

**МОГАТЕЛЬНЫЕ**

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**«БЕЛГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ НАЦИОНАЛЬНЫЙ  
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ»**  
( **Н И У « Б е л Г У »** )

ИНСТИТУТ ИНЖЕНЕРНЫХ ТЕХНОЛОГИЙ И ЕСТЕСТВЕННЫХ НАУК  
ФАКУЛЬТЕТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ И ПРИКЛАДНОЙ МАТЕМАТИКИ  
**Кафедра общей математики**

**ИССЛЕДОВАНИЕ ОДНОЙ МОДЕЛИ ДВИЖЕНИЯ ВЯЗКОЙ ЖИДКОСТИ**

**Выпускная квалификационная работа студента**

**очной формы обучения**  
**направления подготовки 01.03.02 Прикладная математика и информатика**  
**4 курса группы 07001206**  
**Курганов Андрей Викторович**

Научный руководитель  
доцент кафедры  
Некрасова И.В.

Рецензент  
к.ф.-м.н., доц.

**БЕЛГОРОД 2016**

## СОДЕРЖАНИЕ

ВВЕДЕНИЕ.....	3
1 Вспомогательные предложения.....	7
2 Аналитический обзор.....	7
2.1 История применения метода.....	8
2.2 Описание метода.....	14
2.3 Простые примеры применения метода Галеркина.....	18
3 Корректность периодической начально-краевой задачи для уравнения Стокса.....	22
3.1 Постановка задачи.....	22
3.2 Основные результаты.....	23
3.3 Доказательство теоремы 1.....	25
3.4 Доказательство теоремы 2.....	25
3.5 Доказательство теоремы 3.....	25
4 Программная реализация.....	29
4.1 Выбор среды разработки.....	29
4.1.1 Описание языка программирования.....	30
4.2 Описание программной реализации.....	32
4.3 Проверка работоспособности программного продукта.....	45
ЗАКЛЮЧЕНИЕ.....	54
СПИСОК ЛИТЕРАТУРЫ.....	59
ПРИЛОЖЕНИЕ.....	62

## ВВЕДЕНИЕ

Исследование краевых задач для параболических уравнений является одним из важных разделов современной теории дифференциальных уравнений с частными производными. Интерес к этому типу уравнений объясняется как теоретической значимостью получаемых результатов, так и их важными практическими приложениями.

Параболические уравнения и системы уравнений являются основой математических моделей разнообразных процессов: конвекции и диффузии, в том числе уравнение диффузии и его частный случай – уравнение теплопроводности; движения жидкости и газов, описываемых системой уравнений Навье – Стокса; воздействия электромагнитного поля на заряженные частицы.

Исследование методов решения краевых задач для параболических уравнений было рассмотрено в работах О.А. Олейник, О.А. Ладыженской, В.П. Михайлова, В.А. Солонникова, Н.Н. Уральцевой.

**Актуальность** исследований таких краевых задач обоснована как внутренней логикой развития соответствующих разделов теории дифференциальных уравнений в частных производных, так и ясными перспективами использования этих задач при математическом моделировании различных процессов.

В настоящей работе рассматривается периодическая начально – краевая задача для уравнений Стокса. Такого рода задачи возникают при усреднении дифференциальных уравнений, описывающих процессы фильтрации вязкой жидкости в упругом пористом теле, где пористый скелет представляет собой периодическое повторение ячейки малого размера .

Решение периодической начально – краевой задачи на ячейке необходимо для построения коэффициентов усредненной системы уравнений.

Для поставленной задачи определяется понятие обобщенного решения, получены априорные оценки решения, доказывається корректность. Для обоснования существования обобщенного решения задачи используется метод Галёркина.

Методы Галёркина давно применяются для решения дифференциальных уравнений с частными производными.

К настоящему времени они применены при решении многочисленных задач механики конструкций, динамики сооружений, гидромеханики, теории гидродинамической устойчивости, акустики, магнитной гидродинамики, теории теплообмена, теории распространения микроволн, теории переноса нейтронов и т.д.

**Объектом** исследования является нестационарная система Стокса, описывающая движение вязкой жидкости.

**Цель** работы состоит в доказательстве корректности поставленной задачи, и программной реализации с использованием такой среды программирования как Visual Studio Express и языка на котором будет реализованная программа C# .

Для достижения поставленной цели необходимо решить ряд **задач**:

- изучить литературу по проблеме исследования;
- изучить аналитические и численные методы исследования уравнений параболического типа;
- исследовать разрешимость поставленной задачи;
- разработать программу в среде Visual Studio Express.

Выпускная квалификационная работа изложена на 69 страницах, состоит из введения, заключения, четырех разделов, списка использованных источников и приложения.

Введение содержит общие сведения о работе, актуальность выбранной темы, объект, цель и задачи.

Первый раздел имеет вспомогательный характер. В нем изложен ряд теоретических понятий и предложений, используемых в основном тексте при исследовании краевых задач для дифференциальных уравнений.

Во втором разделе проводится краткий обзор исследований, связанных с темой работы дается краткое описание схемы метода Галеркина, приведен ряд примеров, иллюстрирующих применение метода.

В третьем разделе рассматривается начально – краевая периодическая задача, моделирующая движение вязкой жидкости. Приводится определение обобщенного решения задачи. Получены априорные оценки решения. Доказана однозначная разрешимость задачи.

В четвертом разделе описывается среда разработки программного приложения и язык программирования, программная реализация.

Разработана программа для численного решения поставленной задачи. Здесь мы ограничимся рассмотрением дифференциального уравнения с частными производными параболического типа, когда эти уравнения являются линейными, а искомая функция зависит от двух переменных. В качестве области моделирования рассмотрен канал с квадратным поперечным сечением.

В заключении отражены общие выводы по работе, значимость разработки, достигнутые задачи.

В приложении представлен текст программы и результаты выполнения тестовых расчетов.

## 1 Вспомогательные предложения

Этот раздел имеет вспомогательный характер. В нем мы изложим ряд понятий и предложений функционального анализа, которые используются в основном тексте при исследовании краевых задач для дифференциальных уравнений.

*Областью* в пространстве  $R^n$  называется открытое множество  $\Omega$  (как правило связное), граница  $\Gamma$  которой локально представима в виде

$$y_n = f^{(k)}(y_1, \dots, y_{n-1}). \quad (1.1)$$

Напомним, что границей области называется множество  $\Gamma = \bar{\Omega} \setminus \Omega$ , где множество  $\bar{\Omega}$  есть замыкание множества  $\Omega$ . Локальность означает, что границу  $\Gamma$  можно погрузить в объединение областей  $\Omega_k$ ,  $k = 1, 2, \dots, m$ , так что в каждой из областей  $\Omega_k$  найдется ортогональная система координат  $\mathbf{y} = (y_1, \dots, y_n)$ , в которой граница представима в виде (1.1). Если функции  $f^{(k)}$   $m$  – раз непрерывно дифференцируемы, то граница  $\Gamma$  называется границей класса  $C^m$ .

Пусть граница  $\Gamma$  области  $\Omega$  является границей класса  $C^1$ . Тогда в каждой точке границы определен вектор внешней нормали  $\mathbf{v} = (v_1, \dots, v_n)$  и для всякой непрерывно дифференцируемой в замыкании области  $\Omega$  функции  $u(\mathbf{x})$  справедлива формула *Гаусса – Остроградского*

$$\int_{\Omega} \frac{\partial u}{\partial x_i} dx = \int_{\Gamma} uv_i ds. \quad (1.2)$$

см. Олейник [3, с.14].

Пусть непрерывная в замыкании области  $\Omega$  функция  $u$  непрерывно дифференцируемо зависит от параметра  $\tau$ :  $u = u(\mathbf{x}, \tau)$ . Тогда интеграл

$$I(\tau) = \int_{\Omega} u(x, \tau) dx$$

есть непрерывно дифференцируемая функция параметра  $\tau$  и справедлива формула дифференцирования интеграла по параметру

$$\frac{dI}{d\tau} = \int_{\Omega} \frac{du}{d\tau}(x, \tau) dx. \quad (1.3)$$

Для одной пространственной переменной справедлива формула дифференцирования по верхнему и нижнему пределу:

$$\frac{d}{d\tau} \int_{a(\tau)}^{b(\tau)} u(x) dx = u(b(\tau)) \frac{db}{d\tau}(\tau) - u(a(\tau)) \frac{da}{d\tau}(\tau). \quad (1.4)$$

Линейное пространство  $B$  называется линейным *нормированным* пространством, если для каждого элемента  $x \in B$  определено неотрицательное число  $\|x\|$ , называемое нормой элемента  $x$ , такое что

- 1)  $\|x\| = 0 \Rightarrow x = 0$ ,
  - 2)  $\|\lambda x\| = |\lambda| \|x\|, \quad \forall \lambda \in R,$
  - 3)  $\|x + y\| \leq \|x\| + \|y\|.$
- (1.5)

Последовательность элементов  $\{x_m\}$  называется *сходящейся* в  $B$ , если в  $B$  найдется элемент  $x$  такой, что  $\|x_m - x\| \rightarrow 0$  при  $m \rightarrow \infty$ . Будем обозначать сходимость последовательности элементов  $\{x_m\}$  к элементу  $x$  символом  $x_m \rightarrow x$ .

Последовательность элементов  $\{x_m\}$  называется *фундаментальной* в  $B$ , если для всякого положительного  $\varepsilon$  найдется номер  $N$ , такой что при  $m, n > N$  будет выполнено неравенство  $\|x_m - x_n\| < \varepsilon$ .

Нормированное пространство называется *полным (Банаховым)*, если всякая фундаментальная последовательность является сходящейся.

Множество  $M$  называется всюду плотным в пространстве  $B$ , если для всякого элемента  $x \in B$  найдется последовательность элементов  $\{x_m\}$  из  $M$ , сходящаяся к элементу  $x$ .

Нормированное пространство  $B$  называется *сепарабельным*, если в нем существует счетное всюду плотное множество.

Линейное пространство  $E$  называется *Евклидовым*, если в нем для каждой пары элементов  $x, y \in E$  определено число  $(x, y)$ , называемое *скалярным произведением*, такое что

- 1)  $(x, x) \geq 0, (x, x) = 0 \Leftrightarrow x = 0,$
- 2)  $(x, y) = (y, x),$  (1.6)
- 3)  $(\lambda x, y) = \lambda(x, y) \forall \lambda \in R.$

Всякое Евклидово пространство является нормированным, если положить  $\|x\| = (x, x)^{\frac{1}{2}}.$

Таким образом, полное сепарабельное евклидово пространство называется *Гильбертовым* пространством.

Два элемента  $x, y \in E$  называются *ортогональными*, если  $(x, y) = 0.$

Множество  $M$  называется *ортонормированным*, если оно ортогонально и все его элементы имеют норму равную единице.

Счетное ортонормированное множество  $\{e_1, e_2, \dots\}$  называется *базисом* в пространстве  $E$ , если для всякого элемента  $x \in E$

$$S_m = \sum_{k=1}^m c_k e_k \rightarrow x \text{ при } m \rightarrow \infty,$$

где  $c_k = (x, e_k).$



**Теорема 1.** В Гильбертовом пространстве существует хотя бы один базис.

Последовательность элементов  $\{x_m\}$  Евклидова пространства  $E$  называется слабо сходящейся к элементу  $x \in E$ , если  $(x_m - x, y) \rightarrow 0$ ,  $m \rightarrow \infty$  для всех  $y \in E$ .

Множество  $M$  Евклидова пространства  $E$  называется слабо компактным, если всякая последовательность элементов этого множества содержит слабо сходящуюся подпоследовательность.

**Теорема 2.** В Гильбертовом пространстве всякое замкнутое ограниченное множество является слабо компактным.

Совокупность всех измеримых функций  $u(x)$ ,  $x \in \Omega \subset R^n$ , с конечным интегралом

$$\|u\|_p = \left( \int_{\Omega} |u(x)|^p dx \right)^{\frac{1}{p}} \quad (1.7)$$

для  $p \geq 1$  образует Банахово пространство  $L_p(\Omega)$ , если норму в нем определить равенством (1.7). Доказательство этого факта базируется на *неравенстве Коши*

$$\left| \int_{\Omega} u(x)v(x) dx \right| \leq \left( \int_{\Omega} |u(x)|^2 dx \right)^{\frac{1}{2}} \left( \int_{\Omega} |v(x)|^2 dx \right)^{\frac{1}{2}} \quad (1.8)$$

для  $p = 2$ , и его обобщении – неравенстве Гёльдера

$$\left| \int_{\Omega} u(x)v(x) dx \right| \leq \left( \int_{\Omega} |u(x)|^p dx \right)^{\frac{1}{p}} \left( \int_{\Omega} |v(x)|^q dx \right)^{\frac{1}{q}}, \quad \frac{1}{p} + \frac{1}{q} = 1, \quad (1.9)$$

для  $p \neq 2$ , см. Ладыженская [4, с. 35].

Как известно, если нормированное (Евклидово) пространство не полное, то существует полное нормированное (Евклидово) пространство, в котором данное пространство будет всюду плотным. Эта операция называется *замыканием* неполного пространства.

Для определения пространства Соболева  $W_2^1(\Omega)$  рассмотрим множество всех непрерывно дифференцируемых функций  $u(x)$  в области  $\Omega$  с нормой

$$\|u\|_2^{(1)} = \left( \int_{\Omega} (|u|^2 + |\nabla u|^2) dx \right)^{\frac{1}{2}}. \quad (1.10)$$

Замыкание этого пространства в норме (1.10) называется пространством Соболева  $W_2^1(\Omega)$ , см. Ладыженская [3, с. 54].

Функция  $u(x)$  из пространства  $L_1(\Omega)$  обладает в области  $\Omega$  обобщенной производной  $v_i \in L_1(\Omega)$  по переменной  $x_i$ , если

$$\int_{\Omega} \left( u \frac{\partial \varphi}{\partial x_i} + v_i \varphi \right) dx = 0$$

для произвольной бесконечно дифференцируемой функции  $\varphi$ , равной нулю на границе области  $\Omega$ . Как обычно, функцию  $v_i$  обозначим общепринятым символом частной производной:  $v_i = \partial u / \partial x_i$ .

