

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное
учреждение высшего образования
«КАЗАНСКИЙ (ПРИВОЛЖСКИЙ) ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
Институт вычислительной математики и информационных технологий
Кафедра прикладной математики и искусственного интеллекта

Направление: 01.03.04. Прикладная математика

Профиль: Прикладная математика

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

**ПРИБЛИЖЕННОЕ РЕШЕНИЕ НЕЛИНЕЙНОЙ
СТАЦИОНАРНОЙ КРАЕВОЙ ЗАДАЧИ**

Студент гр.09-822

«17» июля 2022г.

 И.Ж. Шабakov


Научный руководитель,
ассистент

«17» июля 2022г.

 Г.О. Трифонова

Заведующий кафедрой,
к.ф.-м.н., доцент

«17» июля 2022г.

 Д.Н. Тумаков

Казань 2022

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Нелинейная краевая задача Дирихле	6
1.1 Теоретическая часть	8
1.1.1 Метод простой итерации для решения уравнения с сильно мо- тонным оператором в Гильбертовом пространстве	8
1.1.2 Аппроксимация МКЭ	8
1.1.3 Итерационный метод для решения аппроксимирующей конечно- мерной нелинейной системы уравнений	11
1.2 Практическая часть	12
1.2.1 Построение нелинейных тестовых задач с известными решениями. .	12
1.2.2 Решение тестовых нелинейных задач средствами созданного ком- плекса программ и методом Ньютона из пакета PDE MatLab.	15
2 Нелинейная задача с точечным источником внутри области	17
2.1 Теоретическая часть	17
2.1.1 Линейная задача с точечным источником. Фундаментальное реше- ние уравнения Лапласа.	17
2.1.2 Метод решения нелинейной задачи с точечным источником с вы- делением особенности в аддитивном виде. Вспомогательная линейная задача, итерационный метод.	18
2.1.3 Формулировка нелинейной краевой задачи со смешанными крае- выми условиями.	20
2.1.4 Формулировка нелинейной краевой задачи со "скважиной" (краевые условия Неймана).	21
2.1.5 Предельный переход в вариационной нелинейной задаче со "сква- жиной" к задаче с точечным источником.	21
2.1.6 Построение решения нелинейных тестовых задач с точечным ис- точником.	22
2.2 Практическая часть	25

2.2.1 Оценки зависимости погрешности от параметров сетки	26
2.2.2 Оценки зависимости погрешности и числа итераций от τ	31
ЗАКЛЮЧЕНИЕ	38
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ	39
ПРИЛОЖЕНИЕ А. Листинг кода(гладкая правая часть)	41
ПРИЛОЖЕНИЕ Б. Листинг кода(точечный источник)	49

ВВЕДЕНИЕ

Работа посвящена изучению нелинейных краевых задач Дирихле в ограниченной области $\Omega \subset R^2$. Большой интерес представляет задача с точечным источником в начале координат. Существование решения данной задачи доказано [1] и представляется в аддитивном виде. Первым слагаемым является решение линейной задачи (принадлежит пространству $W_1^{(1)}$), в которой учтена наша особенность, а именно точечный источник. Второе же слагаемое получается из решения нелинейной задачи (принадлежит пространству $W_2^{(1)}$), поиск которого осуществляется итерационным методом.

Тема работы имеет актуальность и практическую пользу: подобные задачи часто возникают в реальной жизни при рассмотрении задач фильтрации несжимаемой жидкости, которая подчиняется некоторому закону фильтрации. Не мало исследований посвящено данной тематике [1–3].

Целью проекта является построение комплекса программ для решения нелинейных стационарных краевых задач Дирихле и применение численных методов для их решений [4]. В задачи научно-исследовательской работы входит:

- рассмотрение нелинейной краевой задачи Дирихле: формулировка [5] и построение решения тестовых задач,
- изучение метода простой итерации в гильбертовом пространстве [6],
- изучение сущности метода конечных элементов [7–10],
- рассмотрение нелинейной краевой задачи Дирихле с особенностью в правой части (точечный источник): формулировка и построение решения. Представление решения в аддитивном виде,
- проведение серий экспериментов с целью оценки зависимости погрешности решения, полученного с помощью построенного комплекса про-

грамм, от параметров сетки, а так же оценка зависимости погрешности и числа итераций от итерационного параметра.

Работа включает два раздела в основной части, каждый из которых поделен на два пункта: теоретическая и практическая часть.

Первый раздел посвящен нелинейной краевой задаче с гладкой правой частью и итерационному методу решения в гильбертовом пространстве. Во втором же разделе пойдет речь о задаче с особенностью в правой части и подходе, позволяющим нам решать данные задачи.

Программный комплекс реализован в среде MATLAB, предоставляющей множество различных пакетов инструментов, к которым мы прибегнем в ходе выполнения выпускной работы. Стоит выделить пакет среды Matlab Partial Differential Equation(далее PDE) Toolbox, что позволяет решать уравнения в частных производных различных типов [11]. Данному пакету будет посвящен подраздел в основной части проекта.

Работа оформлена в редакторской системе \TeX , которая имеет расширенный функционал для профессиональной верстки сложных документов[12].

1 Нелинейная краевая задача Дирихле

Рассмотрим краевую задачу с условиями Дирихле в ограниченной области $\Omega \subset R^2$, с Липшиц-непрерывной границей Γ :

$$-\operatorname{div} \left(\frac{g(|\nabla w(x)|)}{|\nabla w(x)|} \nabla w(x) \right) = f(x), \quad x \in \Omega, \quad (1)$$

$$w(x) = w_\gamma(x), \quad x \in \Gamma. \quad (2)$$

Здесь, и далее, (\cdot, \cdot) , $|\cdot|$ - скалярное произведение и норма в пространстве R^2 , $f \in L_2(\Omega)$, функция w_γ задана на границе и является следом функции из пространства Соболева $W_2^{(1)}(\Omega)$. Определим вектор-функцию $G : R^2 \rightarrow R^2$ следующим равенством:

$$G(p) = \frac{g(|p|)}{|p|} p, \quad \forall p = (p_1, p_2) \in R^2. \quad (3)$$

Относительно функции $g : R_+^1 \rightarrow R_+^1$ будем предполагать выполненными следующие условия:

- сильной монотонности с константой $m > 0$:

$$g(s) - g(t) \geq m(s - t), \quad \forall s, t \in R_+^1, \quad (4)$$

- Липшиц-непрерывности с константой $M > 0$:

$$|g(s) - g(t)| \leq M|s - t|, \quad \forall s, t \in R_+^1, \quad (5)$$

- равенство

$$g(0) = 0. \quad (6)$$

Получим для (1), (2) вариационную формулировку: для этого умножим обе части уравнения (1) на функцию $\eta \in C_0^\infty(\Omega)$ (здесь $C_0^\infty(\Omega)$ - пространство

функций из $C^\infty(\overline{\Omega})$ равных нулю в окрестности множества Γ) и проинтегрируем по области Ω , после чего воспользуемся формулой Остроградского-Гаусса. В итоге получим:

$$\int_{\Omega} (G(\nabla w(x)), \nabla \eta(x)) dx = \int_{\Omega} f(x)\eta(x)dx, \quad \forall \eta \in C_0^\infty(\Omega). \quad (7)$$

Равенство (7) эквивалентно краевой задаче (1), (2): если $w \in W_2^{(1)}(\Omega)$ удовлетворяет (1) и краевому условию (2), то оно так же удовлетворяет (7) (стоит отметить, что (7) называют интегральным тождеством, соответствующим краевой задаче (1)).

Так как функция g обладает условиями сильной монотонности с константой $m > 0$ и Липшиц-непрерывности с константой $M > 0$, то левая часть уравнения (1) порождает нелинейный, сильно-монотонный, Липшиц-непрерывный оператор в гильбертовом пространстве $W_2^{(1)}$ [1]. Введем норму и скалярное произведение для $H = W_2^{(1)}$:

$$(u, \eta)_H = \int_{\Omega} (\nabla u, \nabla \eta) dx, \quad \|u\|_H^2 = \int_{\Omega} |\nabla u|^2 dx, \quad \forall u, \eta \in H. \quad (8)$$

Имеем нелинейный оператор $A : H \rightarrow H$:

$$(Au, \eta)_H = \int_{\Omega} (G(\nabla w(x)), \nabla \eta) dx.$$

Пусть x, y произвольные элементы из H , тогда оператор A называется сильно-монотонным с постоянной константой $m > 0$, если:

$$(Ax - Ay, x - y)_H \geq m \|x - y\|_H^2, \quad (9)$$

и Липшиц-непрерывным, если существует такая константа M , что:

$$\|Ax - Ay\|_H \leq M \|x - y\|_H, \quad \forall x, y. \quad (10)$$

Получаем операторное уравнение, эквивалентное (1):

$$Au = f.$$

1.1 Теоретическая часть

1.1.1 Метод простой итерации для решения уравнения с сильно монотонным оператором в Гильбертовом пространстве

Покажем, что итерационные методы в гильбертовом пространстве обладают свойством сходимости. Основную роль в доказательстве сходимости играет Лемма 3.1(принцип сжимающих отображений) [6, с. 104]. Теорема 3.4 [6, с. 104] показывает, что операторное уравнение $Au = f$ имеет единственное решение $u \in H$, где H - гильбертово пространство, $A \in (H \rightarrow H)$, $f \in H$. При любом $\tau \in (0, 2m/M^2)$ и начальном приближении u_0 решение представляется в виде последовательности итераций:

$$u^{i+1} = u^i - \tau(Au_i - f), i = 0, 1, 2, \dots, \quad (11)$$

u является сильным пределом этой последовательности.

Имеет место следующая оценка погрешности [6]:

$$\|u_i - u\|_H \leq \frac{k^i \tau}{1 - k} \|Au_0 - f\|_H, \quad (12)$$

где $k(\tau) = \sqrt{1 - 2m\tau + M^2\tau^2} \leq 1$. Для итерационного параметра из интервала $(0, 2m/M^2)$ итерационная последовательность сходится в норме H как геометрическая прогрессия с показателем $k(\tau) = \sqrt{1 - 2m\tau + M^2\tau^2}$. Причем функция k принимает минимальное значение $k_0 = k(\tau_0) = \sqrt{1 - (m/M)^2}$ в точке $\tau = \tau_0 = m/M^2$.

1.1.2 Аппроксимация МКЭ

В основе PDE Toolbox лежит метод конечных элементов для задач на ограниченных областях [9]. В большинстве случаев элементарные функции не позволяют выразить решения даже простых уравнений в частных производных на сложных геометриях. Метод конечных элементов же описывает сложные геометрии как множество подобластей путем генерации сетки на области. Например, мы можем аппроксимировать область Ω с помощью треугольников(т.н. триангуляция области). Треугольники формируют сетку,

а каждая вершина треугольника называется узлом сетки. Следующий шаг - аппроксимация исходной задачи для каждой подобласти более простыми уравнениями.

Рассмотрим эллиптическое уравнение:

$$-\nabla \cdot (c \nabla u) + au = f, x \in \Omega.$$

Предполагается, что Ω имеет кусочно-гладкую границу $\partial\Omega = \partial\Omega_D \cup \partial\Omega_N$ ($\partial\Omega_D \cap \partial\Omega_N = \emptyset$). На границе $\partial\Omega_D$ задано краевое условие Дирихле ($u = u_D, x \in \partial\Omega_D$), а на границе $\partial\Omega_N$ задано краевое условие Неймана ($\nabla \cdot (c \nabla u) \nu + \sigma u = \mu, x \in \partial\Omega_N$, где ν - единичный вектор внешней нормали в точке $x \in \partial\Omega_N$).

