые только с наибольшей степенью у  $(2n+1)$ , равной  $(-3)$  (остальные убывают значительно быстрее и их влияние на остаток слабее):

$$|R| \leq \frac{l^2}{5\pi^3 a^2} \left( \frac{D}{C} + 1 \right) \sum_{n=N+1}^{\infty} (2n+1)^{-3}. \quad (47)$$

Для оценки (47) воспользуемся следующим фактом:

$$\sum_{n=N+1}^{\infty} f(n) \leq \int_N^{\infty} f(x) dx, \quad (48)$$

если  $f(x)$  - монотонно убывающая функция.

Так как функция  $(2x + 1)^{-3}$  очевидно является монотонно убывающей, запишем следующую оценку:

$$|R| \leq \frac{l^2}{5\pi^3 a^2} \left( \frac{D}{C} + 1 \right) \int_N^{\infty} (2x + 1)^{-3} dx. \quad (49)$$

Вычислив интеграл, получаем:

$$|R| \leq \frac{l^2}{20\pi^3 a^2} \left( \frac{D}{C} + 1 \right) (2N + 1)^{-2}. \quad (50)$$

Итоговый модуль погрешности должен быть меньше некоторой заранее заданной точности  $\epsilon$ , следовательно:

$$N \geq \frac{\sqrt{\frac{l^2}{20\pi^3 a^2 \epsilon} \left( \frac{D}{C} + 1 \right)} - 1}{2}. \quad (51)$$

## 4 Результаты расчётов

Ниже приведены результаты численного моделирования физического процесса диффузии вещества внутри трубки с пористым поглощающим материалом.

На рисунке 1 изображена концентрация вещества в трубке в момент времени  $t = 0$ . Как видно из изображения, концентрация в начальный момент равна 0, как и требовалось в условии задачи.