Можно показать, что элементами пространства  $W_2^1(\Omega)$  являются все функции  $u \in L_2(\Omega)$  имеющие обобщенные производные  $\partial u / \partial x_i \in L_2(\Omega)$ ,  $i = 1, 2, \dots, n$ .

**Теорема 3.** *Пространство  $W_2^1(\Omega)$  является Гильбертовым пространством.*

**Линейные дифференциальные уравнения второго порядка параболического типа**

Линейным параболическим уравнением второго порядка называется дифференциальное уравнение

$$\frac{\partial u}{\partial t} = \sum_{i,j=1}^n a_{ij}(x,t) \frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_{i=1}^n a_i(x,t) \frac{\partial u}{\partial x_i} + a(x,t)u + f(x,t), \quad (1.11)$$

в котором функции  $a_{ij}(x,t)$ ,  $a_i(x,t)$ ,  $a(x,t)$ ,  $i, j = 1, 2, \dots, n$  и  $f(x,t)$  считаются заданными, матрица  $a_{ij}(x,t)$  симметрична и строго положительно определена:

$$\sum_{i,j=1}^n a_{ij}(x,t) \xi_i \xi_j \geq \alpha_0 |\xi|^2 > 0, \quad \alpha_0 = \text{const}, \quad |\xi|^2 = \xi_1^2 + \dots + \xi_n^2. \quad (1.12)$$

Условие (1.12) называют *условием строгой параболичности* для уравнения (1.11), см. Михайлов [5, с. 339].

Частным случаем уравнения (1.11) является *уравнение теплопроводности*

$$\frac{\partial u}{\partial t} = \kappa \sum_{i=1}^n \frac{\partial^2 u}{\partial x_i^2} \equiv \kappa \Delta u. \quad (1.13)$$

Положительная постоянная  $\kappa$  называется *коэффициентом температуропроводности*.

Для нахождения решения уравнения (1.11) в пространстве  $R^n$  характерна задача Коши, когда дополнительно к уравнению (1.11) в пространстве  $R^n$  при  $t > 0$  задается начальное условие при  $t = 0$ :

$$u(x,0) = u_0(x). \quad (1.14)$$

Задача (1.11), (1.14) называется *задачей Коши*.

Если решение уравнения (1.11) ищется в некоторой ограниченной области  $\Omega \subset R^n$ , то дополнительно к начальному условию

$$u(x,0) = u_0(x), \quad x \in \Omega, \quad (1.15)$$

нужно задать краевые условия на границе  $\Gamma$  области  $\Omega$ . В частности, краевое условие

$$u(x, t) = u^0(x, t), \quad x \in \Gamma, \quad t > 0, \quad (1.16)$$

называется первым краевым условием, или условием Дирихле. Соответственно, задача (1.11), (1.15), (1.16) называется первой начально-краевой задачей.

Если вместо условия Дирихле (1.16) рассмотреть на границе  $\Gamma$  краевое условие

$$\nabla u(x, t) \cdot \mathbf{v}(x) \equiv \frac{\partial u}{\partial \nu} = v^0(x, t), \quad x \in \Gamma, \quad t > 0, \quad (1.17)$$

где  $\mathbf{v} = (v_1, \dots, v_n)$  - единичный вектор внешней нормали к границе  $\Gamma$ , которое называется *вторым краевым условием* или *условием Неймана*, то соответствующая задача (1.11), (1.15), (1.17) называется *второй начально-краевой задачей*.

Наконец, если на границе  $\Gamma$  для определения решения уравнения (1.11) задается третье краевое условие

$$\frac{\partial u}{\partial \nu} + b(x, t)u = v^1(x, t), \quad x \in \Gamma, \quad t > 0, \quad (1.18)$$

то задача (1.11), (1.15), (1.19) называется *третьей начально-краевой задачей*.

## 2 АНАЛИТИЧЕСКИЙ ОБЗОР

### 2.1 История применения метода

Целый ряд методов, предназначенных для решения самых разнообразных задач математической физики и техники, базируется на идеях советских ученых И.Г. Бубнова (1913) и Б.Г. Галеркина (1915). Они оказались чрезвычайно плодотворными после перехода к массовому использованию ЭВМ в научно-инженерной практике. Идеи Бубнова-Галеркина помогли развитию вариационных методов, методов взвешенных невязок, наконец, способствовали разработке метода конечных элементов и таких его многочисленных модификаций, как, например, спектральный метод или метод граничных элементов.

Методы Галеркина к настоящему времени применены при решении многочисленных задач механики конструкций, динамики сооружений, гидромеханики, теории гидродинамической устойчивости, акустики, магнитной гидродинамики, теории теплообмена, теории распространения микроволн, теории переноса нейтронов и т.д.

Применение метода к исследованию задач устойчивости гидродинамических течений было реализовано Г. И. Петровым, который доказал сходимость метода Галёркина для отыскания собственных значений широкого класса уравнений, включая уравнения для неконсервативных систем, такие, как например уравнения колебаний в вязкой жидкости [8].

В гидродинамике наиболее эффективно метод Галёркина работает в задачах о конвекции, в силу их самосопряжённости. Задачи о течениях таковыми не являются, и сходимость метода при неудачном выборе базиса может быть сильно затруднена.

На основе подходов Галеркина проводятся исследования обыкновенных дифференциальных уравнений, уравнений в частных производных и интегральных уравнений; стационарные и нестационарные задачи, а также задачи

на собственные значения. По существу, задача, для которой можно выписать определяющее уравнение, может быть решена с помощью одной из разновидностей метода Галеркина.

Метод Галеркина был хорошо известен в русской научной литературе по работам самого Галеркина и его коллег. Этот метод впервые привлек внимание западных ученых благодаря работам Дункана, посвященных динамике авиационных конструкций. Бикли [9] применил метод Галеркина к решению задачи о нестационарной теплопередаче путем рассмотрения эквивалентного электрического контура, получив решения с помощью метода Галеркина. Бикли сравнил их с решениями, построенными методом коллокаций и методом наименьших квадратов.

В результате сочетания метода Галеркина и представлений Фурье были развиты спектральные методы Галеркина, которые при специальном способе учета нелинейных членов позволили обратиться к прямому моделированию турбулентности [9] и к глобальным методам прогноза погоды (Бурке 10)]. В этих задачах особенно то, что сложность физической формулировки сочетается с простотой формы границ.

Существует тесная взаимосвязь между методом Галеркина и таким вариационным методом, как метод Рэлея – Ритца. Эта взаимосвязь приобрела особое значение в связи с разработкой метода конечных элементов.

Первоначально метод конечных элементов рассматривался как специальная инженерная процедура для построения матричных решений задач о расчете напряжений. Вскоре стало очевидно, что этой процедуре можно дать вариационную интерпретацию, если ввести в рассмотрение потенциальную энергию системы. Установление связи с методом Галеркина позволило распространить процедуры с конечными элементами на те области, где ни один вариационный принцип не имеет очевидной силы; это касается многих проблем газодинамики и теории теплопередачи [11]. Можно утверждать, что метод Галеркина с конеч-

ными элементами является наиболее распространенным среди разнообразных методов конечных элементов.

Метод Галёркина имеет несколько основных усовершенствованных вариантов:

Метод Галеркина – Петрова – разложение решения производится по одному базису, а ортогональность невязки необходима по другому.

Метод Галёркина – Канторовича – позволяет свести уравнения в частных производных к обыкновенным дифференциальным уравнениям. Например, в двумерной задаче решение представляется в виде:

$$\psi(x, y) = \sum_n b_n a_n(x) \varphi_n(y),$$

тогда процедура Галёркина проводится применительно лишь к одним функциям ( $a_n(x)$  либо  $\varphi_n(y)$ ). В итоге получается система обыкновенных дифференциальных уравнений, для решения которых существуют эффективные численные методы.

Некоторые особенности метода Галеркина станут очевидными после рассмотрения простых примеров.

## 2.2 Описание метода

### 2.2.1 Основа метода

Первым шагом в реализации метода Галёркина является выбор набора базисных функций, которые удовлетворяют граничным условиям, а также образуют полную систему в пределе бесконечного количества элементов базиса.

Конкретный вид функций определяется из специфики задачи и удобства работы. Часто в качестве базисных функций выступают тригонометрические функции, ортогональные полиномы (полиномы Лежандра, Чебышёва, Эрмита).

Решение представляется в виде разложения по базису:

$$\psi(x) = \sum_{k=1}^n \alpha_k \varphi_k(x).$$

Затем приближенное решение подставляется в исходное дифференциальное уравнение, и вычисляется его невязка. Для однородного уравнения:

$$L \left[ \sum_{k=1}^n \alpha_k \varphi_k(x) \right] = N(x).$$

Для неоднородного уравнения  $L(u) = f(x)$  невязка будет иметь вид  $N(x) = L(u) - f(x)$ .

Далее выдвигается требование ортогональности невязки к базисным функциям, то есть:

$$\int_a^b N(x) \varphi_k(x) dx = 0.$$

Откуда получается однородная система уравнений для коэффициентов в разложении, и удается приближенно найти собственные значения задачи.



## 2.2.2 Решение первой начально-краевой задачи для уравнения теплопроводности с одной пространственной переменной

На отрезке  $0 < x < 1$  при  $t > 0$  рассмотрим следующую начально-краевую задачу

$$\begin{aligned} \frac{\partial u}{\partial t} &= \kappa \frac{\partial^2 u}{\partial x^2} + \tilde{f}(x, t), \quad u(x, 0) = \tilde{u}_0(x), \\ u(0, t) &= u^0(t), \quad u(1, t) = u^1(t). \end{aligned} \quad (2.1)$$

Представляя решение  $u(x, t)$  этой задачи в виде

$$u(x, t) = v(x, t) + u^0(t) + x(u^1(t) - u^0(t)),$$

Сведем задачу (2.1) к аналогичной задаче, но уже с однородными (нулевыми) условиями на границе области (в точках  $x = 0$  и  $x = 1$ ):

$$\frac{\partial v}{\partial t} = \kappa \frac{\partial^2 v}{\partial x^2} + f(x, t), \quad v(0, t) = 0, \quad v(1, t) = 0, \quad (2.2)$$

$$v(x, 0) = u_0(x). \quad (2.3)$$

Идея метода Галёркина заключается в преобразовании дифференциального уравнения (2.2) в интегральное тождество и в специальном выборе приближенных решений.

Вместо уравнения (2.2) мы рассмотрим интегральное тождество

$$\int_0^1 \left( \frac{\partial v}{\partial t} \cdot \varphi(x) + \kappa \frac{\partial v}{\partial x} \cdot \frac{\partial \varphi}{\partial x} \right) dx = \int_0^1 f(x, t) \cdot \varphi(x) dx, \quad (2.4)$$

которое получим после умножения уравнения теплопроводности на произвольную функцию  $\varphi(x)$ , равную нулю при  $x = 0$  и  $x = 1$ , и интегрирования по частям в слагаемом, содержащем производные по пространственной переменной (перебросили дифференцирование с функции  $v(x, t)$  на пробную функцию  $\varphi(x)$ ).

Для гладких функций  $v(x,t)$  тождество (2.4) эквивалентно уравнению (2.1). Чтобы показать это, достаточно провести в (2.4) обратную операцию интегрирования по частям (перебросить дифференцирование с пробной функции  $\varphi(x)$  на функцию  $v(x,t)$ ) и воспользоваться тем, что тождество

$$\int_0^1 F(x) \cdot \varphi(x) dx = 0 \quad \forall \varphi$$

влечет

$$F(x) \equiv 0.$$

Далее рассмотрим подпространство  $\overset{o}{W}_2^1(0,1)$  пространства  $W_2^1(0,1)$ , состоящее из всех функций, равных нулю при  $x=0$  и  $x=1$ . В качестве базиса в этом пространстве выберем функции  $\varphi_k(x) = \sqrt{2} \sin k\pi x$ ,  $k=1, 2, \dots$

Выбор решения интегрального тождества (2.4) из пространства  $\overset{o}{W}_2^1(0,1)$  автоматически удовлетворяет краевым условиям в (2.2). Далее ослабим требования на пробные функции – будем считать их элементами пространства  $\overset{o}{W}_2^1(0,1)$ .

Согласно методу Галёркина ищутся приближенные решения  $v^N(x,t)$  интегрального тождества (2.4) в виде

$$v^N(x,t) = \sum_{k=1}^N c_k(t) \varphi_k(x), \quad (2.5)$$

удовлетворяющих интегральному тождеству (2.4) для первых  $N$  функций

$\varphi \in \overset{o}{W}_2^1(0,1)$  базиса

$$\int_0^1 \left( \frac{\partial v^N}{\partial t} \cdot \varphi_k(x) + \kappa \frac{\partial v^N}{\partial x} \cdot \frac{\partial \varphi_k}{\partial x} \right) dx = \int_0^1 f(x,t) \cdot \varphi_k(x) dx, \quad k=1, 2, \dots, N \quad (2.4)$$

и начальным условиям (2.3) приближенно:

$$v^N(x,0) = \sum_{k=1}^N u_0^k \varphi_k(x). \quad (2.7)$$

В (2.7)

$$u_0^k = \int_0^1 u_0(x) \varphi_k(x) dx. \quad (2.8)$$

Пользуясь ортогональностью базиса задачу (2.6) – (2.7) можно переписать в виде

$$\frac{dc_k}{dt} = - \sum_{m=1}^N \left( \int_0^1 \kappa \frac{d\varphi_k}{dx} \cdot \frac{d\varphi_m}{dx} dx \right) c_m + \int_0^1 f(x,t) \cdot \varphi_k(x) dx, \quad k = 1, 2, \dots, N \quad (2.9)$$

$$c_k(0) = u_0^k. \quad (2.10)$$

В силу теоремы Коши, задача Коши (2.10) для линейной системы (2.9) имеет единственное решение при всех  $t > 0$ . Таким образом, определено приближенное решение  $v^N(x,t)$  задачи (2.3), (2.4).

