

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования
**«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ТОМСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ»**

О.В. Гальцева

**ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ
В ПРИБОРОСТРОЕНИИ**

Учебное пособие

Томск
2020

УДК 681.2:004(076.5)
ББК 34.9:32.81я73
Г177

Гальцева О.В.

Г177 Информационные технологии в приборостроении: учебное пособие / О. В. Гальцева. – Томск : Издательство ООО «Вайар», 2020.— 96 с.

В пособии содержится материал по работе с программными продуктами, которые позволяют решать задачи реального промышленного производства, дефектоскопии и приборостроения. Основное внимание уделено изучению таких программных продуктов, как Mathcad, MATLAB, Elcut, C++. Так же приведены примеры и задания по измерению, расчету, исследованию и моделированию технических объектов, процессов и систем.

Пособие предназначено для магистрантов направления «Приборостроение».

УДК 681.2:004(076.5)
ББК 34.9:32.81я73

Рецензенты

Кандидат тех. наук, доцент Омского государственного
технического университета, г. Омск
В.В. Кузнецов

Кандидат тех. наук, начальник лаборатории АО «Полюс»,
г. Томск
О.Ю. Завьялова

© Издательство ООО «Вайар», 2020
© Гальцева О.В., 2020

Оглавление

ВВЕДЕНИЕ.....	4
Лабораторная работа 1. Визуализация результатов проектной деятельности в Mathcad.....	6
Лабораторная работа 2. Исследование переходных процессов схем в Mathcad	13
Лабораторная работа 3. Проектирование простейших моделей в программе MatLab	20
Лабораторная работа 4. Виртуальное проектирование технических устройств в программе MatLab	26
Лабораторная работа 5. Реализация процессов широтно-импульсной модуляции в MatLab	38
Лабораторная работа 6. Расчет электростатических полей методом конечных элементов с помощью программного продукта ELCUT	42
Лабораторная работа 7. Решение задач электромагнитной дефектоскопии с применением программы ELCUT	55
Лабораторная работа 8. Тепловая дефектоскопия жилого помещения с помощью ELCUT	60
Лабораторная работа 9. Моделирование приборных устройств с помощью программного продукта ELCUT	66
Лабораторная работа 10. Создание программ в среде разработки Turbo C++.....	71
Лабораторная работа 11. Построение блок-схем вычислительных процессов.....	79
Лабораторная работа 12. Программирование циклических алгоритмов в C++	87
Список использованных источников.....	92

ВВЕДЕНИЕ

Представленное учебное пособие представляет собой методический лабораторный практикум, направленный на практическое закрепление теоретического материала по курсу «Информационные технологии в приборостроении». Этот курс является частью программы по подготовке магистрантов направления «Приборостроение».

Лабораторные работы посвящены изучению базовых и специальных методов и средств аппаратно-программной реализации для проектирования и создания технических объектов и систем приборостроения, промышленной томографии, а также изучение методов и приборов для измерения различных величин.

Занятия проводятся в аудиториях, оснащенных ПЭВМ с индивидуальными местами для выполнения лабораторных работ.

Цикл лабораторных работ состоит из 12 работ.

Работы 1-2 посвящены изучению программного продукта Mathcad. Как известно, этот программный продукт давно нашел широкое применение в инженерных расчетах.

Лабораторная работа 1 посвящена визуализации результатов исследовательской и проектной деятельности. В ходе работы рассматриваются алгоритмы построения и анализа различных типов графиков функций.

Лабораторная работа 2 знакомит с возможностями программы Mathcad для исследования переходных процессов исследуемого технического объекта. В работе рассмотрен алгоритм изучения переходных процессов и их характеристик при различных режимах нагрузки исследуемого объекта, подробный анализ результатов измерений параметров переходных процессов.

Работы 3-5 посвящены изучению программного продукта Matlab для математического моделирования и изучения процессов схем и технических систем.

Лабораторная работа 3 посвящена знакомству с возможностями программы MatLab на примере проектирования простейших моделей схем.

Лабораторная работа 4 знакомит с возможностями программы MatLab для измерения и анализа параметров переходных процессов исследуемого технического объекта. Этот анализ позволяет сделать выводы об устойчивой работе технического устройства в разных режимах.

Лабораторная работа 5 посвящена исследованию процессов широтно-импульсной модуляции, моделированию и измерению основных параметров этого процесса.

Работы 6-9 посвящены изучению программного продукта ELCUT (Студенческая версия). Данный продукт активно используется для решения конечных полевых задач практически во всех областях промышленности, что его самое основное достоинство - это его бесплатный доступ для студентов. Как основа для изучения и моделирования процессов, этот продукт подходит идеально.