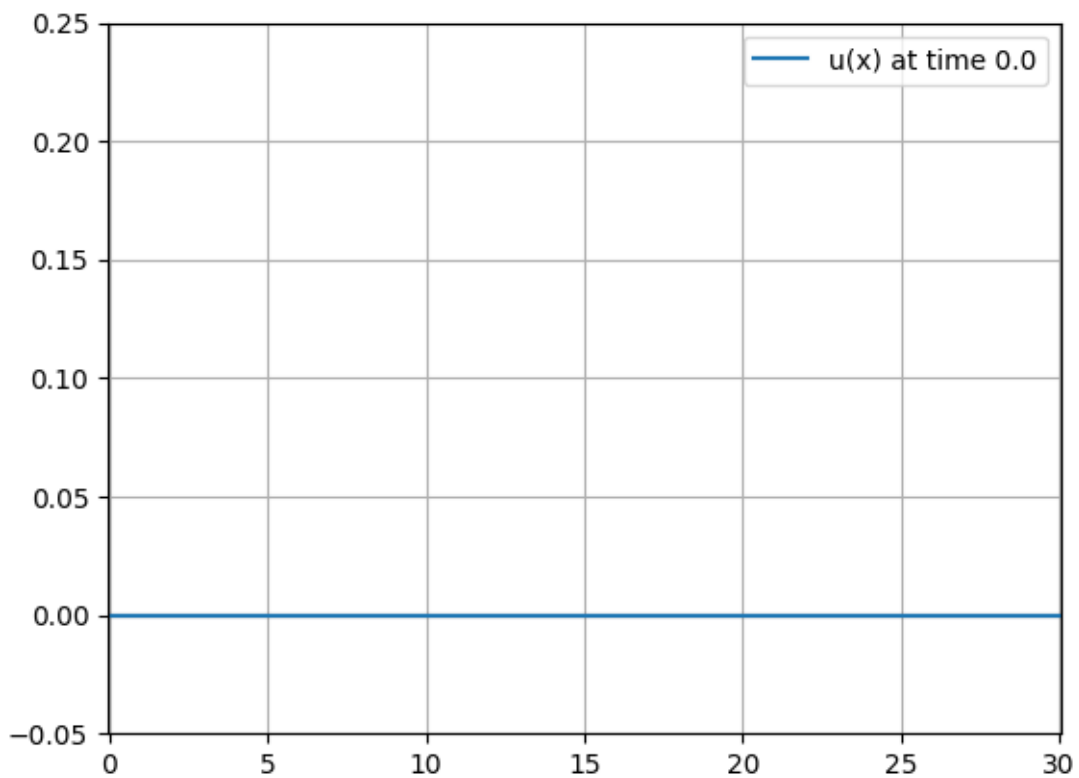


Рисунок 1 – Концентрация вещества в трубке в момент времени  $t = 0$ .

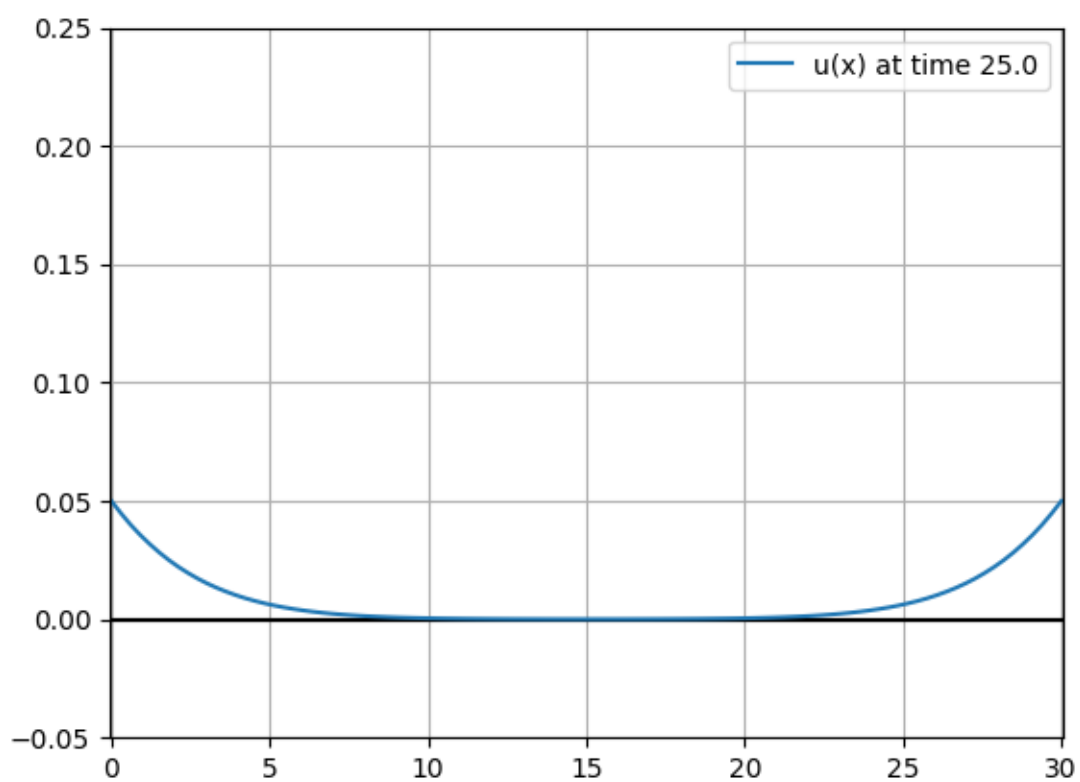


Рисунок 2 – Концентрация вещества в трубке в момент времени  $t = \frac{T}{4}$ .