Наличие априорной оценки

$$\begin{aligned} \max_{0 < t < T} \int_0^1 \left| \frac{\partial v^N}{\partial t}(x,t) \right|^2 dx + \int_0^1 \int_0^1 \kappa \left| \frac{\partial v^N}{\partial x}(x,t) \right|^2 dx dt \leq \\ e^T \int_0^1 |u_0(x)|^2 dx + e^T \int_0^1 \int_0^1 f^2(x,t) dx dt, \end{aligned} \quad (2.11)$$

которая следует из тождеств (2.6), если  $k$ -тое тождество умножить на  $dc_k/dt$  и все тождества сложить. Это позволит выделить подпоследовательность  $\{v^{N_k}\}$ , сходящуюся слабо в  $L_2(0,1)$  вместе со своими первыми производными при всех  $t$ ,  $0 < t < T$  к функции  $v(x,t)$ . Осуществляя слабый предельный переход в тождестве (2.6) убеждаемся, что  $v(x,t)$  является искомым решением.

## 2.3 Простейшие приемы применения метода Галеркина.

Рассмотрим технику применения метода Галеркина к решению некоторых характерных задач. В первом примере метод Галеркина используется для того, чтобы привести обыкновенное дифференциальное уравнение к системе алгебраических уравнений. Во втором примере иллюстрируется применение метода Галеркина с целью приведения параболических дифференциальных уравнений в частных производных к системе обыкновенных дифференциальных уравнений, зависящих от времени.

### 1.2.1 Обыкновенное дифференциальное уравнение

Рассмотрим обыкновенное дифференциальное уравнение

$$\frac{dy}{dx} - y = 0 \quad (1.1)$$

с граничным условием  $y = 1$  при  $x = 0$ . Приближенное решение ищется в области  $0 \leq x \leq 1$ . Точное решение имеет форму  $y = e^x$ . Приближенное (пробное) решение представляется в виде

$$y_a = 1 + \sum_{j=1}^N a_j x^j, \quad (1.2)$$

где главный член вводится для удовлетворения граничному условию. После этого пробные функции  $x^j$  позволяют удовлетворить однородным граничным условиям. (Обычно практикуемым приемом при реализации традиционного метода Галеркина является представление пробного решения в форме, обеспечивающей удовлетворение граничным условиям). Это позволяет получить максимально точное решение при заданном числе неизвестных  $N$ . Формулу (1.2) можно также переписать в виде

$$y_a = \sum_{j=0}^N a_j x^j, \quad (1.3)$$

где коэффициент  $a_0$  выбран так, чтобы удовлетворить граничному условию, то есть  $a_0 = 1$ . Подстановка формулы (1.2) в уравнение (1.1) дает выражение невязки

$$R = -1 + \sum_{j=1}^N a_j (jx^{j-1} - x^j). \quad (1.4)$$

Составляя внутреннее произведение, определяемое согласно (0.5), получим

$$(R, x^{k-1}) = 0. \quad (1.5)$$

Если записать это для каждого  $k = 1, \dots, N$ , то получится система уравнений, записываемая в виде

$$MA = D, \quad (1.6)$$

где  $A$  – вектор, составленный из неизвестных коэффициентов  $a_j$ . Элемент матрицы  $D$  дается выражением  $d_k = (1, x^{k-1}) = 1/k$ .

Элемент матрицы  $M$  соответствует формуле

$$m_{kj} = (jx^{j-1} - x^j, x^{k-1}) = \frac{j}{j+k-1} - \frac{1}{j+k}.$$

Очевидно, точность решения быстро возрастает вместе с увеличением числа  $N$ . Здесь видна очень важная особенность традиционного метода Галеркина, состоящая в том, что высокая степень точности может быть достигнута за счет алгебраических усилий. Числовая характеристика этой особенности может быть дана путем вычисления дискретной  $L_2$  - погрешности, соответствующей разности между приближенным и точным решениями:  $\|y - y_a\|_{2,d}$ .

Погрешность в норме  $L_2$  ( $L_2$  - погрешность) определяется формулой

$$\|y - y_a\|_2 = \left[ \int_0^1 (y - y_a)^2 dx \right]^{1/2}. \quad (1.8)$$

Дискретная  $L_2$  - погрешность определяется как

$$\|y - y_a\|_{2,d} = \left[ \sum_{l=1}^L (y - y_a)_l^2 \right]^{1/2}. \quad (1.9)$$

### 1.2.2. Течение вязкой жидкости в канале

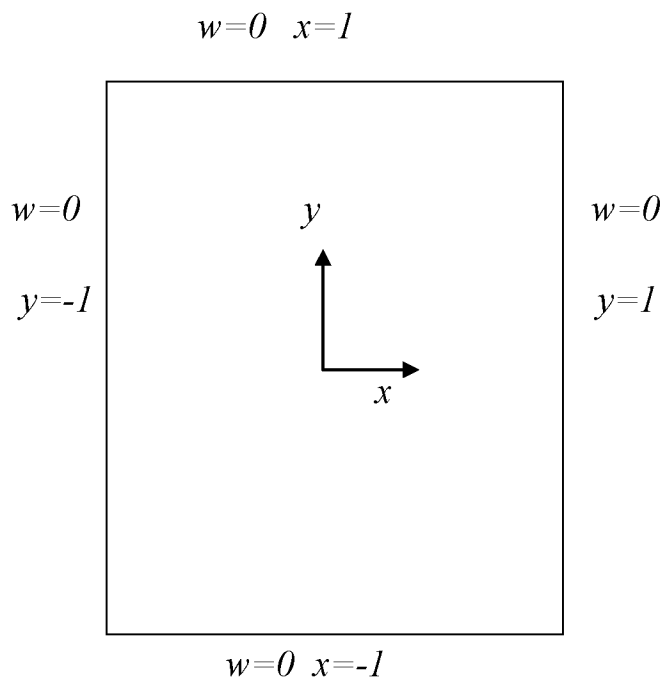


Рис. 1.

Рассмотрим канал, имеющий квадратное поперечное сечение (рис.1). Для описания установившегося течения вязкой жидкости в таком канале служит проекция уравнения импульсов на ось  $z$ , то есть

$$u \frac{\partial w}{\partial x} + v \frac{\partial w}{\partial y} + w \frac{\partial w}{\partial z} + \frac{1}{\rho} \frac{\partial p}{\partial z} = \nu \left( \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + \frac{\partial^2 w}{\partial z^2} \right). \quad (1.10)$$

Вдали от выходного или входного сечения течение остается неизменным в направлении оси  $z$ ; тогда уравнение (1.10) принимает вид

$$\frac{1}{\mu} \frac{\partial p}{\partial z} = \frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2}. \quad (1.11)$$

При течении такого рода величина  $\partial p / \partial z$  постоянна. Приведение к безразмерной форме записи за счет надлежащего выбора масштабов преобразует (1.11) к виду

$$\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + 1 = 0, \quad (1.12)$$

то есть к виду уравнения Пуассона для функции  $w$ . Граничные условия имеют вид

$$w = 0 \text{ при } x = \pm 1 \text{ и } y = \pm 1. \quad (1.13)$$

Если пробное решение основано на тригонометрических функциях, то возможно для каждой пробной функции удовлетворение граничным условиям.

Таким образом, положим

$$w_a = \sum_{i=1,3,5}^N \sum_{j=1,3,5}^N a_{ij} \cos i \frac{\pi}{2} x \cos j \frac{\pi}{2} y. \quad (1.14)$$

Подставляя формулу (1.14) в уравнение (1.12), получим невязку

$$R = - \left[ \sum_{i=1,3,5}^N \sum_{j=1,3,5}^N a_{ij} \cos i \frac{\pi}{2} x \cos j \frac{\pi}{2} y \times \left\{ \left( i \frac{\pi}{2} \right)^2 + \left( j \frac{\pi}{2} \right)^2 - 1 \right\} \right]. \quad (1.15)$$

Метод Галеркина позволяет получить алгебраические уравнения для коэффициентов  $a_{ij}$  путем составления внутренних произведений

$$\left( R, \cos i \frac{\pi}{2} x \cos j \frac{\pi}{2} y \right) = 0, \quad i = 1, 3, 5, \dots; \quad j = 1, 3, 5, \dots \quad (1.16)$$

Обращаясь к примеру, приведенному в 1.2.1, видим, что из уравнений типа (1.16) получается система уравнений, которую и следует разрешить относительно неизвестных коэффициентов. Применительно к данному примеру специальный выбор пробных функций позволяет определить все  $a_{ij}$  непосредственно. А именно

$$a_{ij} = + \left( \frac{8}{\pi^2} \right)^2 \frac{(-1)^{(i+j)/2-1}}{ij(i^2 + j^2)}. \quad (1.17)$$

Избавление от необходимости решения системы уравнений стало результатом выбора таких пробных функций, которые являются ортогональными в рассматриваемой области. Такой выбор дает важные преимущества. Применение метода Галеркина с тригонометрическими пробными функциями приводит к таким же уравнениям и решениям, как и применение метода Фурье с конечным числом членов разложения ([5]).

Подстановка выражений (1.17) в формулу (1.14) дает искомое решение

$$w_a = \left( \frac{8}{\pi^2} \right)^2 \sum_{i=1,3,5}^N \sum_{j=1,3,5}^N \frac{(-1)^{(i+j)/2-1}}{ij(i^2 + j^2)} \cos i \frac{\pi}{2} x \cos j \frac{\pi}{2} y. \quad (1.18)$$

Точное решение для безразмерного расхода  $q$  было дано в книге [6]. Расход  $q$  связан со скоростью  $w_a$  соотношением

$$q = \int_{-1}^1 \int_{-1}^1 w_a(x, y) dx dy. \quad (1.19)$$

Вычисление правой части дает

$$q = \left( \frac{8}{\pi^2} \right)^3 \sum_{i=1,3,5}^N \sum_{j=1,3,5}^N \frac{1}{i^2 j^2 (i^2 + j^2)}. \quad (1.20)$$

Решения могли бы быть получены также и с другими подходящими пробными функциями. Например, Финлейсон [7] использовал представление



$$w_a = \sum_{j=1}^N a_j (1-x^2)^j (1-y^2)^j. \quad (1.21)$$

Использование пробного решения в таком виде, как (1.21) потребовало бы разрешения системы уравнений типа (1.6) относительно неизвестных коэффициентов  $a_j$ . Увеличение числа  $N$  требует полного обновления набора неизвестных коэффициентов  $a_j$ . В противоположность этому применение ортогональных функций позволяет проводить расчет коэффициентов  $a_{ij}$  независимо от предшествующих коэффициентов.

### 3. КОРРЕКТНОСТЬ ПЕРИОДИЧЕСКОЙ НАЧАЛЬНО-КРАЕВОЙ ЗАДАЧИ ДЛЯ УРАВНЕНИЯ СТОКСА

#### 3.1. Постановка задачи

Рассмотрим единичный куб  $Y = (0,1) \times (0,1) \times (0,1) \subset R^3$ , моделирующий твердое тело. Пусть область  $Y_f$ , заполненная вязкой жидкостью такова, что  $Y_f \subset Y$ . Граница  $\gamma = \partial Y_f \cap Y$  есть липшицева поверхность (см. рис. 2).

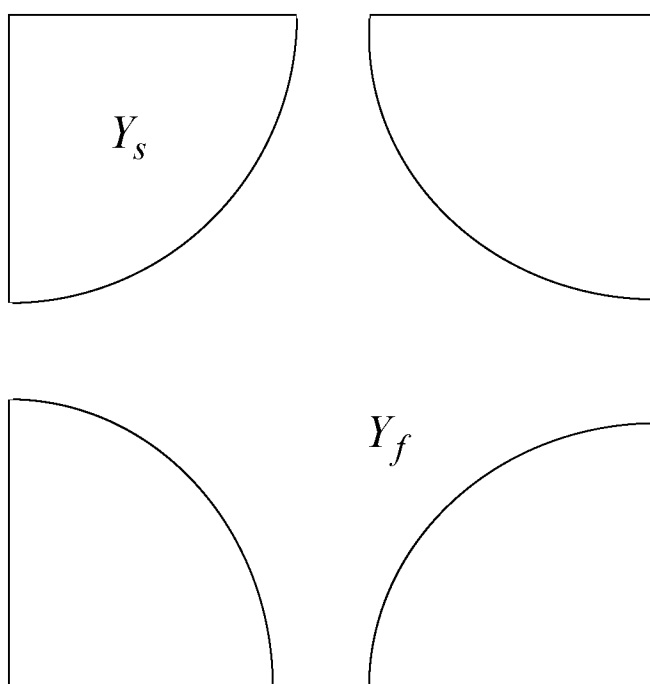


Рис. 2

Рассмотрим следующую периодическую начально-краевую задачу для скорости жидкости  $\partial W(\mathbf{y}, t) / \partial t$  в области  $Y_f$ , состоящую из системы дифференциальных уравнений Стокса:

$$\frac{\partial^2 W}{\partial t^2} = \nabla_{\mathbf{y}} \cdot \left( \mu D \left( \mathbf{y}, \frac{\partial W}{\partial t} \right) - QI \right), \quad \nabla_{\mathbf{y}} \cdot W = 0, \quad \mathbf{y} \in Y_f \times (0, T) \quad (3.1)$$

начально-краевых условий

$$W(y,0) = W_0(y), \quad y \in Y_f, \quad (3.2)$$

$$W(y,t) = 0, \quad y \in \gamma \times (0,T), \quad (3.3)$$

и условий 1-периодичности по пространственной переменной  $y$  на границе  $\partial Y_f \cap \partial Y$  для функции  $W$ .

Здесь  $D\left(y, \frac{\partial W}{\partial t}\right) = \frac{1}{2}\left(\nabla_y\left(\frac{\partial W}{\partial t}\right) + \nabla_y^*\left(\frac{\partial W}{\partial t}\right)\right)$  – симметрическая часть градиента,  $I$  – единичная матрица.

Обозначим через  $L^2((0,T); \overset{\circ}{W}_{2,\Pi}(Y_f))$  пространство функций  $\varphi(y,t) \in L^2((0,T); \overset{\circ}{W}_{2,\Pi}(Y_f))$ , 1-периодичных по переменной  $y$ .

**Определение 1.** Обобщенным решением задачи (1.3) – (3.3) называется вектор-функция  $W(y,t) \in L^2((0,T); \overset{\circ}{W}_{2,\Pi}(Y_f))$ , удовлетворяющая интегральному тождеству

$$-\int_0^T \int_{Y_f} \frac{\partial W}{\partial t} \frac{\partial \eta}{\partial t} dy dt + \mu \int_0^T \int_{Y_f} D\left(y, \frac{\partial W}{\partial t}\right) : D\eta dy dt = \int_0^T \frac{\partial W_0}{\partial t}(y) \eta(y,0) dy \quad (3.4)$$

для любой вектор-функции  $\eta(y,t) \in \overset{\circ}{W}_{2,\Pi}(Y_T)$ , соленоидальной ( $\operatorname{div} \eta = 0$ ) в области  $Y_f$  и равной нулю при  $t = T$ .