Теперь мы должны перейти к интегральному тождеству, действия аналогичны описанным выше в пункте вариационной формулировки задачи с краевыми условиями Дирихле (1), (2): умножаем на тестовую функцию η , значение которой равно нулю в окрестности $\partial\Omega_D$ и интегрируем по области Ω . В результате получаем:

$$\int_{\Omega} (c \nabla u \cdot \nabla \eta + au\eta) d\Omega + \int_{\partial\Omega_N} \sigma u \eta d\partial\Omega_N = \int_{\Omega} f \eta d\Omega + \int_{\partial\Omega_N} \mu \eta d\partial\Omega_N. \quad (13)$$

Далее мы разбиваем (дискретизация) нашу область Ω на лагранжевые конечные элементы Ω^e , где $\Omega = \cup \Omega^e$. Так мы перешли к конечномерному подпространству, где (13) аппроксимируется нашими элементами:

$$\begin{aligned} \int_{\Omega^e} (c \nabla u_h \cdot \nabla \eta_h + au_h \eta_h) d\Omega^e + \int_{\partial\Omega_N^e} \sigma u_h \eta_h d\partial\Omega_N^e = \\ = \int_{\Omega^e} f \eta_h d\Omega^e + \int_{\partial\Omega_N^e} \mu \eta_h d\partial\Omega_N^e. \end{aligned} \quad (14)$$

Далее, пусть ϕ_i , где $i = 1, 2, \dots, N_p$ являются кусочно-полиномиальными базисными функциями для пространства конечных элементов (N_p равно количеству узлов в сетке). Тогда любой конкретный u_h может быть выражен как

линейная комбинация базисных функций:

$$u_h = \sum_1^{N_p} U_i \phi_i,$$

где U_i - это еще не определенные скалярные коэффициенты. Подставляя u_h и $\eta_h = \phi_i$ (в качестве тестовых функций) в (14), а так же выполняя интегрирование по элементу, получим систему N_p уравнений.

Стоит обратить внимание на то, что метод конечных элементов аппроксимирует решение путем минимизации соответствующей функции ошибки[13]. Процесс минимизации автоматически находит линейную комбинацию базисных функций, которая наиболее близка к решению u . Метод конечных элементов дает систему $KU = F$, где матрица K и правая часть F содержат интегралы, содержащие тестовые функции ϕ_i и ϕ_j и коэффициенты c, a, f, σ, μ . Вектор решения U содержит коэффициенты разложения u_h , которые так же являются значениями в узлах сетки. МКЭ позволяет решать нелинейные задачи. В данном случае нелинейность выражается в том, что коэффициенты могут задаваться функциями: получается система $K(U)U = F(U)$.

В наборе инструментов PDE Toolbox используются сетки с треугольными элементами для двумерной геометрии [10]. По умолчанию используется квадратичный геометрический порядок, предоставляется возможность переключиться на линейный порядок. Сетка всегда состоит из элементов одного и того же порядка. Треугольники в двумерных сетках задаются тремя узлами для линейных элементов или шестью узлами для квадратичных элементов. Треугольник, представляющий собой линейный элемент, имеет узлы по углам. Треугольник, представляющий собой квадратичный элемент, имеет узлы в своих углах и центрах ребер. На рисунке 1 показаны треугольные элементы с линейным и квадратичным порядком:

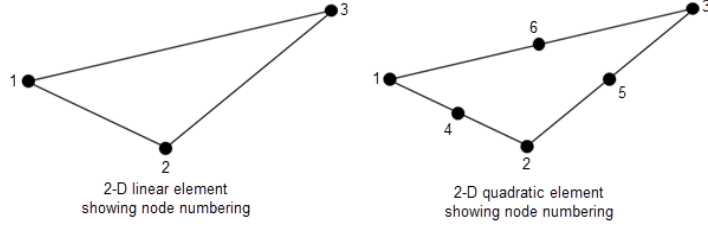


Рисунок 1 — Треугольные элементы линейного и квадратичного порядка

1.1.3 Итерационный метод для решения аппроксимирующей конечно-мерной нелинейной системы уравнений

После обсуждения метода простой итерации в гильбертовом пространстве и сущности метода конечных элементов, перейдем к подходу решения нелинейных краевых задач, который будет использоваться далее. Этот подход основан на совмещении метода конечных элементов с итерационным методом, а именно методом простой итерации.

Наша задача — найти решение $w \in W_2^{(1)}$ для (1). Решение будем искать с помощью итерационного процесса (11), который будет представлен в виде интегрального тождества на каждой итерации. Напомним, что решение интегрального тождества будет эквивалентно решению краевой задачи: если $w \in W_2^{(1)}(\Omega)$ удовлетворяет (1) и краевому условию (2), то оно так же удовлетворяет (7). Запишем вариационную формулировку итерационного процесса (11). Опустим подробности, так как действия аналогичны вариационной записи уравнения (1), (2) Получаем:

$$\int_{\Omega} (w^{k+1}, \eta) dx = \int_{\Omega} (w^k, \eta) dx - \tau \left(\int_{\Omega} (G(\nabla w^k), \nabla \eta) dx - \int_{\Omega} (f, \eta) dx \right),$$

перенесем $\int_{\Omega} (w^k, \eta) dx$ в левую часть и поделим интегральное тождество на $-\tau$:

$$\int_{\Omega} \left(\frac{w^{k+1} - w^k}{-\tau}, \eta \right) dx = \left(\int_{\Omega} (G(\nabla w_k), \nabla \eta) dx - \int_{\Omega} (f, \eta) dx \right).$$

Сделаем замену $\omega = \frac{\omega^{k+1} - \omega^k}{-\tau}$. На каждой итерации итерационного процесса получим:

$$\int_{\Omega} (\omega, \eta) dx = \left(\int_{\Omega} (G(\nabla w^k), \nabla \eta) dx - \int_{\Omega} (f, \eta) dx \right),$$

$$u^{k+1} = u^k - \tau \omega.$$

То есть, решая интегральное тождество на каждом шаге, получаем ω^k , с помощью которого вычисляем значение u^{k+1} для следующей итерации (в качестве u_0 возьмем нулевое приближение). Итерационный процесс продолжается до тех пор, пока не выполнится условие:

$$|u^{k+1} - u^k| < \epsilon.$$

Оценкой погрешностью является запись (12).

1.2 Практическая часть

1.2.1 Построение нелинейных тестовых задач с известными решениями.

Решаем уравнение

$$-\operatorname{div} \left(\frac{g(|\nabla u|)}{|\nabla u|} \nabla u \right) = f, \quad u(x) = u_R, \quad x \in \Gamma \quad (15)$$

с осесимметричной правой частью $f = f(r)$. Для функции $u = u(r)$ имеем

$$\operatorname{div} \left(\frac{g(|\nabla u|)}{|\nabla u|} \nabla u \right) \equiv \frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{g(|\partial u / \partial r|)}{|\partial u / \partial r|} \frac{\partial u}{\partial r} \right).$$

Получаем уравнение

$$-\frac{1}{r} \frac{\partial}{\partial r} \left(r \frac{g(|\partial u / \partial r|)}{|\partial u / \partial r|} \frac{\partial u}{\partial r} \right) = f(r) \quad r > 0,$$

далее имеем

$$\int_0^\rho \frac{\partial}{\partial r} \left(r \frac{g(|\partial u / \partial r|)}{|\partial u / \partial r|} \frac{\partial u}{\partial r} \right) dr = - \int_0^\rho r f(r) dr,$$

$$\rho \frac{g(|\partial u / \partial r|)}{|\partial u / \partial r|} \frac{\partial u(\rho)}{\partial r} = - \int_0^\rho r f(r) dr \quad \rho > 0.$$

Заметим, что

$$\operatorname{sign} \left(\frac{\partial u(\rho)}{\partial r} \right) = \operatorname{sign} \left(- \int_0^\rho r f(r) dr \right) \quad \rho > 0,$$

тогда

$$\begin{aligned} g(|\partial u(\rho)/\partial r|) \operatorname{sign} \left(\frac{\partial u(\rho)}{\partial r} \right) &= -\frac{1}{\rho} \int_0^\rho r f(r) dr \quad \rho > 0, \\ g(|\partial u(\rho)/\partial r|) &= -\frac{1}{\rho} \int_0^\rho r f(r) dr \operatorname{sign} \left(- \int_0^\rho r f(r) dr \right) = \left| - \int_0^\rho r f(r) dr \right|, \end{aligned}$$

$$\begin{aligned} \operatorname{sign} \left(\frac{\partial u(\rho)}{\partial r} \right) \frac{\partial u(\rho)}{\partial r} &= g^{-1} \left(\left| -\frac{1}{\rho} \int_0^\rho r f(r) dr \right| \right) \quad \rho > 0, \\ u_R - u(s) &= \operatorname{sign} \left(\frac{\partial u(\rho)}{\partial r} \right) \int_s^R g^{-1} \left(\left| -\frac{1}{\rho} \int_0^\rho r f(r) dr \right| \right) d\rho, \\ u(s) &= u_R - \operatorname{sign} \left(- \int_0^\rho r f(r) dr \right) \int_s^R g^{-1} \left(\left| -\frac{1}{\rho} \int_0^\rho r f(r) dr \right| \right) d\rho = \\ &= u_R + \operatorname{sign} \left(\int_0^\rho r f(r) dr \right) \int_s^R g^{-1} \left(\frac{1}{\rho} \left| \int_0^\rho r f(r) dr \right| \right) d\rho. \end{aligned}$$

Считаем, что $f(r) \geq 0$, тогда

$$\int_0^\rho r f(r) dr > 0, \quad \frac{\partial u(\rho)}{\partial r} > 0 \quad \text{при} \quad \rho > 0.$$

Получаем

$$u(s) = u_R + \int_s^R g^{-1} \left(\frac{1}{\rho} \int_0^\rho r f(r) dr \right) d\rho.$$

Найдем точное решение для постоянной правой части уравнения.

Положим $f(r) \equiv c_0 > 0$ и пусть

$$g(t) = \begin{cases} kt, & 0 \leq t \leq 1, \\ k_0 t + k - k_0, & 1 < t. \end{cases} \quad (16)$$

$$g^{-1}(t) = \begin{cases} t/k, & 0 \leq t \leq k, \\ t/k_0 + (k_0 - k)/k_0, & k < t. \end{cases}$$

Тогда

$$\frac{1}{\rho} \int_0^\rho r f(r) dr = \frac{1}{\rho} \int_0^\rho r c_0 dr = \frac{c_0 \rho}{2\rho^2} = \frac{c_0}{2}.$$

Пусть $c_0 s^*/2 = k$ или $s^* = 2k/c_0$. При $\rho < s^*$ имеем $c_0 \rho/2 < k$:

$$\begin{aligned} \int_s^R g^{-1} \left(\frac{1}{\rho} \int_0^\rho r f(r) dr \right) d\rho &= \int_s^{s^*} g^{-1} \left(\frac{c_0 \rho}{2} \right) d\rho + \int_{s^*}^R g^{-1} \left(\frac{c_0 \rho}{2} \right) d\rho = \\ &= \int_s^{s^*} \frac{c_0 \rho}{2k} d\rho + \int_{s^*}^R \left(\frac{c_0 \rho}{2k_0} + \frac{k_0 - k}{k_0} \right) d\rho. \end{aligned}$$

Для нашей программы получается надо взять круг радиуса R , на границе взять $u_R = 0$, правую часть $f(r) \equiv c_0 > 0$ и при $k_0 = 1$ точным решением будет:

$$u(s) = \begin{cases} \frac{c_0}{4k}((s^*)^2 - s^2) + \frac{c_0}{4k_0}(R^2 - (s^*)^2) + \frac{k_0 - k}{k_0}(R - s^*), & 0 \leq s \leq s^*, \\ \frac{c_0}{4k_0}(R^2 - s^2) + \frac{k_0 - k}{k_0}(R - s), & s^* < s. \end{cases} \quad (17)$$

Покажем, что функция g является нелинейной. Выведем для задачи $R = 2$, $k = 0.5$, $h = 0.1$ и $c_0 = 1$ пару информативных графиков. На рисунке 2 видим, что при превышении градиентом некоторого порогового значения закон фильтрации меняется, поэтому функция g принимает нелинейный вид, правая часть гладкая.