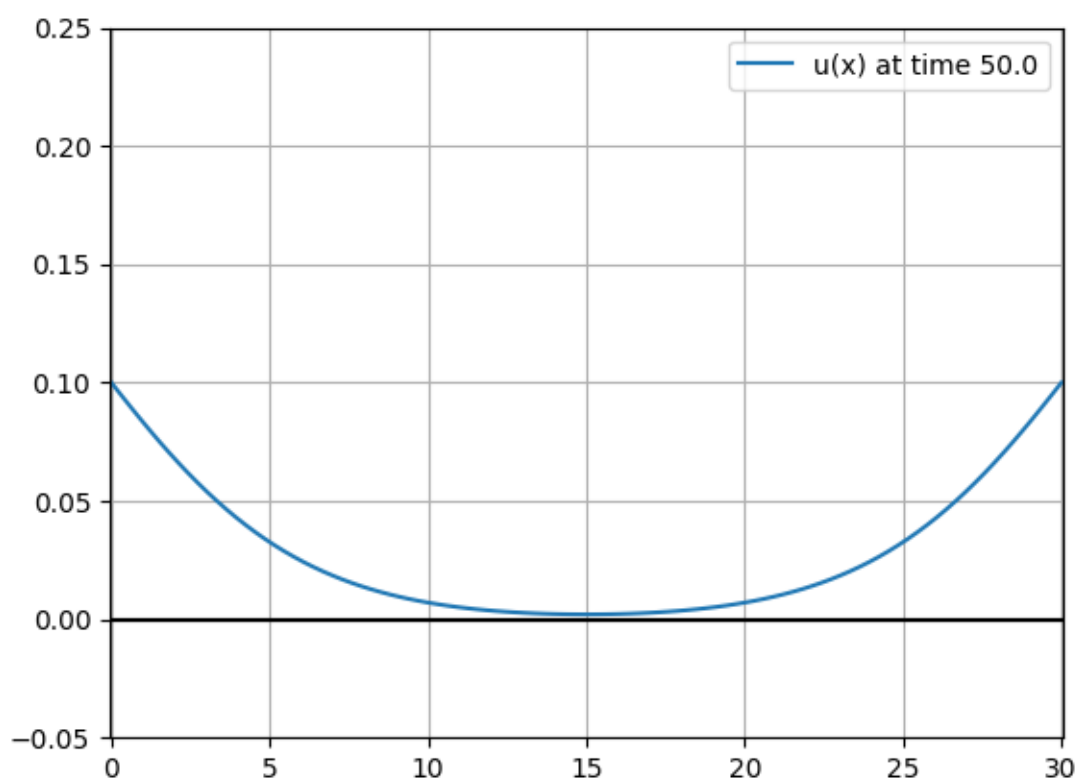


Рисунок 3 – Концентрация вещества в трубке в момент времени  $t = \frac{T}{2}$ .