Здесь  $A : B = \operatorname{tr}(AB^T)$ ,  $A$  и  $B$  – квадратные матрицы,  $Y_T = Y \times (0,T)$  – цилиндр.

Для доказательства существования обобщенного решения задачи (3.1) – (3.3) находятся решения вспомогательных задач в более широкой области, слабо сходящиеся к решению исходной задачи.

Для этого введем положительный параметр  $\lambda$ . Задачу (3.1) – (3.3) будем рассматривать как предельную при  $\lambda \rightarrow \infty$  для следующей вспомогательной задачи с параметром  $\lambda$ , определенной в области  $Y = (0,1) \times (0,1) \times (0,1)$

$$\frac{\partial^2 W^\lambda}{\partial t^2} = \nabla_y \cdot \left( \mu D \left( y, \frac{\partial W^\lambda}{\partial t} \right) - QI \right) + \lambda(1 - \chi(y)) D \Phi, W^\lambda, \\ Q + \lambda \nabla_y \cdot W^\lambda = 0, \quad (3.5)$$

$$W^\lambda(y, 0) = \chi W_0(y) \quad (3.6)$$

с условиями 1-периодичности по переменной  $y$  на границе  $\partial Y_f \cap \partial Y$  для функции  $W^\lambda$ .

Здесь  $\chi(y)$  – характеристическая функция  $Y_f$  в  $Y$ , то есть

$$\chi(y) = \begin{cases} 0, & y \in Y_s, \\ 1, & y \in Y_f. \end{cases}$$

**Определение 2.** Обобщенным решением задачи (3.5)- (3.6) называется вектор-функция  $W^\lambda(y, t) \in L^2((0, T); W_{2, \Pi}^1(Y))$ , удовлетворяющая интегральному тождеству

$$- \int_{Y_T} \frac{\partial W^\lambda}{\partial t} \frac{\partial \eta}{\partial t} dy dt + \mu \int_{Y_T} D \left( y, \frac{\partial W^\lambda}{\partial t} \right) : D \Phi, \eta dy dt + \lambda \int_{Y_T} \operatorname{div} \frac{\partial W^\lambda}{\partial t} \operatorname{div} \eta dy dt + \\ + \lambda \int_{Y_T} (1 - \chi(y)) W^\lambda \eta dy dt = \int_{Y_T} \chi W_0(y) \eta(y, 0) dy \quad (3.7)$$

для любой вектор-функции  $\eta(y, t) \in \overset{o}{W}_{2, \Pi}^{1,1}(Y_T)$  и равной нулю при  $t = T$ .

**Теорема 1.** Задача (3.5) – (3.6) может иметь не более одного обобщенного решения из  $L^2((0, T); W_{2, \Pi}^1(Y))$ .

**Теорема 2.** *Обобщенное решение задачи (3.5) – (3.6) существует, единственно и для него справедлива оценка:*

$$\begin{aligned} & \max_{0 < t < T} \int_Y \left| \frac{\partial W^\lambda}{\partial t} \right|^2 dy + \mu \int_{Y_T} \left| D \left( y, \frac{\partial W^\lambda}{\partial t} \right) \right|^2 dy dt + \lambda \int_{Y_T} \left( \operatorname{div} \frac{\partial W^\lambda}{\partial t} \right)^2 dy dt + \\ & + \lambda \int_{Y_T} (1 - \chi) |W^\lambda|^2 dy dt \leq C_1. \end{aligned} \quad (1.10)$$

где  $C_1$  – постоянная, не зависящая от параметра  $\lambda$ .

**Теорема 3.** *При всех положительных значениях  $\lambda$  существует единственное обобщенное решение задачи (3.1) – (3.3) и для него справедлива оценка:*

$$\max_{0 < t < T} \int_Y \left| \frac{\partial W}{\partial t} \right|^2 dy + \mu \int_{Y_T} \left| D \left( y, \frac{\partial W}{\partial t} \right) \right|^2 dy dt \leq C_0. \quad (1.11)$$

где  $C_0$  – некоторая постоянная, не зависящая от геометрии области.

### 3.2 Доказательство теоремы 1

Предположим, что задача (3.5) – (3.6) имеет два обобщенных решения, тогда их разность  $w$  будет удовлетворять интегральному тождеству

$$\begin{aligned}
 & - \int_{0Y_f}^T \int w \frac{\partial \eta}{\partial t} dy dt + \mu \int_{0Y_f}^T \int D \Psi, w \rceil D \Psi, \eta \rceil dy dt + \lambda \int_{0Y_f}^T \int \operatorname{div} w \operatorname{div} \eta dy dt + \\
 & + \lambda \int_{0Y_f}^T \int (1 - \chi(y)) w \eta dy dt = 0
 \end{aligned} \tag{3.8}$$

для любой вектор-функции  $\eta(y, t) \in \overset{o}{W}_{2,1}(Y_T)$  и равной нулю при  $t = T$ .

Таким образом, функция  $w$  есть обобщенное решение задачи (3.5) – (3.6) с однородным начальным условием  $W(y, 0) = 0$ .

Выберем в качестве  $\eta(y, t)$  функцию  $w(y, t)$ . Получим интегральное тождество

$$\begin{aligned}
 & \frac{1}{2} \int_Y w^2(y, 0) dy dt + \mu \int_{Y_f} \nabla w : \nabla w dy dt + \lambda \int_{Y_f} |\operatorname{div} w|^2 dy dt + \\
 & + \lambda \int_{Y_f} (1 - \chi(y)) |w|^2 dy dt = 0,
 \end{aligned} \tag{3.9}$$

из которого следует, что  $\|w\| = 0$ , то есть  $w = 0$ .

### 3.3 Доказательство теоремы 2

Для обоснования существования обобщенного решения задачи (3.5) – (3.6) будем использовать метод Галёркина.

Выберем в пространстве  $W_{2,\Pi}^1(Y)$  фундаментальную ортонормированную в  $L_2(Y)$  систему линейно независимых вектор-функций  $\{\varphi_k(y)\}$  (будем называть их базисными функциями), таких что

$$(\varphi_i, \varphi_j) = \int_Y \varphi_i(y) \varphi_j(y) dy = \delta_{i,j}$$

Пусть конечномерное подпространство  $H_N$  представляет собой линейную оболочку выбранных базисных функций.

Приближенное решение  $V_N(y,t) = \partial W_N(y,t)/\partial t$  будем искать в виде:

$$V_N(y,t) = \sum_{m=1}^N c_m(t) \varphi_m(y). \quad (3.12)$$

(Индекс  $\lambda$  будем опускать для удобства обозначений). Функция  $V_N(y,t)$  должна удовлетворять начальному условию

$$V_N(y,0) = V_\theta^N(y). \quad (3.13)$$

Функция  $V_\theta^N(y)$  является ортогональной проекцией вектора  $\chi V_\theta(y)$  на подпространство  $H_N$ . Следовательно,  $\chi V_\theta(y) = V_\theta^N(y) + z$ , где  $z$  есть ортогональная составляющая вектора  $V_\theta^N(y)$  относительно  $H_N$ . Поскольку  $V_\theta^N(y) \in H_N$ , то

$$V_\theta^N(y) = \sum_{m=1}^N b_m \varphi_m(y),$$

где

$$\begin{aligned}
b_m(t) &= \left( \chi_0^N(y), \varphi_m(y) \right)_{L_2(Y)} = \left( \chi V_0(y) - z, \varphi_m(y) \right)_{L_2(Y)} \\
&= \left( \chi V_0(y), \varphi_m(y) \right)_{L_2(Y)} - \left( z, \varphi_m(y) \right)_{L_2(Y)}
\end{aligned}$$

Учитывая, что  $z \perp \varphi_m(y)$ , окончательно получим

$$b_m(t) = \left( \chi V_0(y), \varphi_m(y) \right)_{L_2(Y)} = \int_Y \chi V_0(y) \varphi_m(y) dy. \quad (3.14)$$

Коэффициенты разложения  $c_m(t)$  в формуле (3.12) определяются из условия ортогональности в  $L_2(Y)$  невязки

$$\mathcal{L}(V_N) = \frac{\partial V}{\partial t} - \mu \Delta V_N + \nabla Q + \lambda V_N (1 - \chi(y))$$

подпространству  $H_N$ , то есть из условия

$$\left( \mathcal{L}(V_N), \varphi_m(y) \right)_{L_2(Y)} = 0,$$

$\forall \varphi_m(y) \in H_N, m = 1, 2, \dots, N$ . Таким образом, что  $Q + \lambda \operatorname{div} V = 0$ , для определения коэффициентов  $c_m(t)$  имеем систему уравнений:

$$\int_Y \left( \frac{\partial V_N}{\partial t} \varphi_m + \mu \nabla V_N : \nabla \varphi_m + \lambda \operatorname{div} V_N \operatorname{div} \varphi_m + \lambda (1 - \chi(y)) V_N \varphi_m \right) dy = 0. \quad (1.15)$$

Подставляя в эти уравнения выражения для  $V_N$ , получим систему  $N$  обыкновенных линейных дифференциальных уравнений первого порядка с постоянными коэффициентами:

$$\begin{aligned}
\dot{c}_m(t) + \sum_{k=1}^N c_k(t) \int_Y \left( \mu \nabla \varphi_m : \nabla \varphi_k + \lambda \operatorname{div} \varphi_m \operatorname{div} \varphi_k + \lambda (1 - \chi(y)) \varphi_m \varphi_k \right) dy &= 0, \\
c_k(0) = b_m, \quad m = 1, 2, \dots, N.
\end{aligned} \quad (3.16)$$

Полученная система при заданных начальных условиях имеет единственное решение. Следовательно, функция  $V_N(y, t) = \sum_{m=1}^N c_m(t) \varphi_m(y) \quad \forall N = 1, 2, \dots$  определяется однозначно.



Докажем теперь ограниченность последовательности галеркинских приближений  $\mathcal{V}_N(y, t)$  в норме пространства  $L^2((0, T); W_{2, H}^1(Y))$

$$\|\mathcal{V}(y, t)\|^2 = \int_{Y_T} \left( |\mathcal{V}|^2 + \sum_{i=1}^3 \left| \frac{\partial \mathcal{V}}{\partial y_i} \right|^2 \right) dy dt. \quad (3.17)$$

Умножим уравнение с номером  $m$  из системы (3.16) на  $c_m(t)$  и просуммируем по  $m$  от 1 до  $N$ . Получим

$$\begin{aligned} \sum_{m=1}^N \dot{c}_m c_m + \mu \int_Y \nabla \mathcal{V}_N : \nabla \mathcal{V}_N dy + \lambda \int_Y |\operatorname{div} \mathcal{V}_N|^2 dy + \\ + \lambda \int_Y (1 - \chi(y)) |\mathcal{V}_N|^2 dy = 0, \end{aligned} \quad (3.18)$$

Представим первое слагаемое в следующем виде

$$\sum_{m=1}^N \dot{c}_m c_m = \frac{1}{2} \frac{d}{dt} \int_Y |\mathcal{V}_N|^2 dy$$

и проинтегрируем уравнение (3.18) от 0 до  $t$ :

$$\begin{aligned} \frac{1}{2} \int_Y |\mathcal{V}_N(y, t)|^2 dy + \mu \int_{Y_t} \nabla \mathcal{V}_N : \nabla \mathcal{V}_N dy dt + \lambda \int_{Y_t} |\operatorname{div} \mathcal{V}_N|^2 dy dt + \\ + \lambda \int_{Y_t} (1 - \chi(y)) |\mathcal{V}_N|^2 dy dt = \frac{1}{2} \int_Y |\mathcal{V}_\theta^N|^2 dy. \end{aligned} \quad (3.19)$$

Поскольку вектор  $\mathcal{V}_\theta^N$  является ортогональной проекцией  $\chi \mathcal{V}_\theta$  на подпространство  $H_N$ , то  $|\mathcal{V}_\theta^N| \leq |\chi \mathcal{V}_\theta^N|$ . Следовательно,

$$\frac{1}{2} \int_Y |\mathcal{V}_\theta^N|^2 dy \leq C_1,$$

$C_1 = \text{const}$ . Окончательно получаем

$$\begin{aligned}
& \max_{0 < t < T} \int_Y |V_N|^2 dy + \mu \int_{Y_T} \sum_{i=1}^3 \left| \frac{\partial V_N}{\partial y_i} \right|^2 dy dt + \lambda \int_{Y_T} \text{Div} V_N^2 dy dt + \\
& + \lambda \int_{Y_T} (1 - \chi(y)) |V_N|^2 dy dt \leq C_1,
\end{aligned} \tag{3.20}$$

а, следовательно

$$\max_{0 < t < T} \int_Y |V_N|^2 dy \leq C_1,$$

$$\int_{Y_T} \sum_{i=1}^3 \left| \frac{\partial V_N}{\partial y_i} \right|^2 dy dt \leq \frac{C_1}{\mu},$$

$$\int_{Y_T} \text{Div} V_N^2 dy dt \leq \frac{C_1}{\lambda},$$

$$\int_{Y_T} (1 - \chi(y)) |V_N|^2 dy dt \leq \frac{C_1}{\lambda}. \tag{3.21}$$

Так как  $\mu > 0$ ,  $\lambda > 0$ , имеем ограниченность последовательности  $\{V_N\}$  в норме (1.17).

Из известной теоремы функционального анализа (см. раздел 1, теорема 2) следует слабая компактность множества функций  $\{V_N\}$ . Таким образом, из последовательности  $\{V_N\}$  можно выделить подпоследовательность, слабо сходящуюся к некоторой функции  $V(y, t)$ . Докажем теперь, что эта функция является искомым обобщенным решением задачи, то есть удовлетворяет интегральному тождеству (1.7) для любой вектор-функции  $\eta(y, t) \in \overset{o}{W}_{2, \Pi}^{1,1}(Y_T)$ , равной нулю при  $t = T$ . Для этого достаточно установить выполнение этого условия для некоторого счетного всюду плотного множества в  $\overset{o}{W}_{2, \Pi}^{1,1}(Y_T)$ .