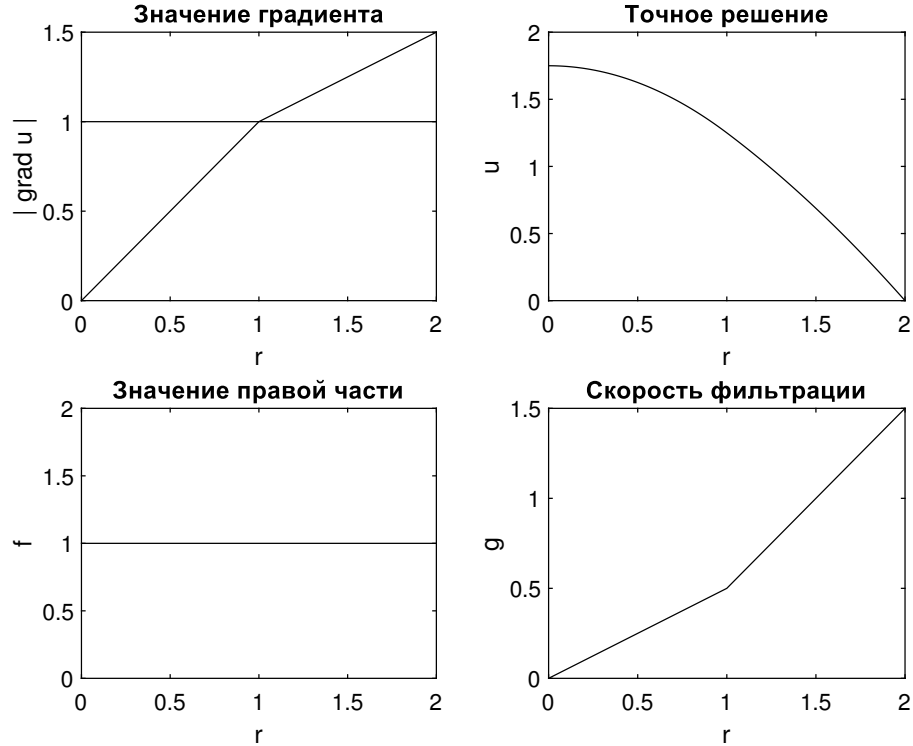


Рисунок 2 — Графики для задачи при $k = 0.5$, $R = 2$ и $c_0 = 1$

1.2.2 Решение тестовых нелинейных задач средствами созданного комплекса программ и методом Ньютона из пакета PDE MatLab.

Перейдем к рассмотрению задачи (15) с постоянной правой частью равной $c_0 = 10$ и радиусом окружности $R = 3$. Точное решение определяется (17), а функция g (16). Интерес представляет изучение зависимости погрешности от размера элементов сетки h . Проведем серию испытаний для $h = 0.1$, $h = 0.05$, $h = 0.03$ и попробуем установить связь. В таблице 1 приведены результаты вычислений. По результатам видим, что при уменьшении значения

Таблица 1 Погрешность вычислений при различных h

h	Погрешность
0.1	0.0080
0.05	0.0027
0.03	0.0010

параметра h , погрешность уменьшается. Созданный комплекс дает отличную точность вычислений, и при уменьшении параметра сетки h точность растет.

Внутри пакета среды Matlab PDE Toolbox имеется функционал, позволяющий решать и нашу задачу. Для решения используется функция `solvepde`, внутри которой заложен метод Ньютона для нелинейных систем уравнений, что указано в документации MATLAB [13]. Сравним результаты вычислений нашего комплекса и пакета PDE Toolbox. В таблице 2 отображено время работы двух программ.

Таблица 2 Время работы комплекса и пакета PDE Matlab

h	Комплекс	Matlab
0.1	0.3653	2.1391
0.05	1.4114	7.5502
0.03	3.0873	16.8810

Таблица показывает значительный прирост времени работы пакета PDE Toolbox по сравнению с нашим комплексом при уменьшении значения параметра h , что говорит о том, что полученный комплекс работает быстрее. Точность же работы у обеих программ на одном уровне, о чем свидетельствует таблица 3.

Таблица 3 Точность(погрешность) решения комплекса и пакета PDE Matlab

h	Комплекс	Matlab
0.1	0.0080	0.0080
0.05	0.0027	0.0027
0.03	0.0010	0.0010

Из вышесказанного можно сделать вывод, что комплекс программ, созданный в ходе выполнения выпускной работы, позволяет решать задачи (15) быстрее стандартных инструментов Matlab без потери точности вычисления. Код программы представлен в приложении А.

2 Нелинейная задача с точечным источником внутри области

Рассмотрим первую краевую задачу для квазилинейного уравнения в ограниченной области $\Omega \subset R^n$, $n \geq 2$, с источником интенсивности q , сосредоточенным в начале координат (ноль - внутренняя точка области):

$$-\operatorname{div} G(\nabla w(x)) = q \delta(x), \quad x \in \Omega, \quad (18)$$

$$w(x) = w_\gamma(x), \quad x \in \Gamma. \quad (19)$$

Здесь Γ - Липшиц-непрерывная граница Ω , функция w_γ - задана на границе и является следом функции из пространства Соболева $W_2^{(1)}(\Omega)$.

Задачу (18), (19) понимаем следующим образом: найти функцию $w \in W_1^{(1)}(\Omega)$, удовлетворяющую условию (19) и вариационному равенству:

$$\int_{\Omega} (G(\nabla w(x)), \nabla \eta(x)) dx = q\eta(0), \quad \forall \eta \in C_0^\infty(\Omega),$$

где $G(p)$ - функция (3). Относительно функции $g : \Omega \times R^n \rightarrow R^n$ выполнены условия (4)–(6).

2.1 Теоретическая часть

2.1.1 Линейная задача с точечным источником. Фундаментальное решение уравнения Лапласа.

В случае $g(s) = c_0 s$, где $c_0 > 0$, $s \in R_+^1$ получим линейную задачу:

$$-c_0 \operatorname{div} (\nabla w(x)) = q \delta(x), \quad x \in \Omega_1,$$

$$w(x) = w_{\gamma_1}(x), \quad x \in \Gamma_1.$$

Фундаментальное решение оператора Лапласа [5, с. 152]:

$$\phi_2(x) = \frac{1}{2\pi} \ln(|x|) \quad , \quad \phi_n(x) = -\frac{1}{(n-2)\sigma_n |x|^{n-2}}, \quad n \geq 3,$$

где $\sigma_n = \operatorname{mes} S$ - площадь единичной сферы $S = \{x \in R^n : |x| = 1\}$.

Функция

$$\Phi(x) = q\phi_n/c_0 \quad (20)$$

будет решением задачи с краевым условием $w_{\gamma_1} = \Phi$.

Чтобы решить задачу с краевыми условиями нужно представить $w = \Phi + u_0$, где u_0 решение следующей краевой задачи:

$$-\operatorname{div}(c_0 \nabla u_0(x)) = 0, \quad x \in \Omega, \quad (21)$$

$$u_0(x) = w_{\gamma_1}(x) - \Phi(x), \quad x \in \Gamma. \quad (22)$$

2.1.2 Метод решения нелинейной задачи с точечным источником с выделением особенности в аддитивном виде. Вспомогательная линейная задача, итерационный метод.

Решение (18) представляем [2] в виде $w = \Phi + u_0 + u$, где функция u принадлежит пространству $\overset{\circ}{W}_2^{(1)}(\Omega)$, а функция $w_0 = \Phi + u_0$ из пространства $W_1^{(1)}(\Omega)$ удовлетворяет граничному условию (19) и вариационному равенству:

$$\int_{\Omega} (G \nabla w_0(x), \nabla \eta(x)) \, dx = q\eta(0), \forall \eta \in C_0^\infty(\Omega).$$

Для поиска $w_0 = \Phi + u_0$ приходим к линейной задаче с точечным источником и формула для Φ это (20) для u_0 это решение линейной задачи (21), (22). В случае решения задачи в пространстве R^2 на круге радиуса R , для функции w_0 решение будет равно функции ϕ по следующим формулам:

$$x = (x_1; x_2) \in R^2, \quad \phi(x) = -\frac{q}{2\pi}(\ln|x| - \ln R),$$

$$\nabla \phi(x) = \frac{-q}{2\pi|x|} \nabla(|x|) = \frac{-q}{2\pi|x|} \frac{x}{|x|} = \frac{-qx}{2\pi|x|^2}.$$

Для поиска решения $u \in W_2^{(1)}$ рассмотрим итерационный процесс, который представляет из себя метод простой итерации в гильбертовом пространстве. Зададим нулевое приближение u^0 и получаем искомый процесс,

на каждом шаге которого формируется вариационная задача:

$$\begin{aligned} \int_{\Omega} \left(\nabla u^{(k+1)}, \nabla \eta \right) dx = \int_{\Omega} \left(\nabla u^{(k)}, \nabla \eta \right) dx - \\ - \tau \int_{\Omega} \left(g(\nabla u^{(k)} + \nabla w_0) - \nabla w_0, \nabla \eta \right) dx, \end{aligned}$$

где η - произвольная функция из $W_2^{(1)}$.

Покажем, что интеграл $\int_{\Omega} \left(g(\nabla u^{(k)} + \nabla w_0) - \nabla w_0, \nabla \eta \right) dx$ имеет смысл. Пусть относительно функции $g(\lambda) : \Omega \times R^n \rightarrow R^n$ выполнены условия (4)–(6), функция v - произвольная из пространства $W_2^{(1)}(\Omega)$, а $w_0 = \Phi + u_0$. Тогда вектор функция $P(x) \equiv g(\nabla v(x) + \nabla w_0(x)) - \nabla w_0(x)$ принадлежит пространству $[L_2(\Omega)]^n$ [1]. Это означает, что интегралы, входящие в определение итерационного метода для поиска функции u , ограничены. Сформулируем наш итерационный процесс в виде операторного уравнения. Норма и скалярное произведение для $H = W_2^{(1)}(\Omega)$ определены в (8).

Из утверждения выше следует, что для функции $u \in H$ и произвольной функции $\eta \in H$ определена линейная и непрерывная по η форма (функционал $l_u : H \rightarrow R^1$):

$$l_u(\eta) = \int_{\Omega} (g(\nabla u + \nabla w_0) - \nabla w_0, \nabla \eta) dx.$$

По теореме Рисса-Фишера существует элемент из пространства H (обозначим его $A(u)$), такой что $(A(u), \eta)_H = l_u(\eta)$ для произвольной функции η из пространства H [1]. Введем оператор $A : H \rightarrow H$ следующим образом:

$$(Au, \eta)_H = \int_{\Omega} (g(\nabla u + \nabla w_0) - \nabla w_0, \nabla \eta) dx.$$

Тогда итерационный процесс для нахождения функции u запишем в виде:

$$(u^{k+1}, \eta)_H = (u^k, \eta)_H - \tau(Au^k, \eta)_H, \quad \forall \eta \in H,$$

или более коротко, в операторном виде:

$$u^{k+1} = u^k - \tau Au^k.$$

Таким образом, рассматриваемый итерационный метод является методом простой итерации для решения уравнения $Au = 0$. Для оператора A выполнены условия (9) и (10) [1], а так же справедлива оценка в норме H (12) [3].

2.1.3 Формулировка нелинейной краевой задачи со смешанными краевыми условиями.