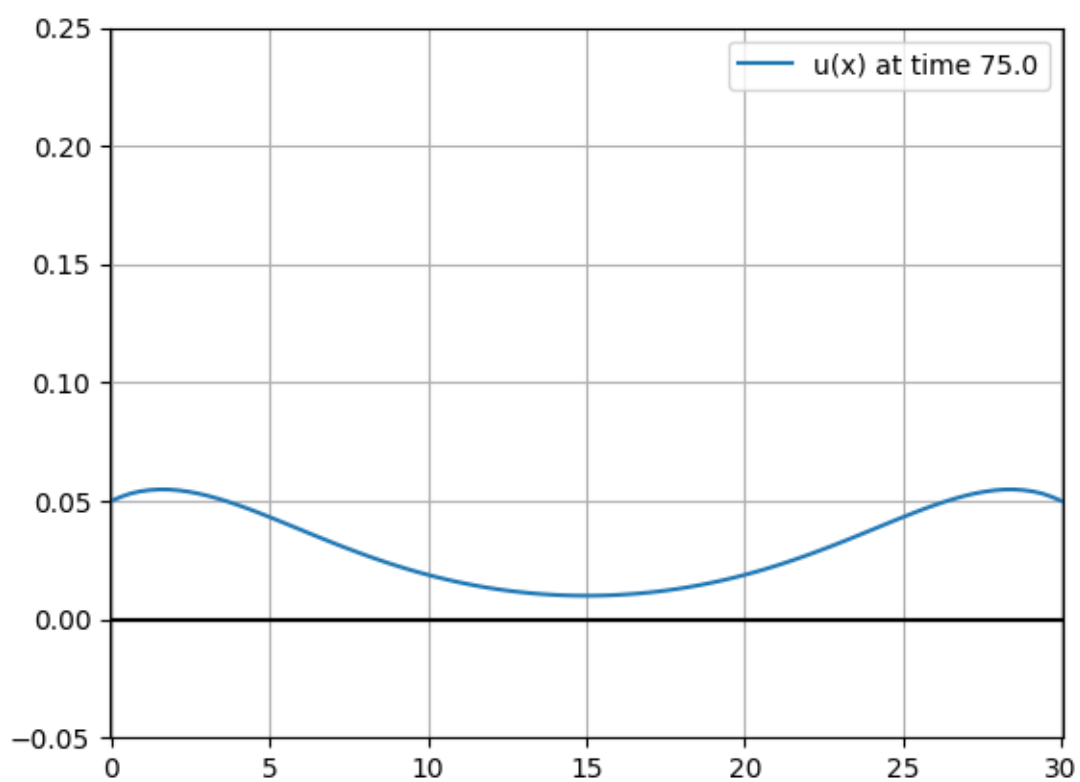
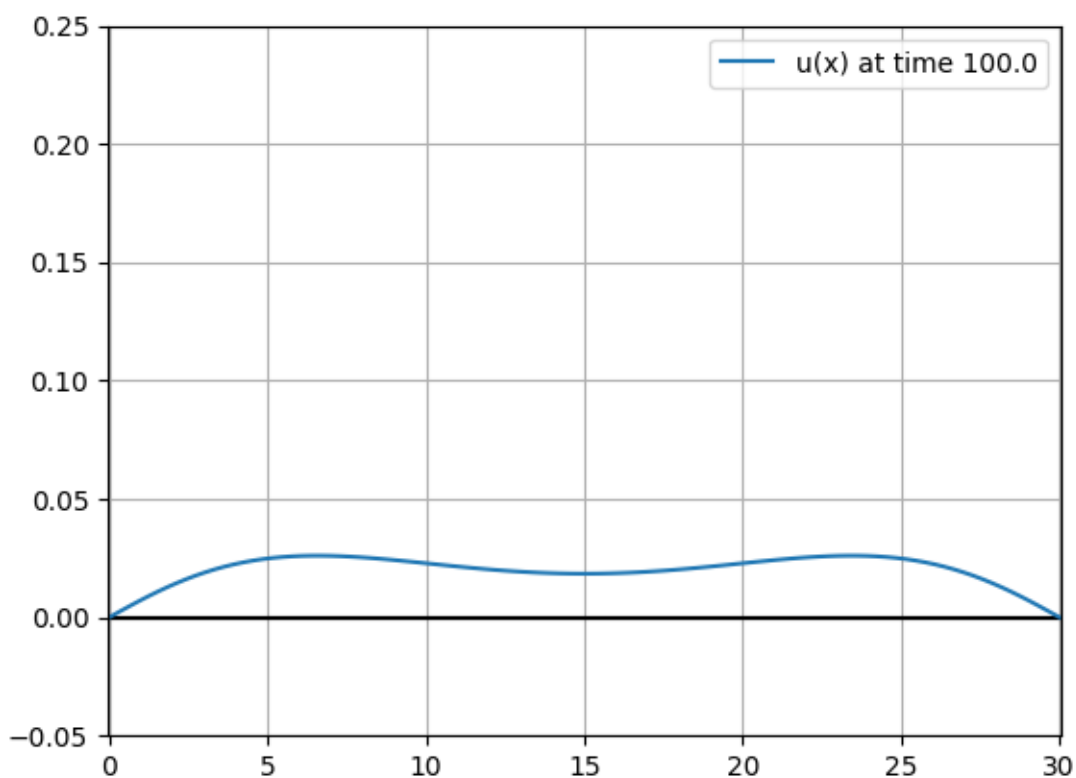


Рисунок 4 – Концентрация вещества в трубке в момент времени  $t = \frac{3T}{4}$ .