Докажем, что интегральное тождество (1.7) справедливо для любой функции  $\eta_N$  вида

$$\eta_N(\mathbf{y}, t) = \sum_{m=1}^N d_m(t) \varphi_m(\mathbf{y}). \quad (3.22)$$

Здесь  $d_m(t)$  – произвольные функции из  $L_2(0, T)$ .

Зафиксируем  $N' \leq N$ . Умножим каждое уравнение системы (3.15) на свою гладкую функцию  $d_m(t)$ , просуммируем по  $m$  от 1 до  $N'$  и проинтегрируем по  $t$  от 0 до  $T$ . Получим

$$\begin{aligned} & \int_0^T dt \left( \int_Y \frac{dV_N}{dt} \eta_{N'} dy + \mu \int_Y \nabla V_N : \nabla \eta_{N'} dy + \lambda \int_Y \operatorname{div} V_N \operatorname{div} \eta_{N'} dy + \right. \\ & \left. + \lambda \int_Y ((1 - \chi(\mathbf{y})) V_N \eta_{N'} dy) \right) = 0. \end{aligned} \quad (3.23)$$

После интегрирования по частям в первом слагаемом, получим

$$\begin{aligned} & - \int_{Y_T} V_N(\mathbf{y}, t) \frac{d\eta_{N'}}{dt} dy dt + \mu \int_{Y_T} \nabla V_N : \nabla \eta_{N'} dy dt + \lambda \int_{Y_T} \operatorname{div} V_N \operatorname{div} \eta_{N'} dy dt + \\ & + \lambda \int_{Y_T} ((1 - \chi(\mathbf{y})) V_N \eta_{N'} dy dt = \int_Y V_0(\mathbf{y}) \eta_{N'}(\mathbf{y}, 0) dy. \end{aligned} \quad (3.24)$$

Зафиксируем функцию  $\eta_{N'}(\mathbf{y}, t)$  и перейдем к пределу при  $N \rightarrow \infty$  по выбранной подпоследовательности. Получим интегральное тождество для функции  $V(\mathbf{y}, t)$

$$\begin{aligned} & - \int_{Y_T} V(\mathbf{y}, t) \frac{d\eta_{N'}}{dt} dy dt + \mu \int_{Y_T} \nabla V : \nabla \eta_{N'} dy dt + \lambda \int_{Y_T} \operatorname{div} V \operatorname{div} \eta_{N'} dy dt + \\ & + \lambda \int_{Y_T} ((1 - \chi(\mathbf{y})) V \eta_{N'} dy dt = \int_Y V_0(\mathbf{y}) \eta_{N'}(\mathbf{y}, 0) dy \end{aligned} \quad (3.25)$$

верное для любой функции  $\eta_{N'}(\mathbf{y}, t)$  вида (3.22). Такие функции всюду плотны в

пространстве  $W_{2, \Pi}^{0, 1, 1}(Y_T)$ , поэтому интегральное тождество выполняется и для

любой функции  $\eta(\mathbf{y}, t) \in W_{2, \Pi}^{0, 1, 1}(Y_T)$ .

Таким образом,  $\lim_{N \rightarrow \infty} V_N = V(\mathbf{y}, t)$  и есть искомое обобщенное решение задачи.

### 3.4 Доказательство теоремы 3

**Лемма 1.** *Решение  $V^\lambda$  вспомогательной задачи (3.5) – (3.6) слабо в  $L_2(Y_T)$  сходится к решению  $V$  задачи (3.1) – (3.3).*

*Доказательство.* Для каждого фиксированного  $\lambda$  мы нашли решение  $V^\lambda$  вспомогательной задачи (3.5) – (3.6). Очевидно, что функция  $V^\lambda$  как предел последовательности  $V_N^\lambda$  также удовлетворяет неравенствам (3.21). При  $\lambda \rightarrow \infty$  эти неравенства только усиливаются, поэтому последовательность  $V^\lambda$  является ограниченной в норме (3.17), то есть из нее снова можно выделить сходящуюся подпоследовательность.

Выбирая подпоследовательность, для которой будем сохранять прежнее обозначение, получим, что решение  $V^\lambda$  вспомогательной задачи сходится слабо в  $L_2(Y_T)$  к решению  $V$ :

$$V^\lambda \rightharpoonup V. \quad (3.26)$$

Кроме того мы получаем слабую сходимости в  $L_2(Y_T)$   $\nabla V^\lambda$  к  $\nabla V$ :

$$\nabla V^\lambda \rightharpoonup \nabla V \quad (3.27)$$

Неравенства (3.21) дают сильную в  $L_2(Y_T)$  сходимости последовательностей  $\operatorname{div} V^\lambda$  и  $(1-\chi)V^\lambda$  к нулю:

$$\operatorname{div} V^\lambda \rightarrow 0, \quad (3.28)$$

$$(1-\chi)V^\lambda \rightarrow 0 \quad (3.29)$$

Из (3.27) и (3.28) следует, что

$$\operatorname{div} V = 0 \quad (3.30)$$

из (3.26) и (3.28) следует

$$V(y,t) = 0, \quad y \in Y_s, \quad (3.31)$$

Краевое условие  $V(y,t) = 0, \quad y \in \gamma$  вытекает из утверждений (3.31), (3.26) и принадлежности функции  $V$  пространству  $L^2((0,T); \overset{o}{W}_{2,\Pi}(Y))$ .

Рассмотрим интегральное тождество (3.7) при  $\lambda \rightarrow \infty$ . Члены  $\lambda \int_{Y_T} \operatorname{div} V^\lambda : \operatorname{div} \eta \, dy \, dt$  и  $\lambda \int_{Y_T} (1 - \chi(y)) V^\lambda \eta \, dy \, dt$  обращаются в нуль в силу того, что  $\operatorname{div} \eta = 0$ , функция  $\chi(y)$  ограничена, функцию  $\eta(y,t)$  продолжаем нулем в  $Y_s$ .

Таким образом, функция  $V(y,t) = \lim_{\lambda \rightarrow \infty} V^\lambda$  удовлетворяет интегральному тождеству (3.4). Неравенство (3.20) дает нам требуемую оценку (3.11).

## 4. Программная реализация

### 4.1. Выбор среды разработки

В качестве среды разработки был выбран Visual Studio Express для Desktop2013 одна из редакций Visual Studio Express. Visual Studio – линейка продуктов компании Microsoft включающих интегрированную среду разработки программного обеспечения. Данные продукты позволяют разрабатывать как консольные, так и приложения с графическим интерфейсом, а так же веб-приложения [8]. Включает в себя редактор исходного кода, встроенный отладчик (может работать как отладчик исходного кода и как отладчик машинного уровня). Остальные встроенные инструменты включают в себя редактор форм для упрощения создания графического интерфейса, веб-редактор, дизайнер классов и дизайнер схемы базы данных.

VisualStudioExpress – набор легковесных сред разработки, представляющих собой урезанную версию VisualStudio. Она включает в себя набор инструментов, в отличие от полных версий: отсутствует дизайнер классов и многие другие инструменты, а так же поддержка плагинов и удаленных баз данных[4].

Ключевыми особенностями express-версий являются:

- ориентирование на цель разработки, а не на язык;
- необходимость регулярно продлевать регулярную регистрацию для индивидуальных разработчиков, если разработка на express-версии ведется не с целью обучения;
- поддержка компиляции 64-битного кода;
- поддержка unit-текстов.

VisualStudioExpress для Desktop используется для разработки обычных десктопных (классических) приложений и позволяет воспользоваться всеми преимуществами Windowsc помощью конструкторов XAMLпроизводительной среды разработки и различных языков программирования, в том числе C#, VisualBasic и C++. Десктопные приложения - это программы, логика которых

требует наличия оператора, содержащие в себе всю полную функциональность и способные работать отдельно на любой машине изолированно от других приложений.

#### **4.1.1 Описание языка программирования**

В качестве языка программирования был выбран C#. C# - типобезопасный объектно-ориентированный язык, предназначенный для разработки разнообразных безопасных и мощных приложений, выполняемых в среде .NETFramework. C# относится к семье языков с C-подобным синтаксисом, из них его синтаксис наиболее близок к C++ и Java. Язык имеет статическую типизацию, поддерживает полиморфизм, перегрузку операторов (операторов явного и не явного приведения типа), делегаты, атрибуты, события, свойства, обобщенные типы и методы, итераторы, анонимные функции с поддержкой замыканий, LINQ (набор функций, расширяющих возможности синтаксиса C#), исключения, комментарии в формате XML.

С помощью языка C# можно создавать обычные приложения Windows, XML-веб-службы, распределенные компоненты, приложения «клиент-сервер», приложения баз данных и др. VisualC# предоставляют развитый редактор кода, конструкторы с удобным пользовательским интерфейсом, встроенный отладчик и другие средства, упрощающие разработку приложений.

#### **4.2. Описание программной реализации**

Данная программа предназначена для применения метода Галеркина к уравнениям Стокса для прямоугольной поверхности. В данном программном приложении будут присутствовать: график функции, область входных данных (область прямоугольника), область вычислений и кнопка начала вычислений, а так же вывод окна оповещений при некорректном вводе данных. Вид запущенной программы в ожидании ввода данных представлен на рисунке (рис. 1)

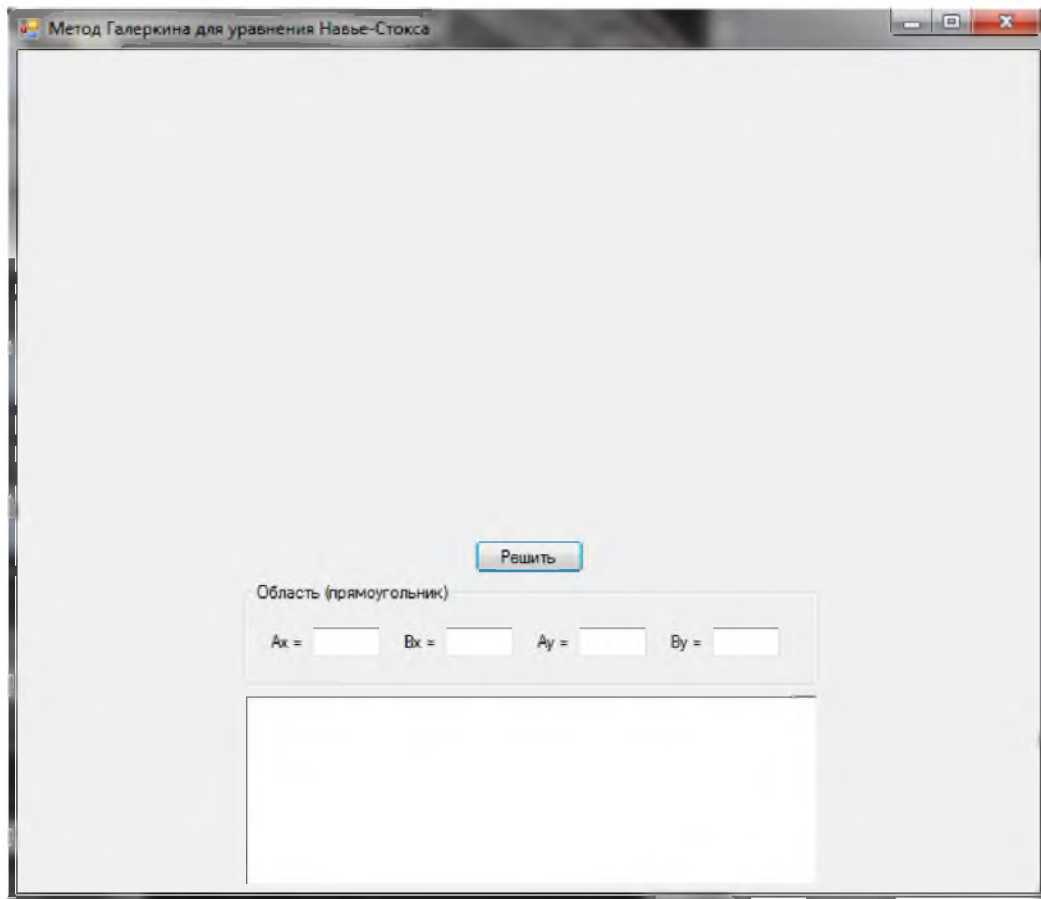


Рис. 1 – Запуск программы.

Мы видим окно ввода данных и кнопку начало решений. В момент написания программы им были присвоены соответствующие действия, а именно для окон ввода данных обработка введенных данных, и проверка их на правильность ввода, и является ли текст числом. Блоки программы отвечающие за вводимые данные:

```
.....  
private void button1_Click(object sender, EventArgs e) //обработка входных  
данных; вывод результата после запуска решения
```



```

{
    graphics.Clear(pictureBox1.BackColor);

    if (!IsNumber(textBox1.Text) || !IsNumber(textBox2.Text) ||
!IsNumber(textBox3.Text) || !IsNumber(textBox4.Text))
        { MessageBox.Show("Неправильный ввод!"); return; }

    float ax = float.Parse(textBox1.Text);

    float bx = float.Parse(textBox2.Text);

    float ay = float.Parse(textBox3.Text);

    float by = float.Parse(textBox4.Text);

    float x = pictureBox1.ClientSize.Width / 2;

    float y = pictureBox1.ClientSize.Height / 2;

    Font font = new Font("Arial", 6, FontStyle.Regular);

    Pen pen = new Pen(Color.Black, 1);

    float Mx = (x - 2 * R) / (Math.Abs(ax) > Math.Abs(bx) ? Math.Abs(ax)
: Math.Abs(bx));

    float My = (y - 2 * R) / (Math.Abs(ay) > Math.Abs(by) ? Math.Abs(ay)
: Math.Abs(by));

    if (My < graphics.MeasureString("1", font).Height) My =
graphics.MeasureString("1", font).Height;

    DrawAxes(pen, font, R, Mx, My, x, y);

    pen.Color = Color.Blue;

```

```

PointF[,] grid = DrawGrid(pen, ax, bx, ay, by, H, K, x, y, Mx, My);

float[,] U = GetFunction(ax, bx, ay, by, H, K);

int m = (int)(1 + Math.Ceiling((bx - ax) / H));

int n = U.Length / m;

richTextBox1.Text="";

font = new Font("Arial", 3, FontStyle.Regular);

for (int i = 0; i < m; i++)

    for (int j = 0; j < n; j++)

        {

            richTextBox1.Text += string.Format("u({0}, {1}) = {2:F15}",

grid[i, j].X, grid[i, j].Y, U[i, j]) + "\n";

            graphics.DrawString(string.Format("{0:F3}", U[i, j]), font,

Brushes.Green, x + grid[i, j].X * Mx, y - grid[i, j].Y * My + 3);

        }

}

private bool IsNumber(string text) //проверка, является ли текст числом

{

    try

    {

        float.Parse(text);

        return true;

    }

}

```

```
catch (FormatException)
```

```
{
```

```
    return false;
```

```
}
```

```
}
```

.....