Рассмотрим краевую задачу со смешанными краевыми условиями в ограниченной области $\Omega \subset R^2$, с Липшиц-непрерывной границей $\Gamma = \Gamma_1 \cup \Gamma_2$, $\Gamma_1 \cap \Gamma_2 = \emptyset$, $\text{mes } \Gamma_1 \neq 0$:

$$-\text{div} \left(\frac{g(|\nabla w(x)|)}{|\nabla w(x)|} \nabla w(x) \right) = f(x), \quad x \in \Omega, \quad (23)$$

$$w(x) = w_{\gamma_1}(x), \quad x \in \Gamma_1, \quad (24)$$

$$(g(\nabla w(x)), \nu(x)) + \sigma(x)w(x) = w_{\gamma_2}(x), \quad x \in \Gamma_2. \quad (25)$$

$f \in L_2(\Omega)$, функция w_{γ_1} задана на границе и является следом функции из пространства Соболева $W_2^{(1)}(\Omega)$, $w_{\gamma_2} \in L_2(\Gamma_2)$, функция $\sigma : \Gamma_2 \rightarrow R^1$ почти всюду неотрицательна и ограничена:

$$0 \leq \sigma(x) \leq \bar{\sigma}.$$

Вектор-функцию $G : R^2 \rightarrow R^2$ определена как (3). Относительно функции $g : R_+^1 \rightarrow R_+^1$ выполняются условия (4)–(6).

Вариационная задача для (23)–(25): найти функцию $w \in W_2^{(1)}(\Omega)$, удовлетворяющую краевому условию (24) и вариационному равенству (здесь $C_0^\infty(\Omega \cup \Gamma_2)$ - пространство функций из $C^\infty(\bar{\Omega})$ равных нулю в окрестности

множества Γ_1):

$$\begin{aligned} \int_{\Omega} (G(\nabla w(x)), \nabla \eta(x)) dx + \int_{\Gamma_2} \sigma(x) w(x) \eta(x) dx = \\ = \int_{\Gamma_2} w_{\gamma_2}(x) \eta(x) dx + \int_{\Omega} f(x) \eta(x) dx, \quad \forall \eta \in C_0^\infty(\Omega \cup \Gamma_2). \end{aligned} \quad (26)$$

2.1.4 Формулировка нелинейной краевой задачи со "скважиной" (краевые условия Неймана).

Пусть $\Omega_1 \subset R^2$ ограниченная область (начало координат - внутренняя точка) с Липшиц-непрерывной границей Γ_1 . "Скважина" в начале координат моделируется кругом $B_\varepsilon(0)$ ($B_r(y) = \{x \in R^n : |x - y| < r\}$) малого радиуса ε , на границе которого S_ε задана известная плотность потока $w_{\gamma_2}(x) \equiv q_\varepsilon$, где $q_\varepsilon = q/(2\pi\varepsilon)$ и функция $\sigma(x) \equiv 0$.

Таким образом $\Omega = \Omega_1 \setminus B_\varepsilon(0)$ с границей $\Gamma = \Gamma_1 \cup S_\varepsilon$. Считаем, что кроме скважины, других источников нет ($f(x) \equiv 0$) и на внешней границе $w \equiv 0$ ($w_{\gamma_1}(x) \equiv 0$).

Вариационная задача имеет вид:

$$\int_{\Omega} (G(\nabla w(x)), \nabla \eta(x)) dx = \frac{q}{2\pi\varepsilon} \int_{S_\varepsilon} \eta(x) dx, \quad \forall \eta \in C_0^\infty(\Omega \cup S_\varepsilon).$$

2.1.5 Предельный переход в вариационной нелинейной задаче со "скважиной" к задаче с точечным источником.

В пределе, когда $\varepsilon \rightarrow 0$ (что соответствует малым размерам скважины по отношению к размерам области решения) получаем:

$$\int_{S_\varepsilon} \eta(x) dx = 2\pi\varepsilon\eta(x_\varepsilon) \rightarrow 2\pi\varepsilon\eta(0),$$

здесь x_ε некоторая точка окружности S_ε (существует по теореме о среднем для интеграла от непрерывной функции). В пределе получаем следующую вариационную задачу (задача с точечным источником): найти функцию

$w \in \overset{\circ}{W}_2^{(1)}(\Omega_1)$ (выполнено $w \equiv 0$ на Γ_1)

$$\int_{\Omega} (G(\nabla w(x)), \nabla \eta(x)) dx = q\eta(0), \quad \forall \eta \in C_0^\infty(\Omega_1).$$

2.1.6 Построение решения нелинейных тестовых задач с точечным источником.

Задача с точечным источником в пространстве R^2

$$-\operatorname{div} \left(\frac{g(|\nabla w(x)|)}{|\nabla w(x)|} \nabla w(x) \right) = q\delta(x), \quad x \in \Omega = B_R,$$

$$w(x) = w_R, \quad x \in S_R.$$

Предельный случай следующей краевой задачи:

$$-\operatorname{div} \left(\frac{g(|\nabla w(x)|)}{|\nabla w(x)|} \nabla w(x) \right) = 0, \quad x \in \Omega = B_R \setminus B_\varepsilon,$$

$$w(x) = w_R, \quad x \in S_R.$$

$$\left(\frac{g(|\nabla w(x)|)}{|\nabla w(x)|} \nabla w(x), \frac{-x}{|x|} \right) = q_\varepsilon, \quad x \in S_\varepsilon,$$

$$q_\varepsilon = \frac{q}{|S_\varepsilon|} = \frac{q}{2\pi\varepsilon} = \frac{q^*}{\varepsilon}, \quad q^* = \frac{q}{2\pi}.$$

Задача осесимметричная и решение ищем осесимметричное $w(x_1, x_2) = w(r, \varphi) \equiv w(r)$. По определению производной по направлению имеем:

$$\left(\nabla w(x), \frac{x}{|x|} \right) = \frac{\partial w(x_1, x_2)}{\partial e_r} = \frac{\partial w(r, \varphi)}{\partial r} = \frac{dw(r)}{dr}, \quad e_r = \frac{x}{|x|},$$

$$\left(\frac{g(|\nabla w(x)|)}{|\nabla w(x)|} \nabla w(x), \frac{-x}{|x|} \right) = -g(|dw(r)/dr|) \frac{dw(r)/dr}{|dw(r)/dr|} =$$

$$= -g \left(\left| \frac{dw}{dr} \right| \right) \operatorname{sign} \left(\frac{dw}{dr} \right).$$

Следовательно

$$\operatorname{sign} \left(\frac{g(|\nabla w(x)|)}{|\nabla w(x)|} \nabla w(x), \frac{-x}{|x|} \right) = -\operatorname{sign} \left(\frac{g(|\nabla w(x)|)}{|\nabla w(x)|} \frac{\partial w(r, \varphi)}{\partial r} \right) =$$

$$= -\operatorname{sign} \left(\frac{\partial w(r, \varphi)}{\partial r} \right).$$

Получаем, что

$$-\text{sign} \left(\frac{dw(\varepsilon)}{dr} \right) = \text{sign} \left(\frac{q}{2\pi\varepsilon} \right) = \text{sign}(q^*) \quad 0 < \varepsilon < R.$$

Задача с точечным источником в пространстве R^2 для осесимметричного случая сводится к следующей краевой задаче

$$\frac{1}{r} \frac{d}{dr} \left(r g \left(\left| \frac{dw(r)}{dr} \right| \right) \text{sign} \left(\frac{dw(r)}{dr} \right) \right) = 0 \quad 0 < r < R,$$

$$w(R) = u_R$$

$$\lim_{\varepsilon \rightarrow +0} -\varepsilon \text{sign} \left(\frac{dw(\varepsilon)}{dr} \right) g \left(\left| \frac{dw(\varepsilon)}{dr} \right| \right) = q^*.$$

Интегрируем уравнение

$$\int_0^\rho \frac{d}{dr} \left(r g \left(\left| \frac{dw(r)}{dr} \right| \right) \text{sign} \left(\frac{dw(r)}{dr} \right) \right) dr = 0 \quad 0 < \rho < R,$$

$$\rho g \left(\left| \frac{dw(\rho)}{dr} \right| \right) \text{sign} \left(\frac{dw(\rho)}{dr} \right) - \lim_{\varepsilon \rightarrow +0} \varepsilon \text{sign} \left(\frac{dw(\varepsilon)}{dr} \right) g \left(\left| \frac{dw(\varepsilon)}{dr} \right| \right) = 0,$$

$$\rho g \left(\left| \frac{dw(\rho)}{dr} \right| \right) (-\text{sign}(q^*)) + q^* = 0 \quad 0 < \rho < R,$$

$$\rho g \left(\left| \frac{dw(\rho)}{dr} \right| \right) - |q^*| = 0 \quad 0 < \rho < R,$$

$$\frac{dw(\rho)}{dr} \text{sign} \left(\frac{dw(\rho)}{dr} \right) = g^{-1} \left(\frac{|q^*|}{\rho} \right) \quad 0 < \rho < R,$$

$$\frac{dw(\rho)}{dr} = \text{sign} \left(\frac{dw(\rho)}{dr} \right) g^{-1} \left(\frac{|q^*|}{\rho} \right) = (-\text{sign}(q)) g^{-1} \left(\frac{|q^*|}{\rho} \right) \quad 0 < \rho < R.$$

Проинтегрируем от r до R

$$w(R) - w(r) = \int_r^R \frac{dw(\rho)}{dr} d\rho = (-\text{sign}(q^*)) \int_r^R g^{-1} \left(\frac{|q^*|}{\rho} \right) d\rho \quad 0 < r < R,$$

$$w(r) = w_R + \text{sign}(q^*) \int_r^R g^{-1} \left(\frac{|q^*|}{\rho} \right) d\rho \quad 0 < r < R.$$

Решение задачи будем искать для следующей функции

$$g(s) = \begin{cases} ks, & 0 \leq s \leq 1, \\ s - 1 + k, & 1 < s. \end{cases}, \quad g^{-1}(s) = \begin{cases} k^{-1}s, & 0 \leq s \leq k, \\ s + 1 - k, & k < s. \end{cases}$$

Пусть

$$\frac{|q^*|}{\rho} = s \leq k = \frac{|q^*|}{r^*},$$

тогда

$$\rho \geq r^* = \frac{|q^*|}{k} = k^{-1}|q^*| = (2\pi k)^{-1}|q|.$$

Вернемся к интегралу

$$\begin{aligned} \int_r^R g^{-1}\left(\frac{|q^*|}{\rho}\right) d\rho &= \int_r^{r^*} g^{-1}\left(\frac{|q^*|}{\rho}\right) d\rho + \int_{r^*}^R g^{-1}\left(\frac{|q^*|}{\rho}\right) d\rho = \\ &= \int_r^{r^*} \left(\frac{|q^*|}{\rho} + 1 - k\right) d\rho + \int_{r^*}^R k^{-1} \frac{|q^*|}{\rho} d\rho = \\ &= (|q^*| \ln \rho + (1 - k)\rho) \Big|_r^{r^*} + k^{-1}|q^*| \ln \rho \Big|_{r^*}^R = \\ &= \begin{cases} k^{-1}|q^*|(\ln R - \ln r), & r^* \leq r \leq R, \\ |q^*|(\ln r^* - \ln r) + (1 - k)(r^* - r) + \\ k^{-1}|q^*|(\ln R - \ln r^*), & 0 < r \leq r^*, \end{cases} = \\ &= \begin{cases} k^{-1}|q^*|(\ln R - \ln r), & r^* \leq r \leq R, \\ |q^*|(\ln R - \ln r) + (1 - k)(r^* - r) + \\ (k^{-1} - 1)|q^*|(\ln R - \ln r^*), & 0 < r \leq r^*, \end{cases} = \\ &= \begin{cases} k^{-1} \operatorname{sign}(q) \phi(r), & r^* \leq r \leq R, \\ \operatorname{sign}(q) \phi(r) + (1 - k)(r^* - r) + (k^{-1} - 1) \operatorname{sign}(q) \phi(r^*), & 0 < r \leq r^*, \end{cases} = \\ &= \operatorname{sign}(q) \begin{cases} k^{-1} \phi(r), & r^* \leq r \leq R, \\ \phi(r) + \operatorname{sign}(q)(1 - k)(r^* - r) + (k^{-1} - 1) \phi(r^*), & 0 < r \leq r^*, \end{cases} \end{aligned}$$