*Рисунок 5 – Концентрация вещества в трубке в момент времени  $t = T$ .*

На рисунках 2-5 изображены графики концентрации вещества в трубке в различные моменты времени. Как видно из графиков, концентрация вещества на концах трубки возрастает, а затем убывает, следуя за колебаниями концентрации вещества в растворе. Концентрация вещества в центре с течением времени постепенно растёт, сходясь к некоторому среднему значению с течением времени.

## 5 Экспериментальное исследование качества оценки остатка ряда (Наумов М. Е.)

Проведём экспериментальное исследование качества оценки остатка ряда следующим образом:

1. Будем менять требуемую точность вычислений от  $10^{-1}$  до  $10^{-8}$ .
2. Для каждой требуемой точности рассчитаем  $N_{\text{избыточное}}$  с помощью формулы (51).
3. Рассчитаем  $N_{\text{достаточное}}$ , уменьшая количество слагаемых в сумме, начиная с  $N_{\text{избыточное}}$  до тех пор, пока результат сохраняет требуемую точность.
4. Шаги 1-3 проделаем для моментов времени  $t = 0$ ,  $t = 5$  и  $t = 10000$ . Это необходимо для того, чтобы проследить за изменением  $N_{\text{достаточное}}$  на разных временных промежутках.

*Таблица 1 – Значения  $N_{\text{избыточное}}$  и  $N_{\text{достаточное}}$  при разных  $\epsilon$  и  $t$ .*

$\epsilon$	$t = 0$		$t = 5$		$t = 10000$	
	$N_{\text{избыточное}}$	$N_{\text{достаточное}}$	$N_{\text{избыточное}}$	$N_{\text{достаточное}}$	$N_{\text{избыточное}}$	$N_{\text{достаточное}}$
$10^{-1}$	2	0	2	0	2	0
$10^{-2}$	8	0	8	0	8	2
$10^{-3}$	25	0	25	2	25	3
$10^{-4}$	78	0	78	9	78	5
$10^{-5}$	246	0	246	25	246	8
$10^{-6}$	778	0	778	65	778	15
$10^{-7}$	2462	0	2462	152	2462	25
$10^{-8}$	7784	0	7784	310	7784	38

Как видно из таблицы 1 в моменты времени  $t$  близкие к 0, достаточно небольшого количества слагаемых. Причиной этого является тот факт, что в начальный момент времени концентрация в трубке нулевая, а следовательно достаточно нуля слагаемых для описания системы.

Разница между моментами времени  $t = 5$  и  $t = 10000$  объясняется тем, что в итоговой формуле (32) весь ряд умножается на  $e^{g_{2n+1}t}$ , где  $g_{2n+1}$  - отрицательная величина. Следовательно, в моменты времени  $t \gg 0$  слагаемые убывают с большей скоростью, чем в моменты времени более близкие к моменту  $t = 0$ . Это означает что при больших  $t$  достаточно меньшего количества слагаемых, так как каждый следующий член ряда убывает всё быстрее и быстрее.

## **ЗАКЛЮЧЕНИЕ**

В ходе выполнения данной работы были изучены аналитические методы решения краевой задачи диффузии. Осуществлена математическая постановка краевой задачи для физического процесса, получено аналитическое решение в виде ряда Фурье.