Соответственно при вводе текста не являющимся числом, программа не сможет корректно работать и известит пользователя о том что была допущена ошибка (рис.2).

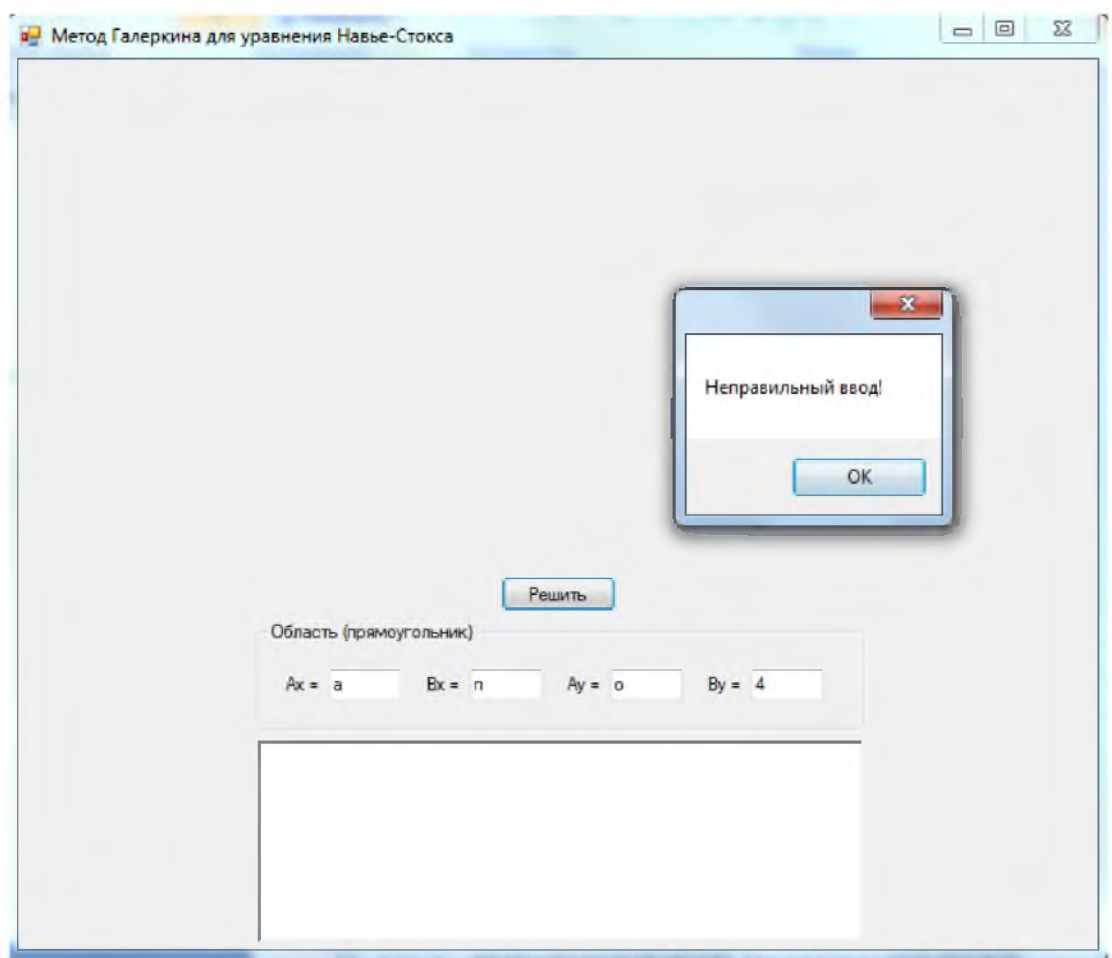


Рис.2 –извещение о неправильном вводе

Часть кода отвечающая за вывод представленного окна на (рис.2) представлена ниже:

```
graphics.Clear(pictureBox1.BackColor);  
  
if (!IsNumber(textBox1.Text) || !IsNumber(textBox2.Text) ||  
!IsNumber(textBox3.Text) || !IsNumber(textBox4.Text))  
{ MessageBox.Show("Неправильный ввод!"); return; }
```

Как мы видим из кода, программа не выполнит запрос на решение и после подтверждения ошибки вернет нас на начало по средствам цикла. После чего нам просто необходимо вести данные удовлетворяющие установленному условию и продолжить работу.

Рассмотрим область для построения осей и выполнения дальнейшего решения, опираясь на них. Как мы могли заметить в окне программы по мимо окон ввода и кнопки запуска , есть область отвечающая за вывод данных решения поставленной задачи, и большая область для построения графика, которая находится выше всех имеющихся объектов. Часть кода для рисования осей имеет большой объем, потому что график занимает большую часть окна программы и является основной частью данной программы.

```
private void DrawAxes(Pen pen, Font font, int indentation, float scaleX, float  
scaleY, float middleX, float middleY) //рисует оси координат  
{  
    pen.SetLineCap(LineCap.NoAnchor, LineCap.ArrowAnchor,  
DashCap.Triangle);  
    graphics.DrawLine(pen, 10, middleY, 2 * middleX - 10, middleY);
```

```

graphics.DrawLine(pen, middleX, 2 * middleY - 10, middleX, 10);

pen.SetLineCap(LineCap.NoAnchor,          LineCap.NoAnchor,
DashCap.Flat);

graphics.DrawString("x", font, Brushes.Red, 2 * middleX - 10,
middleY + 8);

graphics.DrawString("y", font, Brushes.Red, middleX + 5, 10);

graphics.DrawString("0", font, Brushes.Black, middleX +
graphics.MeasureString("0", font).Width, middleY + 3);

int n = 1;

for (float i = middleX + scaleX; i < 2 * middleX - 10; i += scaleX, n++)
{
    graphics.DrawLine(pen, i, middleY + 3, i, middleY - 3);

    graphics.DrawString(n.ToString(), font, Brushes.Black, i -
graphics.MeasureString((-n).ToString(), font).Width / 2, middleY + 8);

    graphics.DrawLine(pen, i - 2 * scaleX * n, middleY + 3, i - 2 *
scaleX * n, middleY - 3);

    graphics.DrawString((-n).ToString(), font, Brushes.Black, (i - 2 *
scaleX * n) - graphics.MeasureString((-n).ToString(), font).Width / 2, middleY + 8);
}

n = 1;

for (float i = middleY - scaleY; i > 10; i -= scaleY, n++)
{

```

```

graphics.DrawLine(pen, middleX - 3, i, middleX + 3, i);

graphics.DrawString(n.ToString(), font, Brushes.Black, middleX +
graphics.MeasureString(n.ToString(), font).Width, i);

graphics.DrawLine(pen, middleX - 3, i + 2 * scaleY * n, middleX +
3, i + 2 * scaleY * n);

graphics.DrawString((-n).ToString(), font, Brushes.Black, middleX +
graphics.MeasureString((-n).ToString(), font).Width, i + 2 * scaleY * n);

    }
}

```

Здесь мы наблюдаем, как программа, опираясь на введенные данные, строит ось X и Y ,имея начальную точку отчета «0». Однако для этого куда есть определенные ограничения , виде отступов от края окна, это сделано для удобство использования, и соответственно чтобы график не выходил за отведенное ему место. Данное ограничение установлена с помощью представленного ниже кода:

```
const int R = 10; //отступ от края при рисовании осей
```

```
Graphics graphics;
```

Из которого мы видим , что отступ от края составляет 1 см, и программа ни как не может нарушить данное ограничение.

Итак, выполняя финальную часть работы, программа должна нарисовать сетку по заданным координатам, вывести ее на график , и показать нам найденные точки в области отвечающей за вывод данных.

```

private PointF[,] DrawGrid(Pen pen, float ax, float bx, float ay, float by, float
h, float k, float middleX, float middleY, float scaleX, float scaleY) //рисует сетку
{
    PointF[,] Grid = new PointF[(int)(1 + Math.Ceiling((bx - ax) / h)),
(int)(1 + Math.Ceiling((by - ay) / k))];

    int i = 0, j = 0;

    Color color = pen.Color;

    bool drawing = true;

    for (float x = ax; x <= bx; x += h, i++, j = 0)
    {
        x = (float)Math.Round(x, 1);

        graphics.DrawLine(pen, middleX + x * scaleX, middleY - ay *
scaleY, middleX + x * scaleX, middleY - by * scaleY);

        for (float y = ay; y <= by; y += k, j++)
        {
            y = (float)Math.Round(y, 1);

            if (drawing) graphics.DrawLine(pen, middleX + ax * scaleX,
middleY - y * scaleY, middleX + bx * scaleX, middleY - y * scaleY);

            Grid[i, j] = new PointF(x, y);

            graphics.DrawEllipse(pen, -1 + middleX + x * scaleX, -1 +
middleY - y * scaleY, 2, 2);

        }
    }
}

```

```

        drawing = false;
    }

    pen.Color = Color.Red;

    graphics.DrawLine(pen, middleX + ax * scaleX, middleY - ay *
scaleY, middleX + ax * scaleX, middleY - by * scaleY);

    graphics.DrawLine(pen, middleX + bx * scaleX, middleY - ay *
scaleY, middleX + bx * scaleX, middleY - by * scaleY);

    graphics.DrawLine(pen, middleX + ax * scaleX, middleY - ay *
scaleY, middleX + bx * scaleX, middleY - ay * scaleY);

    graphics.DrawLine(pen, middleX + ax * scaleX, middleY - by *
scaleY, middleX + bx * scaleX, middleY - by * scaleY);

    pen.Color = Color.Red;

    return Grid;

}

private float[,] GetFunction(float ax, float bx, float ay, float by, float h,
float k)
{
    int M = (int)(1 + Math.Ceiling((bx - ax) / h));
    int N = (int)(1 + Math.Ceiling((by - ay) / k));
    float E = 0.0000001F;
    float[,] u = new float[2][,];

```



```

u[0]=new float[M,N];
u[1]=new float[M,N];

float x, y;

int i = 0, j = 0;

for (y = ay; y <= by; y += k, j++)
{
    u[0][0, j] = u[0][M - 1, j] = g(y);
    u[1][0, j] = u[1][M - 1, j] = g(y);
}

for (x = ax; x <= bx; x += h, i++)
{
    u[0][i, N - 1] = f(x); u[0][i, 0] = f(x);
    u[1][i, N - 1] = f(x); u[1][i, 0] = f(x);
}

const int m = 6; i = 0; j = 0;

float Htemp = (bx - ax) / m, Ktemp = (by - ay) / m;

float[] Xtemp = new float[m];

float[] Ytemp = new float[m];

float[,] Utemp = new float[m, m];

for (x = ax; x <= bx && i < m; x += Htemp, i++) { Xtemp[i] = x;
Utemp[i, m - 1] = f(x); Utemp[i, 0] = f(x); }

```

```

    for (y = ay; y <= by && j < m; y += Ktemp, j++) { Ytemp[j] = y;
Utemp[0, j] = Utemp[m - 1, j] = g(y); }

    int n = (m - 2) * (m - 2);

    float[,] A = new float[n, n];

    A[0, 0] = -4;

    A[0, 1] = 1;

    for (i = 1; i < n - 1; i++)
    {
        if (i % 4 != 0) A[i - 1, i] = 1;

        A[i, i] = -4;

        if ((i + 1) % 4 != 0) A[i, i + 1] = 1;

        if (i - 4 > 0) A[i, i - 4] = 1;

        if (i + 4 < n) A[i, i + 4] = 1;
    }

    A[n - 2, n - 1] = 1;

    A[n - 1, n - 1] = -4;

    float[] B = new float[n];

    B[0] = -Utemp[0, 1] - Utemp[1, 0];

    B[1] = -Utemp[2, 0];

    B[2] = -Utemp[3, 0];

    B[3] = -Utemp[4, 0] - Utemp[5, 1];

    B[4] = -Utemp[0, 2];

```

```

B[5] = 0;
B[6] = 0;
B[7] = -Utemp[5, 2];
B[8] = -Utemp[0, 3];
B[9] = 0;
B[10] = 0;
B[11] = -Utemp[5, 3];
B[12] = -Utemp[0, 4] - Utemp[1, 5];
B[13] = -Utemp[2, 5];
B[14] = -Utemp[3, 5];
B[15] = -Utemp[4, 5];
Utemp = SolveSLE(A, B, Utemp, m - 2, 1);
i = 1; j = 1;
for (x = ax + h; x <= bx - h; x += h, i++, j = 1)
    for (y = ay + k; y <= by - k; y += k, j++)
        u[0][i, j] = L(x, y, Xtemp, Ytemp, Utemp, m);
int l = 0;
while (Math.Abs(u[0][1, 1]) - Math.Abs(u[1][1, 1]) > E)
{
    l = l - 1;
    for (i = 1; i < M - 1; i++)
        for (j = 1; j < N - 1; j++)

```

```

        u[l][i, j] = (u[l - 1][i + 1, j] + u[l - 1][i - 1, j] + u[l - 1][i, j + 1] +
u[l - 1][i, j - 1]) / 4;
    }

    return u[l];
}

private float f(float x)
{
    return (float)(2 * Math.Pow((x - 1.5), 2) / 3);
}

private float g(float y)
{
    return 0.75F * y * y * y * (float)Math.Sin(y);
}

private float L(float x, float y, float[] X, float[] Y, float[,] u, int N)
{
    float F = 0, s = 1;

    for (int m = 0; m < N; m++)
    {
        for (int n = 0; n < N; n++)
        {
            for (int i = 0; i < N; i++)
            {

```

```

        if (i != m) s *= (x - X[i]) / (X[m] - X[i]);

        for (int j = 0; j < N; j++)
        {
            if (j != n) s *= (y - Y[j]) / (Y[n] - Y[j]);
        }
    }

    F += u[m, n] * s;
}

}

if (float.IsNaN(F) || float.IsInfinity(F)) return L(x - 0.1F, y - 0.1F, X, Y,
u, N);

return F;
}

private float[,] SolveSLE(float[,] A, float[] B, float[,] X, int m, int offset)
{
    int i, j, k, n = m * m;

    float c;

    for (k = 0; k < n - 1; k++)
    {
        for (i = k + 1; i < n; i++)
        {
            c = A[i, k] / A[k, k];

```

```

    B[i] = B[i] - (c * B[k]);
    for (j = k + 1; j < n; j++)
    {
        A[i, j] = A[i, j] - (c * A[k, j]);
    }
}
}
X[offset + (n - 1) / m, offset + (n - 1) % m] = B[n - 1] / A[n - 1, n - 1];
for (i = n - 2; i >= 0; i--)
{
    c = 0;
    for (j = i + 1; j < n; j++)
    {
        c += A[i, j] * X[offset + j / m, offset + j % m];
    }
    X[offset + i / m, offset + i % m] = (B[i] - c) / A[i, i];
}
return X;
}
}
}

```

### 4.3 Проверка работоспособности программного продукта

После рассмотрения составляющих программы, и выяснения некоторых ее аспектов, необходимо убедиться в ее работоспособности пошагово выполняя описанные ранее инструкции.