Лабораторная работа 6 посвящена изучению программного продукта ELCUT (Студенческая версия). В ходе работы студенты знакомятся с программой ELCUT и проводят моделирование двухмерных электрических полей и измерение их параметров различных технических конструкций.

Лабораторная работа 7 посвящена применению программного продукта ELCUT для решения задач по магнитной и электромагнитной дефектоскопии. В работе показаны возможности программного анализа параметров объектов с учетом нелинейности кривой перемагничивания и анизотропии магнитных свойств материала.

Лабораторная работа 8 посвящена проектированию промышленного оборудования с помощью программного продукта ELCUT. В работе показаны возможности программы ELCUT на примере моделирования установки индукционного нагрева с учетом мощности тепловыделения в нагреваемом элементе.

Лабораторная работа 9 демонстрирует, как с помощью средств программного продукта решать вопросы тепловой дефектоскопии, а именно, показана методика расчета теплового поля реального объекта - жилого помещения.

Работы 10-12 посвящены изучению программного продукта Turbo C++. Этот программный продукт является профессиональным инструментом разработчиков. Язык C++ широко используется для разработки программного обеспечения, являясь одним из самых популярных языков программирования, в частности, для томографов промышленного назначения.

Лабораторная работа 10 посвящена знакомству с особенностями программирования в интегрированной среде разработки Turbo C++.

Лабораторная работа 11 посвящена построению блок-схем вычислительных процессов. Понимание методики построения алгоритмов очень важно при программировании в любой среде.

Лабораторная работа 12 посвящена программированию циклических алгоритмов в C++, а именно рассмотрены этапы программирования: создание, запуск программы; поиск, анализ и устранение найденных ошибок программы.

После изучения программ и проработки заданий, представленных в данном учебном пособии, студент может успешно применять современные прикладные программные продукты для решения задач производства.

Лабораторная работа 1. Визуализация результатов проектной деятельности в Mathcad

В учебном пособии по курсу «Информационные технологии в приборостроении» подробно описаны типы графиков, которые используются для визуализации результатов исследовательской и проектной деятельности.

Mathcad позволяет обрабатывать следующие типы графиков:

--- *двумерные графики:*

- X-Y (декартовый) график (X-Y Plot);
- полярный график (Polar Plot).

--- *трехмерные графики:*

- график трехмерной поверхности (Surface Plot);
- график линий уровня (Contour Plot);
- трехмерная гистограмма (3D Bar Plot);
- трехмерное множество точек (3D Scatter Plot);
- векторное поле (Vector Field Plot).

Такое разделение графиков на типы условно, в Mathcad можно создавать новые типы графиков, а также комбинации графиков.

Для построения графиков рекомендуется использовать шаблоны. Они открываются **Вставка**→**Графики** (рис. 1). Для начального построения того или иного вида необходимо задать тип графика.



Рисунок 1. Типы графиков.

В подменю **Graph** содержится список из семи основных типов графиков:

<i>X-Y Plot</i>	График в декартовой системе координат;
<i>Polar Plot</i>	График в полярных координатах;
<i>Surface Plot</i>	Трехмерный график;
<i>Contour Plot</i>	Контурный график трехмерной поверхности;
<i>3D Scatter Plot</i>	График в виде точек (фигур) в трехмерном пространстве;
<i>3D Bar Chart</i>	График для изображения в виде совокупности столбиков в трехмерном пространстве (гистограмма);
<i>Vector Field Plot</i>	График векторного поля на плоскости.

Для того, чтобы изменить средства изображения графика, нужно щелкнуть дважды по полю графика и установить требуемые параметры в окнах настройки: изменение толщины и цвета линий, вид осей координат, координатные сетки, текстовые комментарии и др.

Для каждого типа графика есть свое диалоговое окно для выбора параметров графика. Для примера приведено диалоговое окно для графика в декартовой системе координат (рис. 2).