Для анализа полученного решения была разработана программа численного моделирования процесса диффузии вещества в трубке с пористым поглощающим материалом. Программа производит расчёт концентрации вещества в трубке с заданными параметрами и точностью, обеспечивает возможность исследования оценки остатка ряда и контроль погрешности.

Анализ практической пригодности оценки остатка ряда показал большую избыточность вычислений количества суммируемых элементов.

## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Дегтярев А.А. Примеры построения и исследования разностных схем. – Электронное учебное пособие [Электронный ресурс] / А.А Дегтярев, 2011.- 54с.
2. Тихонов А.Н., Самарский А.А. Уравнения математической физики (5-е изд.) [Текст]/Тихонов А.Н., Самарский А.А. М.: Наука, 1977.- 742 с.
3. Numpy and Scipy Documentation [Электронный ресурс]: Официальный сайт документации библиотек Numpy и Scipy. - URL: <https://docs.scipy.org/doc/>
4. Свешников А. Г., Боголюбов А. Н., Кравцов В. В. Лекции по математической физике [Текст] / Свешников А. Г., Боголюбов А. Н., Кравцов В. В. М.: МГУ, 1993. - 352 с.

## ПРИЛОЖЕНИЕ А

### Код программы

```
import numpy as np
import matplotlib.pyplot as plt
import os
import logging
from mpmath import mp
from functools import wraps
import argparse
from sys import exit

mp.dps = 15

T = 100
C = 0.5
D = 0.001
l = 30
alpha = 0.3
a = mp.sqrt(alpha / C)
xi = mp.pi / T

def with_dps(dps):
    """
    Param. decorator.
    When applied to function, changes mp.dps inside it.
    """

    def decorator(f):
        @wraps(f)
        def wrapper(*args, **kwds):
            old_dps = mp.dps
            mp.dps = dps

            res = f(*args, **kwds)
```



```

        mp.dps = old_dps

        return res

    return wrapper

return decorator

def phi(t):
    return 0.1 * mp.sin(xi * t)**2

def u(x, t, num_steps=10):
    return v(x, t, num_steps) + phi(t)

def v(x, t, num_steps=10):
    res = 0

    for k in range(num_steps):
        n = 2 * k + 1 #  $n = 1, 3, 5, \dots$ 
        res += v_n(n, x, t)

    return res

def v_n(n, x, t):
    return C_n(n, t) * X_n(n, x)

phi_0 = phi(0)

@with_dps(20)
def C_n(n, t):
    if n % 2 == 0:
        return 0

```

```

if t == 0:
    return -phi_0 * 4 / (mp.pi * n)

g = -((n * mp.pi * a / l)**2 + D / C)

# put e^(g*t) in integrals, otherwise they will grow
# very (VERY) fast, and result will be Inf
#
# def f1(tau):
# return 0.1 * mp.sin(xi * tau)**2 * mp.exp(g * (t - tau))
#
# def f2(tau):
# return 0.2 * xi * mp.sin(xi * tau) * mp.cos(xi * tau) * \
# mp.exp(g * (t - tau))
#
# I1, err = mp.quad(f1, [0, t], error=True)
# if err > 1e-14:
# print('ERR')
# logging.debug(f'I1 error = {err}')
#
# I2, err = mp.quad(f2, [0, t], error=True)
# if err > 1e-14:
# print('ERR')
# logging.debug(f'I2 error = {err}')
#
res = mp.exp(g * t) * C_n(n, 0) - 4 * (D * I1 / C + I2) / (mp.pi * n)

# analytical solution

I1_a = mp.exp(-g * t) * (
    -g**2 + g**2 * mp.cos(2 * xi * t) - 2 * xi * g * mp.sin(2 * xi * t) +
    4 * xi**2 * mp.exp(g * t) - 4 * xi**2) / (20 * (g**3 + 4 * xi**2 * g))
I2_a = xi * mp.exp(-g * t) * (-g * mp.sin(2 * xi * t) - 2 * xi * mp.cos(
    2 * xi * t) + 2 * xi * mp.exp(g * t)) / (10 * (g**2 + 4 * xi**2))

res_a = mp.exp(g * t) * (C_n(n, 0) - 4 *
    (D * I1_a / C + I2_a)) / (mp.pi * n)