Программа имеет файл запуска с разрешением «.exe», для удобства в использовании.

Шаг 1. Запускаем файл программы и видим уже знакомое нам окно ожидания команд (рис.3)

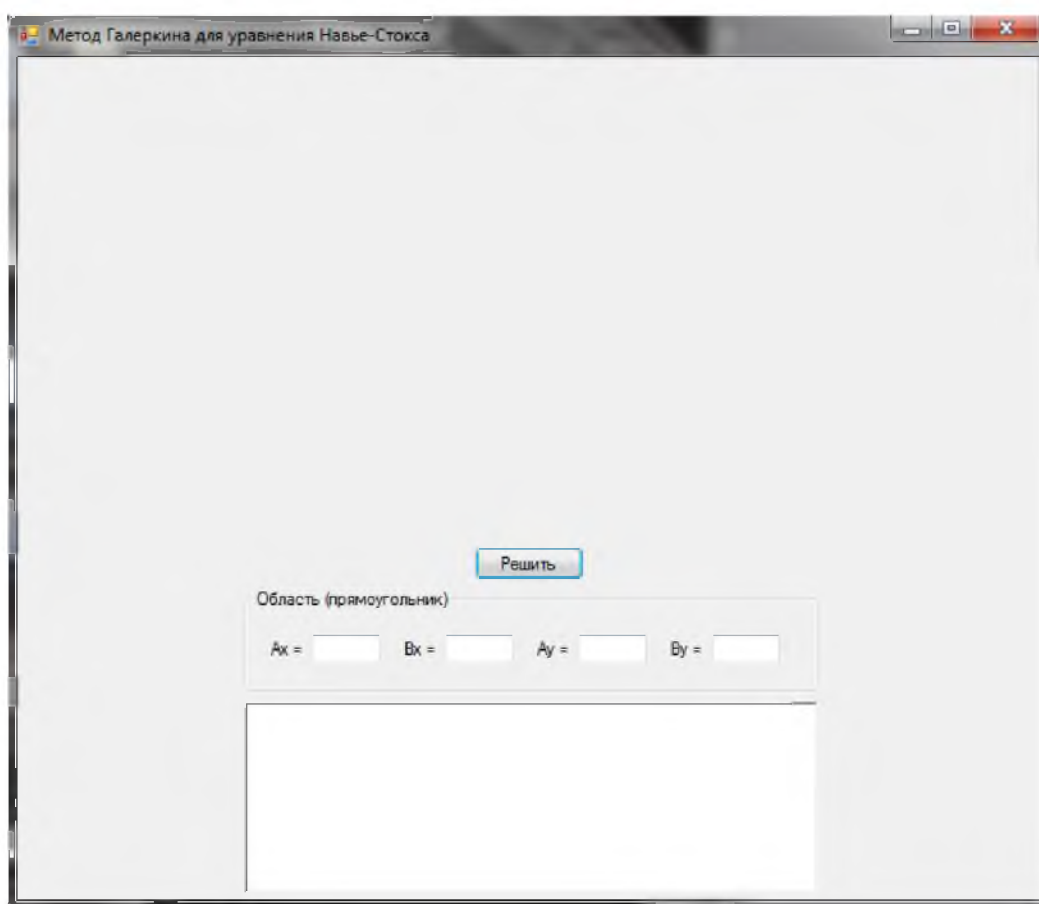


Рис.3

Шаг 2. Вводим данные, заранее убедившись в правильности ввода, и в том что текст является числом (рис.4). Мы используем следующие координаты ( $A_x = (-1)$ ;  $B_x = 2$ ;  $A_y = (-2)$ ;  $B_y = 1$ ).

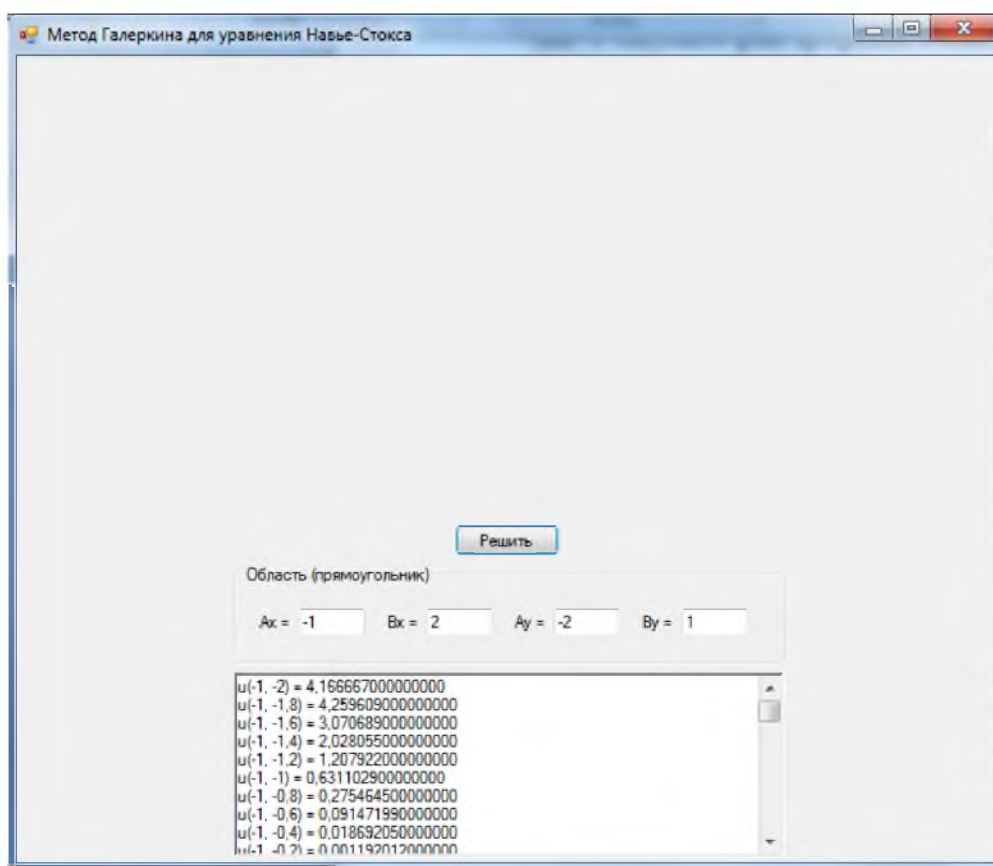


Рис. 4

Шаг 3. Нажимаем кнопку запуска программы «Решить». Поскольку данные введены корректно, программа немедленно приступает к рисованию сетки на построенных осях (рис.5). Следует заметить, что чем больше интервал, тем больше времени займет прорисовка всех точек и выстраивание их, когда, не смотря на это окно ввода выдает точки и их координат не зависимо от того, была ли закончена сетка, или нет.



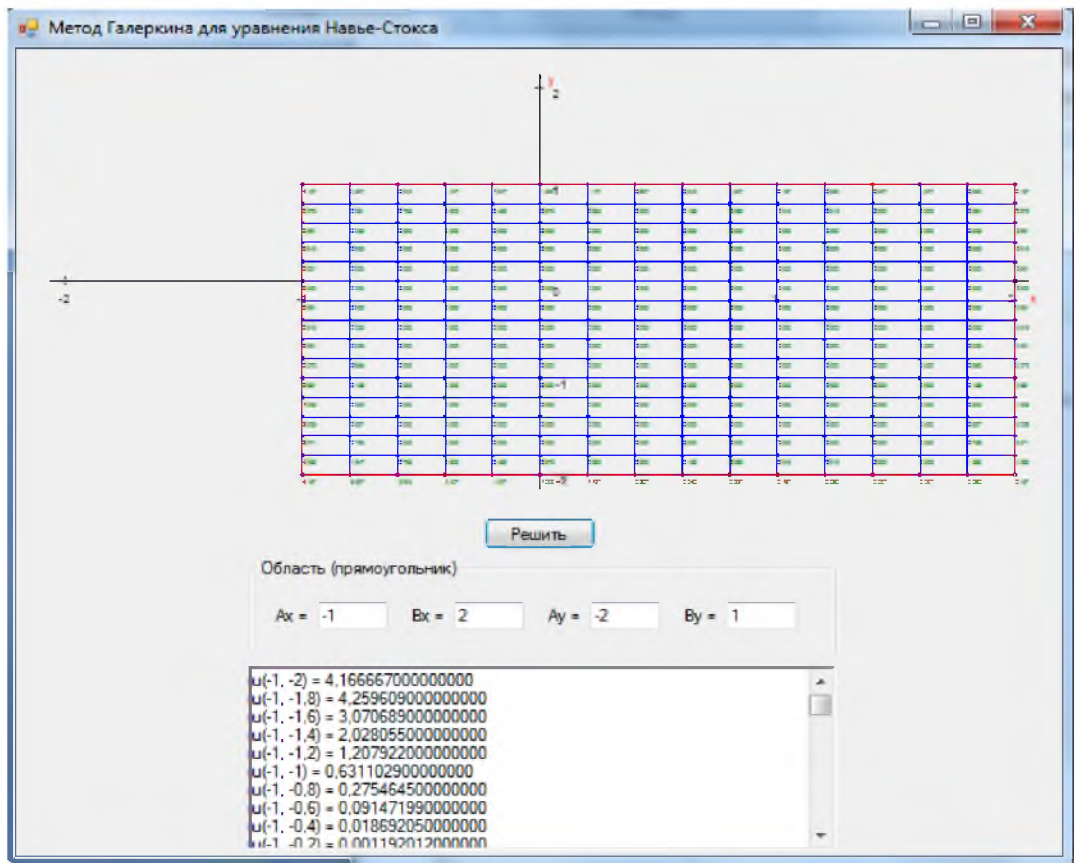


Рис. 5

Как было сказано ранее, после мы имеем право завершить программу либо продолжить работать с ней, используя другие данные.

Можно сделать вывод, что программа работает корректно, строит необходимые визуальные объекты с использованием палитры цветов, для удобного восприятия, однако отказавшись от 3D модели, не удалось избежать проблем связанных с компактностью выводимой сетки на прямую зависящей от введенных данных. Это потребовало добавления окна вывода точек координат для компенсации этой потребности.

## ЗАКЛЮЧЕНИЕ

В работе была рассмотрена нестационарная система Стокса, описывающая движение вязкой несжимаемой жидкости в упругом пористом теле. Получены априорные оценки решения. Доказана однозначная разрешимость задачи.

Проведен подробный анализ литературы по проблеме исследования, краткий обзор исследований, связанных с темой работы, изучены основные теоретические положения, необходимые при исследовании краевых задач для дифференциальных уравнений.

Раскрыты основные особенности схемы метода Галёркина, приведен ряд примеров, иллюстрирующих применение метода.

На основании изученного теоретического материала была разработана программная реализация метода Галёркина, проанализировано ее быстродействие, проведен тестовый расчет, построен графики полученного численного решения.

## СПИСОК ЛИТЕРАТУРЫ

1. Meirmanov A. Derivation of equations of seismic and acoustic wave propagation and equations of filtration via homogenization of periodic structures // Journal of Mathematical Sciences. – 2009. – 163;2 – P.111 – 172.
2. Некрасова И.В. Математические модели гидравлического удара в вязкой жидкости // Сибирские электронные математические известия. – 2012. – Т. 9. – С.274 – 293. – <http://semr.math.nsc.ru/v9/p227-246.pdf>.
3. Олейник О.А. Лекции об уравнениях с частными производными. – М.: БИНОМ. Лаборатория знаний, 2005. – 260 с.
4. Ладыженская О.А. Математические вопросы динамики вязкой несжимаемой жидкости. – М.: Наука, 1970. – 288 с.
5. Ладыженская О.А. Краевые задачи математической физики. – М.: Наука, 1973. – 408с.
6. Михайлов В.П. Дифференциальные уравнения в частных производных. – М.: Наука, 1976. – 391 с.
7. Ладыженская О.А., Солонников В.А., Уральцева Н.Н. Линейные и квазилинейные уравнения параболического типа. – М.: Наука, 1967. – 736 с.
8. Петров Г. И. Применение метода Галёркина к задаче об устойчивости течения вязкой жидкости // Прикладная математика и механика. Новая серия. – 1940. – Т. 4. – Вып. 3. – с. 3– 11.
9. Bickley W.G. – Phil. Mag., 1941, v. 32, p. 50 – 66.
10. Orzag S.A. Numerical simulation of turbulent flows. – In: Handbook of turbulence, Ed. W. Frost, T.H. Moulden – New York: Plenum Press, 1977, p. 281 – 313.