где

$$\begin{aligned}\phi(r) &= \text{sign}(q^*)|q^*|(\ln R - \ln r) = \text{sign}(q)(2\pi)^{-1}|q|(\ln R - \ln r) = \\ &= (2\pi)^{-1}q(\ln R - \ln r) = (2\pi)^{-1}q(\ln(R/r))\end{aligned}$$

– решение уравнения Лапласа на круге B_R с точечным источником интенсивности q в центре и однородными краевыми условиями Дирихле. Тогда для вычисления $w(r)$ справедливо

$$w(r) = w_R + \begin{cases} k^{-1}\phi(r), & r^* \leq r \leq R, \\ \phi(r) + \text{sign}(q)(1-k)(r^* - r) + (k^{-1} - 1)\phi(r^*), & 0 < r \leq r^*. \end{cases}$$

Решение представим в аддитивном виде (выделим особенность, связанную с точечным источником) $w = u + \phi$

$$u(r) = w(r) - \phi(r) = \begin{cases} (k^{-1} - 1)\phi(r), & r^* \leq r \leq R, \\ \text{sign}(q)(1-k)(r^* - r) + (k^{-1} - 1)\phi(r^*), & 0 < r \leq r^*. \end{cases}$$

Учитывая $\phi(r)$

$$u(r) = \begin{cases} q(1-k)(2\pi k)^{-1} \ln(R/r), & r^* \leq r \leq R, \\ -\text{sign}(q)(1-k)r + (1-k)q(2\pi k)^{-1}(1 + \ln(R/r^*)), & 0 < r \leq r^*. \end{cases} \quad (27)$$

Для производной справедливо

$$\frac{du(r)}{dr} = \begin{cases} q(k-1)/(2\pi kr), & r^* \leq r \leq R, \\ \text{sign}(q)(k-1), & 0 < r \leq r^*. \end{cases}$$

2.2 Практическая часть

Прежде чем перейти к задаче (18), стоит сделать замечание: так как в $w(x)$ имеется особенность, целесообразно в качестве исследования использовать функцию $u(r)$, точное решение которой определено (27), что и будет показано в следующих подразделах.

2.2.1 Оценки зависимости погрешности от параметров сетки

Посмотрим, как влияет на точность решения размер элементов h в сетке. Так же установим влияние значения q на погрешность, где q - интенсивность источника. Размер элементов в сетке принимает значения от $h_0 = 0.1$ до $h_k = \frac{h_0}{16}$ с шагом $h_{step} = \frac{h_0}{2}$. Вычисления производились при $\tau = 1$ и $k = 0.5$. Результат продемонстрирован на рисунке 3.

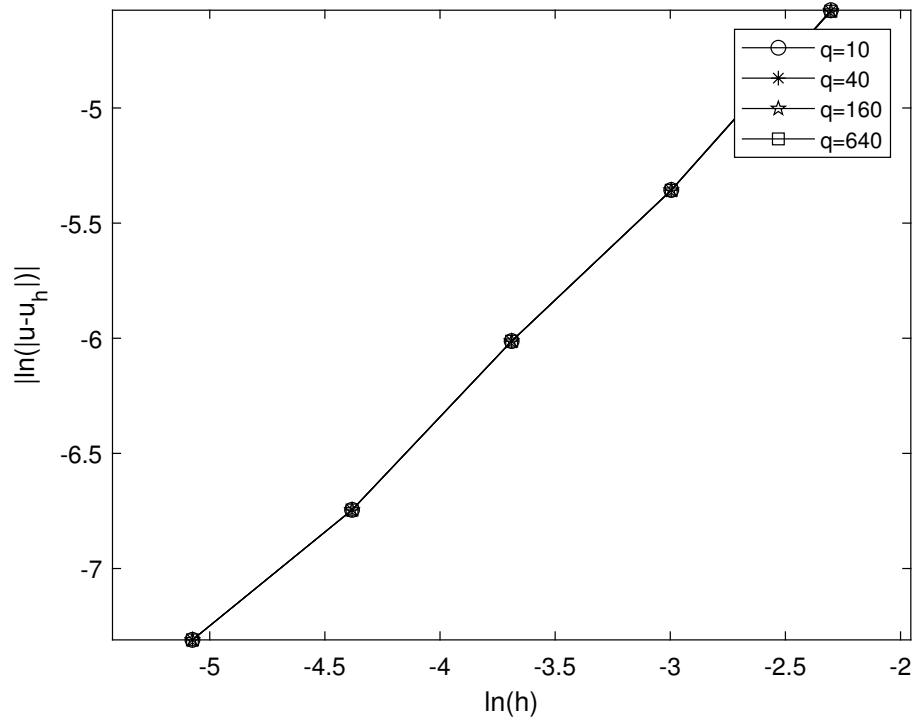


Рисунок 3 — Погрешность решения при различных значениях h и q

Погрешность и значение h представлены в логарифмическом масштабе. Так же отразим результат в виде таблиц 4-7,

Таблица 4 Погрешность при $q = 10$

q= 10.0000			
Итерация	h	Погрешность	Оценка погрешности
14	0.1000	1.03021e-02	1.987
15	0.0500	4.71969e-03	1.788
16	0.0250	2.44895e-03	1.630
16	0.0125	1.17668e-03	1.539
17	0.0063	4.05326e-04	1.539

Таблица 5 Погрешность при $q = 40$

q= 40.0000			
Итерация	h	Погрешность	Оценка погрешности
14	0.1000	1.02704e-02	1.988
15	0.0500	4.71456e-03	1.788
16	0.0250	2.44759e-03	1.630
16	0.0125	1.17604e-03	1.539
17	0.0063	4.05278e-04	1.539

Таблица 6 Погрешность при $q = 160$

q= 160.0000			
Итерация	h	Погрешность	Оценка погрешности
14	0.1000	1.02622e-02	1.989
15	0.0500	4.71322e-03	1.788
16	0.0250	2.44723e-03	1.630
16	0.0125	1.17588e-03	1.539
17	0.0063	4.05265e-04	1.539

Таблица 7 Погрешность при $q = 640$

q= 640.0000			
Итерация	h	Погрешность	Оценка погрешности
14	0.1000	1.02601e-02	1.989
15	0.0500	4.71288e-03	1.788
16	0.0250	2.44715e-03	1.630
16	0.0125	1.17584e-03	1.539
17	0.0063	4.05261e-04	1.539

где Оценка погрешности = $\frac{\text{Погрешность}}{h}$.

Теперь посмотрим как на погрешность повлияет параметр k . Отразим результат на рисунке 4 и таблицах 8-10 при $\tau = 1, q = 100$.

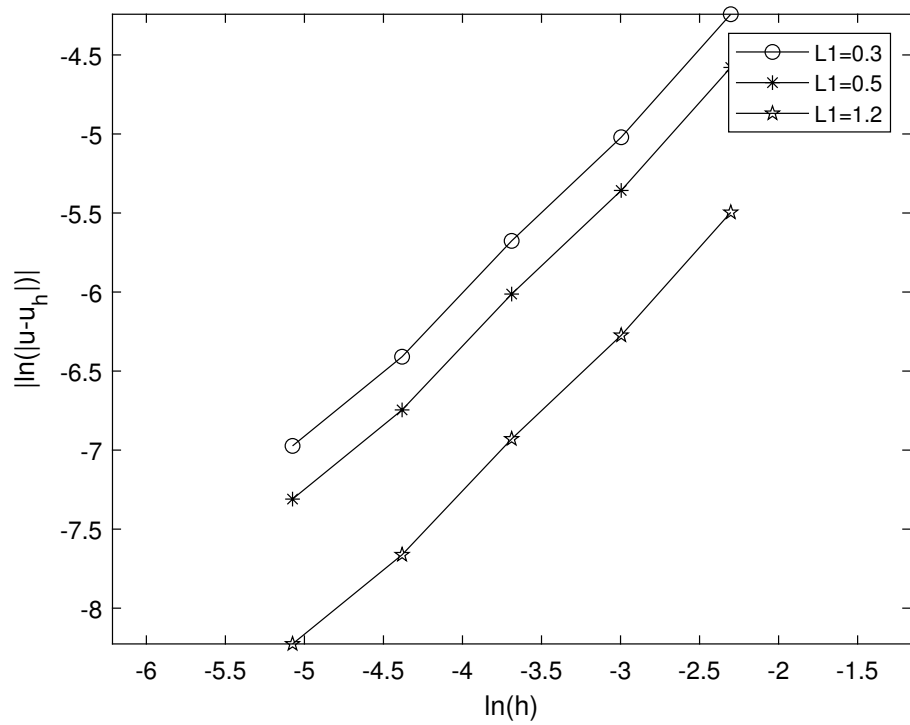


Рисунок 4 — Погрешность решения при различных значениях h и k

Таблица 8 Погрешность при $k = 0.3$

$k = 0.3$			
Итерация	Сетка	Погрешность	Оценка погрешности
24	0.1000	1.43718e-02	1.842
26	0.0500	6.59956e-03	1.676
27	0.0250	3.42621e-03	1.539
28	0.0125	1.64632e-03	1.463
29	0.0063	9.36438e-04	1.374

Таблица 9 Погрешность при $k = 0.5$

$k = 0.5$			
Итерация	Сетка	Погрешность	Оценка погрешности
14	0.1000	1.02638e-02	1.989
15	0.0500	4.71349e-03	1.788
16	0.0250	2.44730e-03	1.630
16	0.0125	1.17591e-03	1.539
17	0.0063	6.68882e-04	1.440

Таблица 10 Погрешность при $k = 1.2$

$k = 1.2$			
Итерация	Сетка	Погрешность	Оценка погрешности
7	0.1000	4.10282e-03	2.387
8	0.0500	1.88502e-03	2.094
8	0.0250	9.78792e-04	1.878
8	0.0125	4.70292e-04	1.749
9	0.0063	2.67548e-04	1.621

Видим, что:

- уменьшение значения параметра h в сетке увеличивает точность решения,
- решение сходится при любых значениях q , и значение q не оказывает большого влияния на точность решения,
- комплекс программ дает отличную погрешность при различных значениях k .

2.2.2 Оценки зависимости погрешности и числа итераций от τ

Так как в качестве решения задачи используется метод простой итерации, мы можем оценить оптимальное значение итерационного параметра, при котором количество итераций было бы минимально. На рисунке 5 продемонстрирован график с оптимальными значениями τ при различных значениях интенсивности источника q и $k = 0.5$:

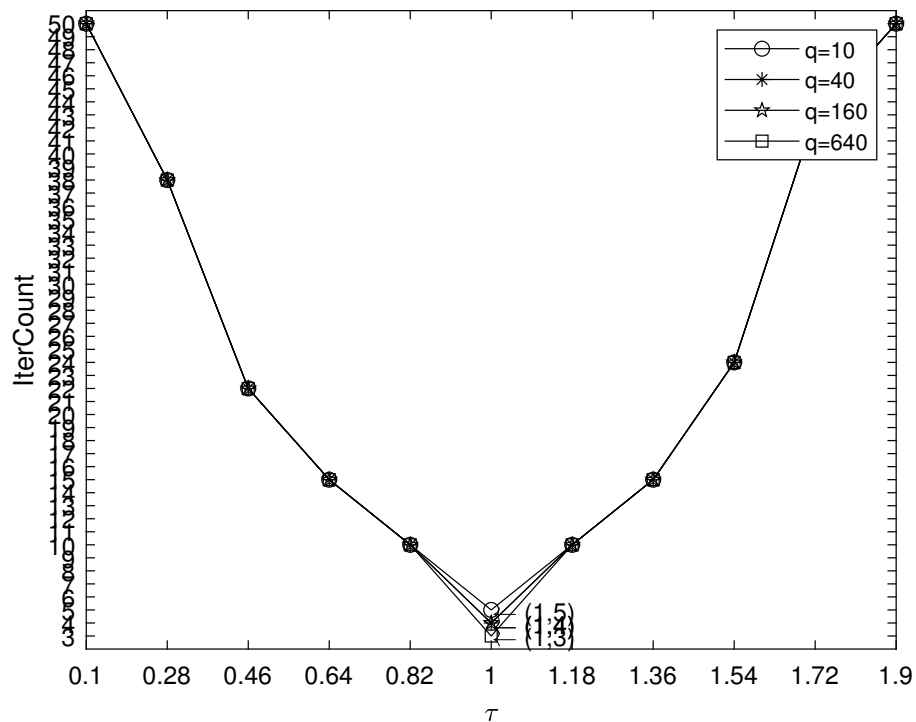


Рисунок 5 — Количество итераций при различных значениях q и τ

На рисунке 6 предоставлена погрешность при различных τ и $k = 0.5$.