```

```

    return res_a

sin_vec = np.vectorize(mp.sin)

def X_n(n, x):
    return sin_vec(mp.pi * n * x / l)

def steps_by_eps(eps):
    n = (mp.sqrt(l**2 / (20 * mp.pi**3 * a**2 * eps) * (D / C + 1)) - 1) / 2
    return int(mp.ceil(n))

if __name__ == '__main__':
    parser = argparse.ArgumentParser()
    parser.add_argument(
        '-v',
        '--verbose',
        help='increase output verbosity',
        action='store_true')
    parser.add_argument(
        '-e', '--eps', help='precision', type=float, default=1e-3)
    parser.add_argument(
        '-t',
        '--times',
        help='number of full cycles to simulate',
        type=int,
        default=4)
    parser.add_argument(
        '-c',
        '--count',
        help='number of simulation points in one cycle',
        type=int,
        default=20)
    parser.add_argument(

```

```

    '-p', '--plot', help='If set, then plots', action='store_true')
parser.add_argument(
    '-T',
    '--T',
    type=int,
    default=T)
parser.add_argument(
    '-C',
    '--C',
    type=float,
    default=C)
parser.add_argument(
    '-D',
    '--D',
    type=float,
    default=D)
parser.add_argument(
    '-l',
    '--l',
    type=float,
    default=l)
parser.add_argument(
    '-a',
    '--alpha',
    type=float,
    default=alpha)
parser.add_argument(
    '--time_point',
    type=float,
    default=T/2)

args = parser.parse_args()
T = args.T
C = args.C
D = args.D
l = args.l
alpha = args.alpha
time_point = args.time_point

```

```

if args.verbose:
    logging.basicConfig(level=logging.INFO)

eps = args.eps
times = args.times
count = args.count

if eps <= 0 or eps > 1:
    print('Setting precision to default (10^-3).')
    eps = 1e-3

if times <= 0:
    print('Setting number of full cycles to simulate to default (4).')
    times = 4

if count <= 0:
    print(
        'Setting number of simulation points in one cycle to default (20).'
    )
    count = 20

num_steps = steps_by_eps(eps)
print(f'Number of steps: {num_steps}')

x = np.linspace(0, 1, 100)
t = np.linspace(0, times * T, times * count)

n = len(t)

if args.plot:
    if not os.path.exists('plots'):
        os.makedirs('plots')

    for i in range(n):
        logging.info(f'At plotting step {i + 1} out of {n}')

        data = u(x, t[i], num_steps)

```

```

fig = plt.figure()
ax = plt.subplot(111)
plt.xlim([-0.05, 1 + 0.05])
plt.ylim([-0.05, .25])
ax.plot([-0.05, 1 + 0.05], [0, 0], c='black')
ax.plot(x, data, label=f'u(x) at time {round(t[i], 2)}')
ax.legend()
ax.grid()
fig.savefig('plots/plot{:03}.png'.format(i))
plt.close(fig)

solution = u(x, time_point, num_steps)
cache = [v_n(2*n + 1, x, time_point) for n in range(num_steps)]
cache = np.array(cache)
possible_solution = np.sum(cache, axis=0) + phi(time_point)

for n in range(num_steps - 1, -1, -1):
    possible_solution -= cache[n]
    if np.any(np.abs(solution - possible_solution) > eps):
        print(f'Estimated: {num_steps} \t Needed: {n + 1}')
        exit()

if np.any(np.abs(solution - possible_solution) > eps):
    print(f'Estimated: {num_steps} \t Needed: {num_steps}')
else:
    print(f'Estimated: {num_steps} \t Needed: {0}')

```