11. Bourke W., McAveney B., Puri K., Thurling R. – Meth. In Comp. Phys., 1977. v. 17, p. 267 – 325.
12. Oden J.T. Finite elements of nonlinear continua. – New York: Mc Graw-Hill. 1972. [Перевод: Оден Дж. Конечные элементы в нелинейной механике сплошных сред. – М. Мир, 1976.]
13. Канторович Л.В., Крылов В.И. Приближенные методы высшего анализа. – М.: Физматгиз, 1962.
14. Dryden H.L., Murnaghan F.P., Bateman H. – Hydrodynamics, New York: Dover, 1956, p. 197.
15. Finlayson B.A. – Brit. Chem. Eng., 1969, v. 14, p. 179 – 182.
16. Ворович И.И. О методе Бубнова – Галеркина в нелинейной теории колебания пологих оболочек. – Доклады АН СССР, 1956. – т. 110. – №5. – с. 723 – 726.
17. Галёркин Б.Г. Стержни и пластинки. Ряды в некоторых вопросах упругого равновесия стержней и пластинок. // Вестник инженеров. – 1915. – т.1. – с. 897 – 908.
18. Канторович Л.В., Крылов В.И. Приближенные методы высшего анализа. – 5-е изд. – Л.-М., 1962.
19. Михлин С.Г. Вариационные методы в математической физике. – 2-е изд. – М.-Л. – 1970.
20. Флетчер К. Численные методы на основе метода Галеркина. – М. – Мир – 1988.
21. Ritz W., Neue Methode zur Lösung gewisser Randwertaufgaben, «Gesellschaft der Wissenschaften zu Göttingen. Math.-physik. Klasse. Nachrichten», Göttingen, 1908.

22. Википедия. [Электронный ресурс]. Метод Галёркина. Режим доступа:  
[https://ru.wikipedia.org/wiki/Метод\\_Галёркина](https://ru.wikipedia.org/wiki/Метод_Галёркина).
23. Википедия. [Электронный ресурс]. Дифференциальное уравнение в частных производных. Режим доступа:  
[https://ru.wikipedia.org/wiki/Дифференциальное\\_уравнение\\_в\\_частных\\_производных](https://ru.wikipedia.org/wiki/Дифференциальное_уравнение_в_частных_производных).

## ПРИЛОЖЕНИЕ.

```
using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Drawing.Drawing2D;

using System.Linq;

using System.Text;

using System.Windows.Forms;

namespace WindowsFormsApplication30
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();

            graphics = pictureBox1.CreateGraphics();
        }

        const float H = 0.2F, K = 0.2F;

        const int R = 10; //отступ от края при рисовании осей

        Graphics graphics;
```

```

private void button1_Click(object sender, EventArgs e) //обработка
ВХОДНЫХ ДАННЫХ; ВЫВОД РЕЗУЛЬТАТА ПОСЛЕ ЗАПУСКА РЕШЕНИЯ
{
    graphics.Clear(pictureBox1.BackColor);

    if (!IsNumber(textBox1.Text) || !IsNumber(textBox2.Text) ||
!IsNumber(textBox3.Text) || !IsNumber(textBox4.Text))
    { MessageBox.Show("Неправильный ввод!"); return; }

    float ax = float.Parse(textBox1.Text);
    float bx = float.Parse(textBox2.Text);
    float ay = float.Parse(textBox3.Text);
    float by = float.Parse(textBox4.Text);

    float x = pictureBox1.ClientSize.Width / 2;
    float y = pictureBox1.ClientSize.Height / 2;

    Font font = new Font("Arial", 6, FontStyle.Regular);
    Pen pen = new Pen(Color.Black, 1);

    float Mx = (x - 2 * R) / (Math.Abs(ax) > Math.Abs(bx) ? Math.Abs(ax)
: Math.Abs(bx));

    float My = (y - 2 * R) / (Math.Abs(ay) > Math.Abs(by) ? Math.Abs(ay)
: Math.Abs(by));

    if (My < graphics.MeasureString("1", font).Height) My =
graphics.MeasureString("1", font).Height;

    DrawAxes(pen, font, R, Mx, My, x, y);
}

```

```

pen.Color = Color.Blue;

PointF[,] grid = DrawGrid(pen, ax, bx, ay, by, H, K, x, y, Mx, My);

float[,] U = GetFunction(ax, bx, ay, by, H, K);

int m = (int)(1 + Math.Ceiling((bx - ax) / H));

int n = U.Length / m;

richTextBox1.Text="";

font = new Font("Arial", 3, FontStyle.Regular);

for (int i = 0; i < m; i++)

    for (int j = 0; j < n; j++)

        {

            richTextBox1.Text += string.Format("u({0}, {1}) = {2:F15}",

grid[i, j].X, grid[i, j].Y, U[i, j]) + "\n";

            graphics.DrawString(string.Format("{0:F3}", U[i, j]), font,

Brushes.Green, x + grid[i, j].X * Mx, y - grid[i, j].Y * My + 3);

        }

}

private bool IsNumber(string text) //проверка, является ли текст числом

{

    try

    {

        float.Parse(text);

        return true;

    }

}

```



```

    }

    catch (FormatException)

    {

        return false;

    }

}

private void DrawAxes(Pen pen, Font font, int indentation, float scaleX,
float scaleY, float middleX, float middleY) //рисует оси координат
{

    pen.SetLineCap(LineCap.NoAnchor,          LineCap.ArrowAnchor,
DashCap.Triangle);

    graphics.DrawLine(pen, 10, middleY, 2 * middleX - 10, middleY);
    graphics.DrawLine(pen, middleX, 2 * middleY - 10, middleX, 10);

    pen.SetLineCap(LineCap.NoAnchor,          LineCap.NoAnchor,
DashCap.Flat);

    graphics.DrawString("x", font, Brushes.Red, 2 * middleX - 10,
middleY + 8);

    graphics.DrawString("y", font, Brushes.Red, middleX + 5, 10);

    graphics.DrawString("0", font, Brushes.Black, middleX +
graphics.MeasureString("0", font).Width, middleY + 3);

    int n = 1;

    for (float i = middleX + scaleX; i < 2 * middleX - 10; i += scaleX, n++)

```

```

    {
        graphics.DrawLine(pen, i, middleY + 3, i, middleY - 3);
        graphics.DrawString(n.ToString(), font, Brushes.Black, i -
graphics.MeasureString((-n).ToString(), font).Width / 2, middleY + 8);
        graphics.DrawLine(pen, i - 2 * scaleX * n, middleY + 3, i - 2 *
scaleX * n, middleY - 3);
        graphics.DrawString((-n).ToString(), font, Brushes.Black, (i - 2 *
scaleX * n) - graphics.MeasureString((-n).ToString(), font).Width / 2, middleY + 8);
    }
    n = 1;
    for (float i = middleY - scaleY; i > 10; i -= scaleY, n++)
    {
        graphics.DrawLine(pen, middleX - 3, i, middleX + 3, i);
        graphics.DrawString(n.ToString(), font, Brushes.Black, middleX +
graphics.MeasureString(n.ToString(), font).Width, i);
        graphics.DrawLine(pen, middleX - 3, i + 2 * scaleY * n, middleX +
3, i + 2 * scaleY * n);
        graphics.DrawString((-n).ToString(), font, Brushes.Black, middleX +
graphics.MeasureString((-n).ToString(), font).Width, i + 2 * scaleY * n);
    }
}

```

```

private PointF[,] DrawGrid(Pen pen, float ax, float bx, float ay, float by,
float h, float k, float middleX, float middleY, float scaleX, float scaleY) //рисует
сетку

{
    PointF[,] Grid = new PointF[(int)(1 + Math.Ceiling((bx - ax) / h)),
(int)(1 + Math.Ceiling((by - ay) / k))];

    int i = 0, j = 0;

    Color color = pen.Color;

    bool drawing = true;

    for (float x = ax; x <= bx; x += h, i++, j = 0)
    {
        x = (float)Math.Round(x, 1);

        graphics.DrawLine(pen, middleX + x * scaleX, middleY - ay *
scaleY, middleX + x * scaleX, middleY - by * scaleY);

        for (float y = ay; y <= by; y += k, j++)
        {
            y = (float)Math.Round(y, 1);

            if (drawing) graphics.DrawLine(pen, middleX + ax * scaleX,
middleY - y * scaleY, middleX + bx * scaleX, middleY - y * scaleY);

            Grid[i, j] = new PointF(x, y);

            graphics.DrawEllipse(pen, -1 + middleX + x * scaleX, -1 +
middleY - y * scaleY, 2, 2);

```

```

    }

    drawing = false;
}

pen.Color = Color.Red;

graphics.DrawLine(pen, middleX + ax * scaleX, middleY - ay *
scaleY, middleX + ax * scaleX, middleY - by * scaleY);

graphics.DrawLine(pen, middleX + bx * scaleX, middleY - ay *
scaleY, middleX + bx * scaleX, middleY - by * scaleY);

graphics.DrawLine(pen, middleX + ax * scaleX, middleY - ay *
scaleY, middleX + bx * scaleX, middleY - ay * scaleY);

graphics.DrawLine(pen, middleX + ax * scaleX, middleY - by *
scaleY, middleX + bx * scaleX, middleY - by * scaleY);

pen.Color = Color.Red;

return Grid;
}

private float[,] GetFunction(float ax, float bx, float ay, float by, float h,
float k)
{
    int M = (int)(1 + Math.Ceiling((bx - ax) / h));
    int N = (int)(1 + Math.Ceiling((by - ay) / k));
    float E = 0.0000001F;
    float[,] u = new float[2][,];

```

```

u[0]=new float[M,N];
u[1]=new float[M,N];

float x, y;

int i = 0, j = 0;

for (y = ay; y <= by; y += k, j++)
{
    u[0][0, j] = u[0][M - 1, j] = g(y);
    u[1][0, j] = u[1][M - 1, j] = g(y);
}

for (x = ax; x <= bx; x += h, i++)
{
    u[0][i, N - 1] = f(x); u[0][i, 0] = f(x);
    u[1][i, N - 1] = f(x); u[1][i, 0] = f(x);
}

const int m = 6; i = 0; j = 0;

float Htemp = (bx - ax) / m, Ktemp = (by - ay) / m;

float[] Xtemp = new float[m];

float[] Ytemp = new float[m];

float[,] Utemp = new float[m, m];

for (x = ax; x <= bx && i < m; x += Htemp, i++) { Xtemp[i] = x;
Utemp[i, m - 1] = f(x); Utemp[i, 0] = f(x); }

```

```

    for (y = ay; y <= by && j < m; y += Ktemp, j++) { Ytemp[j] = y;
Utemp[0, j] = Utemp[m - 1, j] = g(y); }

    int n = (m - 2) * (m - 2);

    float[,] A = new float[n, n];

    A[0, 0] = -4;

    A[0, 1] = 1;

    for (i = 1; i < n - 1; i++)
    {
        if (i % 4 != 0) A[i - 1, i] = 1;

        A[i, i] = -4;

        if ((i + 1) % 4 != 0) A[i, i + 1] = 1;

        if (i - 4 > 0) A[i, i - 4] = 1;

        if (i + 4 < n) A[i, i + 4] = 1;
    }

    A[n - 2, n - 1] = 1;

    A[n - 1, n - 1] = -4;

    float[] B = new float[n];

    B[0] = -Utemp[0, 1] - Utemp[1, 0];

    B[1] = -Utemp[2, 0];

    B[2] = -Utemp[3, 0];

    B[3] = -Utemp[4, 0] - Utemp[5, 1];

    B[4] = -Utemp[0, 2];

```

```

B[5] = 0;
B[6] = 0;
B[7] = -Utemp[5, 2];
B[8] = -Utemp[0, 3];
B[9] = 0;
B[10] = 0;
B[11] = -Utemp[5, 3];
B[12] = -Utemp[0, 4] - Utemp[1, 5];
B[13] = -Utemp[2, 5];
B[14] = -Utemp[3, 5];
B[15] = -Utemp[4, 5];
Utemp = SolveSLE(A, B, Utemp, m - 2, 1);
i = 1; j = 1;
for (x = ax + h; x <= bx - h; x += h, i++, j = 1)
    for (y = ay + k; y <= by - k; y += k, j++)
        u[0][i, j] = L(x, y, Xtemp, Ytemp, Utemp, m);
int l = 0;
while (Math.Abs(u[0][1, 1]) - Math.Abs(u[1][1, 1]) > E)
{
    l = l - 1;
    for (i = 1; i < M - 1; i++)
        for (j = 1; j < N - 1; j++)

```

```

        u[l][i, j] = (u[l - 1][i + 1, j] + u[l - 1][i - 1, j] + u[l - 1][i, j + 1] +
u[l - 1][i, j - 1]) / 4;
    }

    return u[l];
}

private float f(float x)
{
    return (float)(2 * Math.Pow((x - 1.5), 2) / 3);
}

private float g(float y)
{
    return 0.75F * y * y * y * (float)Math.Sin(y);
}

private float L(float x, float y, float[] X, float[] Y, float[,] u, int N)
{
    float F = 0, s = 1;

    for (int m = 0; m < N; m++)
    {
        for (int n = 0; n < N; n++)
        {
            for (int i = 0; i < N; i++)
            {

```



```

        if (i != m) s *= (x - X[i]) / (X[m] - X[i]);

        for (int j = 0; j < N; j++)
        {
            if (j != n) s *= (y - Y[j]) / (Y[n] - Y[j]);
        }

        F += u[m, n] * s;
    }

    }

    if (float.IsNaN(F) || float.IsInfinity(F)) return L(x - 0.1F, y - 0.1F, X, Y,
u, N);

    return F;
}

private float[,] SolveSLE(float[,] A, float[] B, float[,] X, int m, int offset)
{
    int i, j, k, n = m * m;

    float c;

    for (k = 0; k < n - 1; k++)
    {
        for (i = k + 1; i < n; i++)
        {
            c = A[i, k] / A[k, k];

```

```

    B[i] = B[i] - (c * B[k]);
    for (j = k + 1; j < n; j++)
    {
        A[i, j] = A[i, j] - (c * A[k, j]);
    }
}
}
X[offset + (n - 1) / m, offset + (n - 1) % m] = B[n - 1] / A[n - 1, n - 1];
for (i = n - 2; i >= 0; i--)
{
    c = 0;
    for (j = i + 1; j < n; j++)
    {
        c += A[i, j] * X[offset + j / m, offset + j % m];
    }
    X[offset + i / m, offset + i % m] = (B[i] - c) / A[i, i];
}
return X;
}
}
}

```