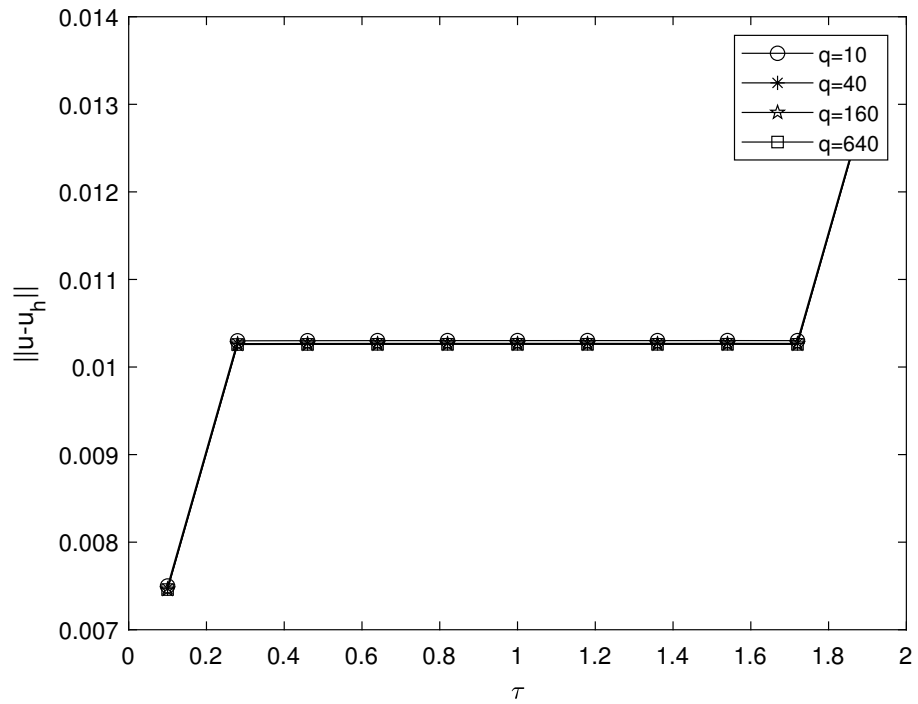


Рисунок 6 — Погрешность при различных значениях q и τ

Для большей наглядности оформим данные в виде таблиц 11-14:

Таблица 11 Погрешность и количество итераций при $q = 10$

τ	Количество итераций	Ошибка
0.100	50	7.49939e-03
0.280	38	1.02992e-02
0.460	22	1.03006e-02
0.640	15	1.03015e-02
0.820	10	1.03017e-02
1.000	5	1.03018e-02
1.180	10	1.03019e-02
1.360	15	1.03015e-02
1.540	24	1.03022e-02
1.720	43	1.03013e-02
1.900	50	1.30951e-02
Лучший $\tau = 1$		$q = 10.0000$

Таблица 12 Погрешность и количество итераций при $q = 40$

τ	Количество итераций	Ошибка
0.100	50	7.46887e-03
0.280	38	1.02675e-02
0.460	22	1.02689e-02
0.640	15	1.02698e-02
0.820	10	1.02700e-02
1.000	4	1.02701e-02
1.180	10	1.02702e-02
1.360	15	1.02698e-02
1.540	24	1.02704e-02
1.720	43	1.02696e-02
1.900	50	1.30665e-02
Лучший $\tau = 1$		q= 40.0000

Таблица 13 Погрешность и количество итераций при $q = 160$

τ	Количество итераций	Ошибка
0.100	50	7.46096e-03
0.280	38	1.02593e-02
0.460	22	1.02607e-02
0.640	15	1.02616e-02
0.820	10	1.02618e-02
1.000	4	1.02619e-02
1.180	10	1.02620e-02
1.360	15	1.02616e-02
1.540	24	1.02622e-02
1.720	43	1.02614e-02
1.900	50	1.30612e-02
Лучший $\tau = 1$		q= 160.0000

Таблица 14 Погрешность и количество итераций при $q = 640$

τ	Количество итераций	Ошибка
0.100	50	7.45896e-03
0.280	38	1.02572e-02
0.460	22	1.02586e-02
0.640	15	1.02595e-02
0.820	10	1.02597e-02
1.000	3	1.02598e-02
1.180	10	1.02599e-02
1.360	15	1.02595e-02
1.540	24	1.02601e-02
1.720	43	1.02593e-02
1.900	50	1.30602e-02
Лучший $\tau = 1$		q= 640.0000

Для нашей задачи мы получили оптимальные значения τ , позволяющие достичь нужной точности за наименьшее число итераций. Погрешность по мере увеличения значения q остается практически неизменной, что видно на рисунке 6.

Теперь взглянем на то, как изменяется интервал сходимости и вследствие этого оптимальное значение итерационного параметра при различных значениях параметра k . Результат отображен на рисунке 7 при $q = 100$. Пока-

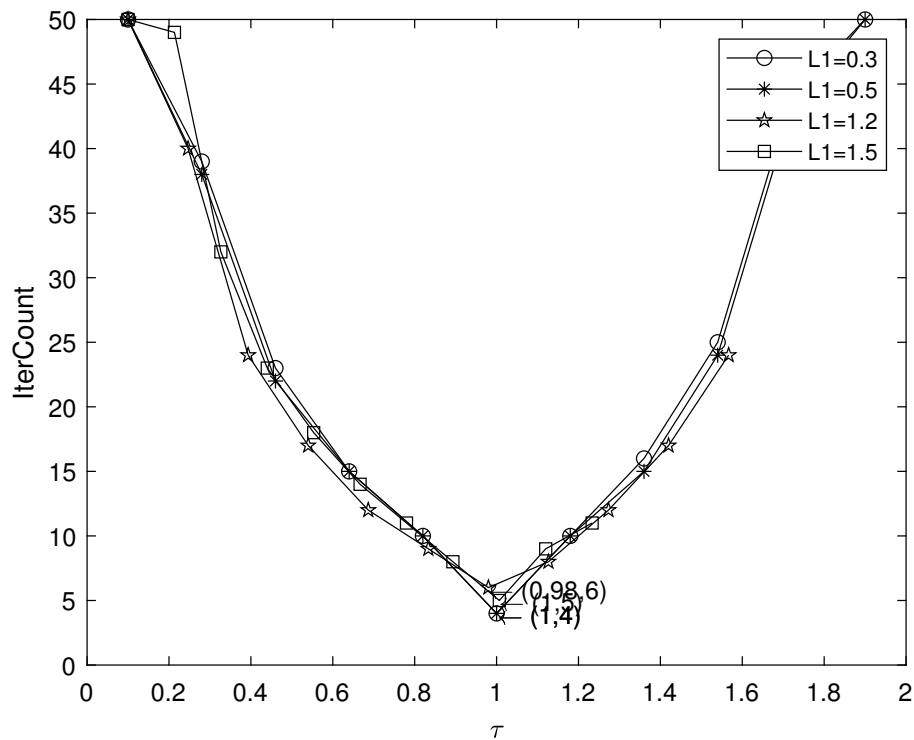


Рисунок 7 — Количество итераций при различных значениях k и τ

жем в таблице 15 полученные оптимальные значения. Видим по рисунку 7 и

Таблица 15 Оптимальное значение τ при различных значениях k

k	Оптимальное τ	Теоретическое τ
0.3	1	1.53
0.5	1	1.33
1.2	0.98	0.90
1.5	1.006	0.80

таблице 15, что интервал сходимости и оптимальные значения параметра меняются в зависимости от значения k , что близко к теоретическим значениям. Код для задачи с точечным источником расположен в приложении Б.

ЗАКЛЮЧЕНИЕ

В ходе выполнения выпускной квалификационной работе, посвященной созданию комплекса программ для решения нелинейных стационарных краевых задач Дирихле, мы достигли следующих результатов:

- Получили решение задачи, (1),(2) с гладкой правой частью,
- Получили решение модельной задачи с особенностью в правой части (18),(19),
- Построили комплекс программ для решения этих задач,
- Нашли оптимальные значения итерационного параметра и интервал сходимости для задачи с особенностью.

Все поставленные задачи и цели выпускной квалификационной работы были выполнены.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

- 1 Задворнов О. А. Существование решения квазилинейной эллиптической краевой задачи при наличии точечных источников // Учен. зап. Казан. ун-та. Сер. Физ.-матем. науки. 2010. Т. 152. № 1. С. 155–163.
- 2 О. А. Задворнов, Г. О. Задворнова, О решении нелинейной стационарной неоднородной задачи фильтрации при наличии точечного источника, статья в российском журнале // Дифференциальные уравнения, 2014, том 50, № 7, с. 984-988
- 3 Трифонова Г.О. Итерационный метод решения нелинейной краевой задачи с точечным источником // Труды 64-й Всероссийской научной конференции МФТИ. - М.: МФТИ, 2021. - С. 43-44.
- 4 Бахвалов Н.С., Жидков Н.П., Кобельков Г.М. Численные методы. – М.: Бином. Лаборатория знаний, 2012. – 598с.
- 5 Владимиров В.С. Уравнения математической физики. М.: Наука, 1988. - 348 с.
- 6 Гаевский Х., Греггер К., Захариас К. Нелинейные операторные уравнения и операторные дифференциальные уравнения / М.: Мир, 1978. 336 с.
- 7 Даутов Р.З., Карчевский М.М. Введение в теорию метода конечных элементов. Учебное пособие. — Казань: Казанский государственный университет им. В.И. Ульянова–Ленина, 2004. — 239 с.
- 8 Сьярле Ф. Метод конечных элементов для эллиптических задач. — М.: Мир, 1980.
- 9 Finite Element Method Basics // MathWorks URL: <https://www.mathworks.com/help/releases/R2021b/pde/ug/basics-of-the-finite-element-method.html>
- 10 Mesh Data // MathWorks URL: <https://www.mathworks.com/help/releases/R2021b/pde/ug/mesh-data.html>
- 11 Help Matlab раздел Partial Differential Equation Toolbox — Examples.

- 12 Львовский С. М. Набор и вёрстка в пакете LATEX. — 3-е изд, 2003.
- 13 PDESolverOptions Properties // MathWorks URL:
[https://www.mathworks.com/help/releases/R2021b/pde/ug/pde.pdesolver-
options-properties.html](https://www.mathworks.com/help/releases/R2021b/pde/ug/pde.pdesolver-options-properties.html)

ПРИЛОЖЕНИЕ А.

(обязательное)

Листинг кода для задачи с гладкой правой частью

Скрипт maintestPDE.

```
clear all; close all; clc
k = 0.5; c0=10;
R = 3;
h = 0.1;
a = 0;
f=c0;
g=@(ug_W) k + (1-k)*(ug_W>1).*(1-1./(ug_W+(ug_W<0.0000001)));
r=@(x,y) sqrt(x.^2+y.^2);
gd = pdegeomKrg(R);
model = createpde;
geometryFromEdges(model,gd);
%Применение solvepde для нелинейной задачи
%Коэффициент c
c = @(~, state) g(r(state.ux, state.uy));
specifyCoefficients(model,'m',0,'d',0,'c',c,'a',a,'f',f);
%Граничные условия
applyBoundaryCondition(model,'dirichlet',...
'Edge',1:model.Geometry.NumEdges,...
'u',0);
mesh_linear = generateMesh(model,'GeometricOrder','linear', 'hmax',h);
model.SolverOptions.ReportStatistics = 'on';
mbs = tic;
result = solvepde(model);
```

```

Uspde = result.NodalSolution;
mbst = toc(mbs);
%Точное решение
[p,e,t] = meshToPet(model.Mesh);
x=p(1,:); y=p(2,:);
u_exact=Tuu(x,y, k, c0, R)';
first_err=norm(u_exact-Uspde,inf);
disp('u_exact-Uspde')
disp(first_err)
figure
pdesurf(p,t, abs(u_exact-Uspde))
xlabel('x_1')
ylabel('x_2')
zlabel('Погрешность решения |u_exact-Uspde|')
colormap('hsv')
%Применение метода простой итерации
c=1;
specifyCoefficients(model,'m',0,'d',0,'c',c,'a',a,'f',f);
applyBoundaryCondition(model,'dirichlet',...
'Edge',1:model.Geometry.NumEdges,...
'u',0);
model.SolverOptions.ReportStatistics = 'on';
mn = tic;
FEMn = assembleFEMatrices(model,'nullspace');
u0 = FEMn.B*(FEMn.Kc\FEMn.Fc) + FEMn.ud;
% Итерационный процесс
% assembF - функция сборки правой части
% Uk0 - начальное приближение
eps = 1e-5;

```

```

counter = 1;Uk0 = u0;
tau=2./(1+k);
%tau=1;
while true
F = assembF(p,t,g,Uk0);
FO = FEMn.B'*F-FEMn.Fc;
w = FEMn.B*(FEMn.Kc\F0);
w= -w*tau;
Uk0 = Uk0 +w;
counter = counter + 1;
if norm(w,'inf') < eps | counter == 100
break
end
end
mnst = toc(mn);
UIt = Uk0;
second_err = norm(u_exact - UIt, 'inf');
disp('u_exact - UIt')
disp(second_err)
disp('Итераций')
disp(counter)
figure
pdesurf(p,t, abs(u_exact - UIt))
xlabel('x_1')
ylabel('x_2')
zlabel('Погрешность решения |u_exact - UIt|')
colormap('hsv')
%Сравнение решения solverpde и простой итерации
second2_err = norm(Uspde - UIt, 'inf');

```

```

disp('Uspde - UIt')
disp(second2_err)
figure
pdesurf(p,t, abs(Uspde - UIt))
xlabel('x_1')
ylabel('x_2')
zlabel('Погрешность решения |Uspde - UIt|')
colormap('hsv')
figure
subplot(1,2,1)
pdesurf(p,t, u_exact)
xlabel('x_1')
ylabel('x_2')
zlabel('Точное решение')
colormap('hsv')
subplot(1,2,2)
pdesurf(p,t, UIt)
xlabel('x_1')
ylabel('x_2')
zlabel('Приближенное решение')
colormap('hsv')
%
%
figure;
subplot(2,2,1)
s=(0:0.01:R);
UG=Tdu(s,0,k,c0);
plot(s,UG,s,s*0+1);
xlabel 'r'

```

```

ylabel '| grad u |'
title('Значение градиента в зависимости от r')
subplot(2,2,2)
s=(0:0.01:R);
U=Tuu(s,0,k,c0,R);
plot(s,U);
xlabel 'r'
ylabel 'u'
title('Точное решение в зависимости от r')
f=@(s) c0*ones(size(s));
subplot(2,2,3)
s=(0:0.01:R);
%U=c0*ones(size(s));
U=f(s);
plot(s,U);
xlabel 'r'
ylabel 'f'
title('Значение правой части в зависимости от r')
subplot(2,2,4)
s=(0:0.01:R);
U=g(s).*s;
plot(s,U);
xlabel 'r'
ylabel 'g'
title('Скорость фильтрации в зависимости от r')

```

Скрипт assembF.

```

function[F] = assembF(p,t,g,Uk0)
np=size(p,2);

```

```

i1=t(1,:);i2=t(2,:);i3=t(3,:);
sdl=t(4,:);
% Градиент базисных функций
g1x=p(2,i2)-p(2,i3);g2x=p(2,i3)-p(2,i1);g3x=p(2,i1)-p(2,i2);
g1y=p(1,i3)-p(1,i2);g2y=p(1,i1)-p(1,i3);g3y=p(1,i2)-p(1,i1);
% Якобиан
J=abs(g3y.*g2x-g3x.*g2y);
% Градиент Uk0
Uk0 = Uk0';
ux_Uk0 = Uk0(i1) .* g1x + Uk0(i2) .* g2x + Uk0(i3) .* g3x;
uy_Uk0 = Uk0(i1) .* g1y + Uk0(i2) .* g2y + Uk0(i3) .* g3y;
gradUk0 = [ux_Uk0; uy_Uk0];
gradW = gradUk0;
ug_W = sqrt(gradW(1,:).^2 + gradW(2,:).^2)./J;
cu = feval(g,ug_W);
fx = (cu.*gradW(1,:))./(2*J); fy = (cu.*gradW(2,:))./(2*J);
ff = fx.*g1x + fy.*g1y;
F=sparse(i1,1,ff,np,1);
ff = fx.*g2x + fy.*g2y;
F=F+sparse(i2,1,ff,np,1);
ff = fx.*g3x + fy.*g3y;
F=F+sparse(i3,1,ff,np,1);
end

```

Скрипт pdegeomKrg.

```

function gd = pdegeomKrg(R)
% Circle is code 4
% Эллипс
% координаты центра (x1,y1)

```

```

% полуоси rx и ry
% угол поворота проти часовой стрелки fi
x1=0; y1=0;
rx=R; ry=R;
fi=0;
R1 = [4,x1,y1,rx,ry,fi]';
% Names for the geometric objects
ns = (char('R1'))';
% Set formula
sf = 'R1';
% Create geometry
gd = decsg(R1,sf,ns);
end

```

Скрипт Tdu.

```

function [z ] = Tdu(x,y,k,c0)
%UNTITLED Summary of this function goes here
% Detailed explanation goes here
r = sqrt(x.^2+y.^2);
ss=c0/2;
k0=1;
u1=@(s) s/k;
u2=@(s) s/k0+(k0-k)/k0;
u=@(s) u1(s).*(s<k)+u2(s).*(s>=k);
z = u(ss.*r);
end

```

Скрипт Tuu.

```

function z = Tuu(x,y,k,c0,R)

```

```

r = sqrt(x.^2+y.^2);
ss=(2*k/c0);
k0=1;
u1=@(s) c0/(4*k)*(ss.^2-s.^2)+c0/(4*k0)*(R.^2-ss.^2)+(k0-k)/k0*(R-ss);
u2=@(s) c0/(4*k0)*(R.^2-s.^2)+(k0-k)/k0*(R-s);
u=@(s) u1(s).*(s<ss)+u2(s).*(s>=ss);
z = u(r);
end

```


ПРИЛОЖЕНИЕ Б.

(обязательное)

Листинг кода для задачи с особенностью в правой части
Скрипт Qmain.

```
clear all; close all; clc
R = 1; L1= 0.5; L2 =1; q = 100;
g = @(s) L1 + (1-L1)*(s>1).*(1-1./(s+(s<0.0000001)));
c = 1; a = 0; f = 0;
eps = 1e-5; h = 0.1; miter = 50;

% Create the geometry
gd = pdegeomKrg(R);
% Create the model
model = createpde;
% Geometry include it in the model.
geometryFromEdges(model,gd);

qV=[10 40 160 640 ];
Col=["co-" "b*-" "gp-" "rs-"];
fileID = fopen('Qtest.txt','w');
figure
ax1 = axes;
figure
ax2 = axes;
for i=1: numel(qV)
q=qV(i);

% Plot the geometry with the edge and face labels.
```

```

% pdegplot(model,'EdgeLabels','on','FaceLabels','on');
% axis equal
% title 'Geometry with Edge and Face Labels';

% Specify the boundary conditions
bcMatrix = @(region,~) uT(region.x,region.y,q,R,L1);

applyBoundaryCondition(model,'dirichlet',...
'Edge',1:model.Geometry.NumEdges,...
'u',bcMatrix);
% Specify the Coefficients
specifyCoefficients(model,'m',0,'d',0,'c',1,'a',0,'f',0);
lin_mesh = generateMesh(model,'GeometricOrder','linear', 'Hmax',h);

[p,e,t] = meshToPet(model.Mesh);
x=p(1,:); y=p(2,:);
% u_ex - точное решение
u_ex = uT(x,y,q,R,L1)';
phi_ex = phi(x,y,R,q)';
% нахождение u0
FEMn = assembleFEMatrices(model,'nullspace');
u0 = FEMn.B*(FEMn.Kc\FEMn.Fc) + FEMn.ud;
Btau=2/(L1+L2);

% Итерационный процесс
% assembF - функция сборки правой части
% Uk0 - начальное приближение
% w0 =  $\Phi$  + u0
counter = 1;Uk0 = u0; w0 = phi_ex + u0;

```

```

while true
F = assembF(p,t,g,Uk0,w0);
FO = FEMn.B'*F;
r = FEMn.B*(FEMn.Kc\F0);
r= -r*Btau;
Uk0 = Uk0 +r;
counter = counter + 1;
if norm(r,'inf') < eps | counter == miter
break
end
end
% Bcont=counter;
% Погрешность решения
norm(Uk0-u_ex,inf)

% figure
% pdeplot(model,'XYData',Uk0,'ZData',Uk0);
% title(['u_h, h=' num2str(h)])
% figure
% pdeplot(model,'XYData',abs(Uk0-u_ex),'ZData',abs(Uk0-u_ex));
% title(['||uT-u_h||, h=' num2str(h)])

% Построение графика зависимости итерации от значения tau
LL=2/max([L1 L2]); % (1, LL) интервал сходимости Ит Мет
tau_arr=linspace(0.1,LL-0.1,11);
tau_it_count = []; norm_it_count = [];
counter = 1;
for tau = tau_arr
counter = 1; Uk0 = u0; w0 = phi_ex + u0;

```

```

while true
F = assembF(p,t,g,Uk0,w0);
FO = FEMn.B'*F;
r = FEMn.B*(FEMn.Kc\F0);
r= -r*tau;
Uk0 = Uk0 + r;
counter = counter + 1;
if norm(r) < eps | counter == miter
break
end
end
norm_it_count = [norm_it_count norm(Uk0-u_ex,inf)];
tau_it_count = [tau_it_count counter];
end

[minIter xtau]=min(tau_it_count);
[maxIter]=max(tau_it_count);

Pr=[tau_arr; tau_it_count; norm_it_count];
fprintf(fileID,'BestTau= %5.4f & q= %5.4f \n',Btau,q);
fprintf(fileID,'%5s & %4s & %11s \\\ \n','tau','IterCount','Uerror');
fprintf(fileID,'%3.3f & %4.0f & %8.5e \\\ \n',Pr);

% figure
% plot(tau_arr,tau_it_count,'-ob')
% hold on
% plot(Btau,Bcont,'-or')
% title(['tau|iter, h=' num2str(h)])
% saveas(gcf,'tau_it_count.jpg')

```

```

%
% figure
% plot(tau_arr,norm_it_count,'-ob')
% xlabel('\tau')
% ylabel(['||uT-u_h||', h=' num2str(h)])
% saveas(gcf,'norm_it_count.jpg')

axes(ax1)
plot(tau_arr,tau_it_count,Col(i),'DisplayName',['q=' num2str(q)])
text(tau_arr(xtau),minIter,['\leftarrow (' num2str(tau_arr(xtau),2) ',
'HorizontalAlignment','left')
hold on
%saveas(gcf,'ris6.jpg')

axes(ax2)
plot(tau_arr,norm_it_count,Col(i),'DisplayName',['q=' num2str(q)])
hold on
end

axes(ax1)
set(gca,'YLim',[minIter-1 maxIter+1],'YTick',[minIter:maxIter])
set(gca,'XLim',[tau_arr(1) tau_arr(end)],'XTick',tau_arr)
xlabel('\tau')
ylabel('IterCount')
legend
saveas(gcf,'Qiter.eps')
axes(ax2)
xlabel('\tau')
ylabel('||u-u_h||')

```

```
legend
saveas(gcf,'Qerror.eps')
```

Скрипт Qmain_ln_err.

```
% Построение графика зависимости погрешности от шага
clear all; close all; clc
R = 1; L1= 0.5; L2 =1; Btau=2/(L1+L2);
g = @(s) L1 + (1-L1)*(s>1).*(1-1./(s+(s<0.0000001))); q = 100;
c = 1; a = 0; f = 0;
eps = 1e-5; h = 0.1; miter = 50;

% Create the geometry
gd = pdegeomKrg(R);
% Create the model
model = createpde;
% Geometry include it in the model.
geometryFromEdges(model,gd);

qV=[10 40 160 640 ];
Col=["co-" "b*-" "gp-" "rs-"];
fileID = fopen('Qtest_log_err.txt','w');
figure
ax3 = axes;
for i=1:numel(qV)
q=qV(i);

bcMatrix = @(region,~) uT(region.x,region.y,q,R,L1);
applyBoundaryCondition(model,'dirichlet',...
```

```

'Edge',1:model.Geometry.NumEdges,...
'u',bcMatrix);
% Specify the Coefficients
specifyCoefficients(model,'m',0,'d',0,'c',1,'a',0,'f',0);

err_arr=[]; h_it_count=[];
h = 0.1;
h_arr = [h h/2 h/4 h/8 h/16];
for h = h_arr

lin_mesh = generateMesh(model,'GeometricOrder','linear', 'Hmax',h);
[p,e,t] = meshToPet(lin_mesh);
x=p(1,:); y=p(2,:);
u_ex = uT(x,y,q,R,L1)';
phi_ex = phi(x,y,R,q)';

FEMn = assembleFEMatrices(model,'nullspace');
u0 = FEMn.B*(FEMn.Kc\FEMn.Fc) + FEMn.ud;

counter = 1; Uk0 = u0; w0 = phi_ex + u0;
while true
F = assembF(p,t,g,Uk0,w0);
F0 = FEMn.B'*F;
r = FEMn.B*(FEMn.Kc\F0);
r=- r*Btau;
Uk0 = Uk0 + r;
counter = counter + 1;
if norm(r) < eps | counter == miter
break

```

```

end
end
h_it_count=[h_it_count counter];
err_arr = [err_arr norm(Uk0 - u_ex, "inf")];
end
c=log(err_arr)./log(h_arr);

Pr=[h_it_count; h_arr; err_arr; c];
fprintf(fileID,'q= %5.4f \n', q);
fprintf(fileID,'%5s & %6s & %11s & %5s\\\\ \n','Iter','h','error','c')
fprintf(fileID,'%5.0f & %5.4f & %8.5e & %3.3f\\\\ \n',Pr);

axes(ax3)
plot(log(h_arr),log(err_arr),Col(i),'DisplayName',['q=' num2str(q)])
hold on
end
axes(ax3)
xlabel('ln(h)')
ylabel('|ln(|u-u_h|)|')

legend
axis equal
saveas(gcf,'Qln(err_arr).eps')

Скрипт L1main.

clear all; close all; clc
R = 1; L2 =1; q = 100; miter = 50;
c = 1; a = 0; f = 0;
eps = 1e-5; h = 0.1;

```



```

% Create the geometry
gd = pdegeomKrg(R);
% Create the model
model = createpde;
% Geometry include it in the model.
geometryFromEdges(model,gd);

L1V=[0.3 0.5 1.2 1.5];
Col=["co-" "b*-" "gp-" "rs-"];
fileID = fopen('L1test.txt','w');
figure
ax1 = axes;
figure
ax2 = axes;
for i=1:numel(L1V)
L1=L1V(i);
g = @(s) L1 + (1-L1)*(s>1).*(1-1./(s+(s<0.0000001)));
% Plot the geometry with the edge and face labels.
% pdegplot(model,'EdgeLabels','on','FaceLabels','on');
% axis equal
% title 'Geometry with Edge and Face Labels';

% Specify the boundary conditions
bcMatrix = @(region,~) uT(region.x,region.y,q,R,L1);

applyBoundaryCondition(model,'dirichlet',...
'Edge',1:model.Geometry.NumEdges,...
'u',bcMatrix);

```

```

% Specify the Coefficients
specifyCoefficients(model,'m',0,'d',0,'c',1,'a',0,'f',0);
lin_mesh = generateMesh(model,'GeometricOrder','linear', 'Hmax',h);

[p,e,t] = meshToPet(model.Mesh);
x=p(1,:); y=p(2,:);
% u_ex - точное решение
% phi_ex - ?
u_ex = uT(x,y,q,R,L1)';
phi_ex = phi(x,y,R,q)';
% нахождение u0
FEMn = assembleFEMatrices(model,'nullspace');
u0 = FEMn.B*(FEMn.Kc\FEMn.Fc) + FEMn.ud;
Btau=2/(L1+L2);

% Итерационный процесс
% assembF - функция сборки правой части
% Uk0 - начальное приближение
% w0 =  $\Phi$  + u0
counter = 1;Uk0 = u0; w0 = phi_ex + u0;
while true
F = assembF(p,t,g,Uk0,w0);
F0 = FEMn.B'*F;
r = FEMn.B*(FEMn.Kc\F0);
r= -r*Btau;
Uk0 = Uk0 +r;
counter = counter + 1;
if norm(r,'inf') < eps | counter == miter

```

```

break
end
end
% Bcont=counter;
% Погрешность решения
norm(Uk0-u_ex,inf)

% figure
% pdeplot(model,'XYData',Uk0,'ZData',Uk0);
% title(['u_h, h=' num2str(h)])
% figure
% pdeplot(model,'XYData',abs(Uk0-u_ex),'ZData',abs(Uk0-u_ex));
% title(['||uT-u_h||, h=' num2str(h)])

% Построение графика зависимости итерации от значения tau
LL=2/max([L1 L2]); % (1, LL) интервал сходимости Ит Мет
tau_arr=linspace(0.1,LL-0.1,11);
tau_it_count = []; norm_it_count = [];
counter = 1;
for tau = tau_arr
counter = 1; Uk0 = u0; w0 = phi_ex + u0;
while true
F = assembF(p,t,g,Uk0,w0);
F0 = FEMn.B'*F;
r = FEMn.B*(FEMn.Kc\F0);
r= -r*tau;
Uk0 = Uk0 + r;
counter = counter + 1;
if norm(r) < eps | counter == miter

```

```

break
end
end
norm_it_count = [norm_it_count norm(Uk0-u_ex,inf)];
tau_it_count = [tau_it_count counter];
end

[minIter xtau]=min(tau_it_count);
[maxIter]=max(tau_it_count);

Pr=[tau_arr; tau_it_count; norm_it_count];
fprintf(fileID,'BestTau= %5.4f & L1= %5.4f \n',Btau,L1);
fprintf(fileID,'%5s & %4s & %11s \\\ \n','tau','IterCount','Uerror');
fprintf(fileID,'%3.3f & %4.0f & %8.5e \\\ \n',Pr);

% figure
% plot(tau_arr,tau_it_count,'-ob')
% hold on
% plot(Btau,Bcont,'-or')
% title(['tau|iter, h=' num2str(h)])
% saveas(gcf,'tau_it_count.jpg')
%
% figure
% plot(tau_arr,norm_it_count,'-ob')
% xlabel('\tau')
% ylabel(['||uT-u_h||', h=' num2str(h)])
% saveas(gcf,'norm_it_count.jpg')

axes(ax1)

```

```

plot(tau_arr,tau_it_count,Col(i),'DisplayName',['L1=' num2str(L1)])
text(tau_arr(xtau),minIter,['\leftarrow (' num2str(tau_arr(xtau),2) ',
'HorizontalAlignment','left')
hold on
%saveas(gcf,'ris6.jpg')

axes(ax2)
plot(tau_arr,norm_it_count,Col(i),'DisplayName',['L1=' num2str(L1)])
hold on
end

axes(ax1)
xlabel('\tau')
ylabel('IterCount')
legend
saveas(gcf,'L1iter.eps')
axes(ax2)
xlabel('\tau')
ylabel('||w-w_h||')
legend
saveas(gcf,'L1error.eps')

```

Скрипт L1main_ln_err.

```

% Построение графика зависимости % Построение графика зависимости погр
clear all; close all; clc
R = 1; L1 = 1; L2 = 1; q = 100;
c = 1; a = 0; f = 0;
eps = 1e-5; h = 0.1; miter = 50;

```

```

% Create the geometry
gd = pdegeomKrg(R);
% Create the model
model = createpde;
% Geometry include it in the model.
geometryFromEdges(model,gd);

L1V=[0.3 0.5 1.2];
Col=["co-" "b*-" "gp-"];
fileID = fopen('L1test_log_err.txt','w');
figure
ax3 = axes;
for i=1:numel(L1V)
L1=L1V(i);

bcMatrix = @(region,~) uT(region.x,region.y,q,R,L1);
applyBoundaryCondition(model,'dirichlet',...
'Edge',1:model.Geometry.NumEdges,...
'u',bcMatrix);
% Specify the Coefficients
specifyCoefficients(model,'m',0,'d',0,'c',1,'a',0,'f',0);

g = @(s) L1 + (1-L1)*(s>1).*(1-1./(s+(s<0.0000001)));
err_arr=[]; h_it_count=[];
h = 0.1;
h_arr = [h h/2 h/4 h/8 h/16];
for h = h_arr

```

```

lin_mesh = generateMesh(model,'GeometricOrder','linear', 'Hmax',h);
[p,e,t] = meshToPet(lin_mesh);
x=p(1,:); y=p(2,:);
u_ex = uT(x,y,q,R,L1)';
phi_ex = phi(x,y,R,q)';

FEMn = assembleFEMatrices(model,'nullspace');
u0 = FEMn.B*(FEMn.Kc\FEMn.Fc) + FEMn.ud;
Btau=2/(L1+L2);

counter = 1; Uk0 = u0; w0 = phi_ex + u0;
while true
F = assembF(p,t,g,Uk0,w0);
F0 = FEMn.B'*F;
r = FEMn.B*(FEMn.Kc\F0);
r=- r*Btau;
Uk0 = Uk0 + r;
counter = counter + 1;
if norm(r) < eps | counter == miter
break
end
end
h_it_count=[h_it_count counter];
err_arr = [err_arr norm(Uk0 - u_ex, "inf")];
end
c=log(err_arr)./log(h_arr);

Pr=[h_it_count; h_arr; err_arr; c];
fprintf(fileID,'L1= %5.4f \n', L1);

```

```

fprintf(fileID,'%5s & %6s & %11s & %5s\\\\ \n','Iter','h','error','c')
fprintf(fileID,'%5.0f & %5.4f & %8.5e & %3.3f\\\\ \n',Pr);

axes(ax3)
plot(log(h_arr),log(err_arr),Col(i),'DisplayName',['L1=' num2str(L1)])
hold on
end
axes(ax3)
xlabel('ln(h)')
ylabel('|ln(|u-u_h|)|')
legend
axis equal
saveas(gcf,'L1ln(err_arr).eps')

```