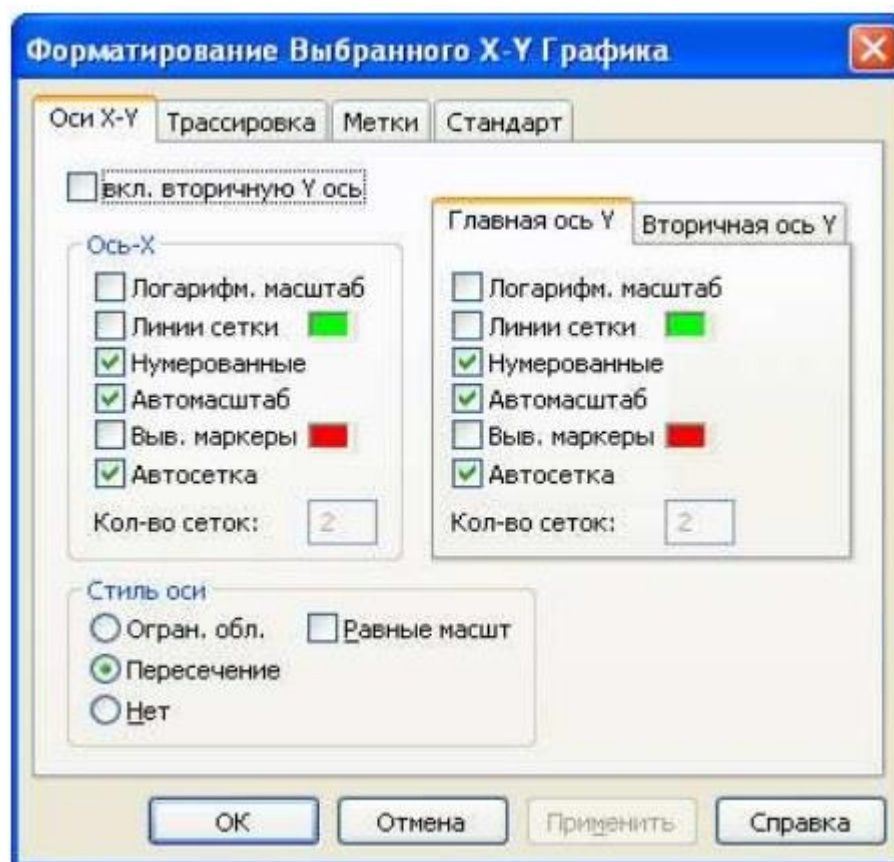


Рисунок 2. Настройка параметров графиков.

Задание: Получить графики различных функций согласно варианту (табл. 1): полностью их обработать: подписать оси, названия графиков и т.д.

1. Построение графика непрерывной функции.

Пусть вольтметр двойного интегрирования имеет погрешность, которая появилась из-за косинусоидальной помехи на входе. Необходимо построить график зависимости погрешности от периода сигнала помехи.

Погрешность определяется, как

$$\Delta = U_p \cdot \cos(\phi) \cdot \frac{\sin(\pi T_V / T)}{\pi T_V / T},$$

где Δ – погрешность; U_p – амплитуда напряжения помехи; ϕ – начальная фаза; T_V – продолжительность первого этапа интегрирования; T – период сигнала помехи. Значения расположены в диапазоне $10^{-4} \leq T \leq 2 \cdot 10^{-2}$.

Ниже показан алгоритм ввода данных для получения графика в программе Mathcad (рис. 3).

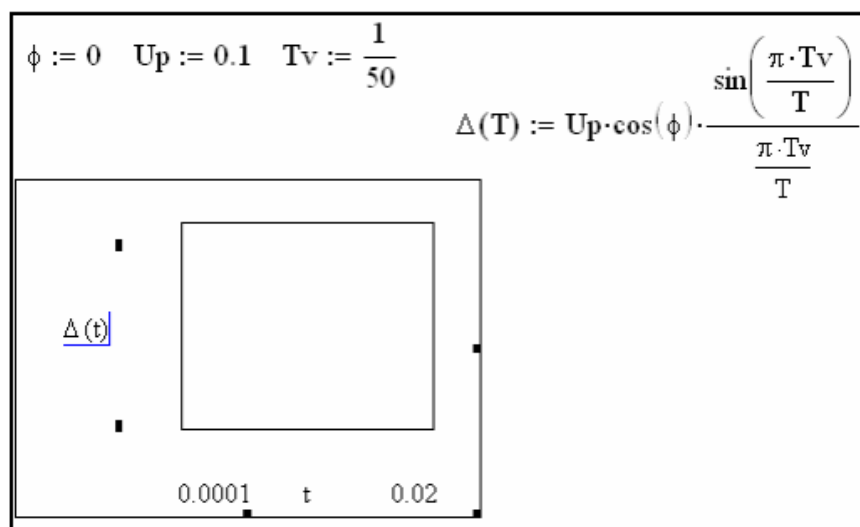


Рисунок 3. Построение графика функции.

Щелкаем мышью вне диапазона построения графика. При правильном введении параметров функции получаем требуемый график для исследования погрешности (рис. 4).

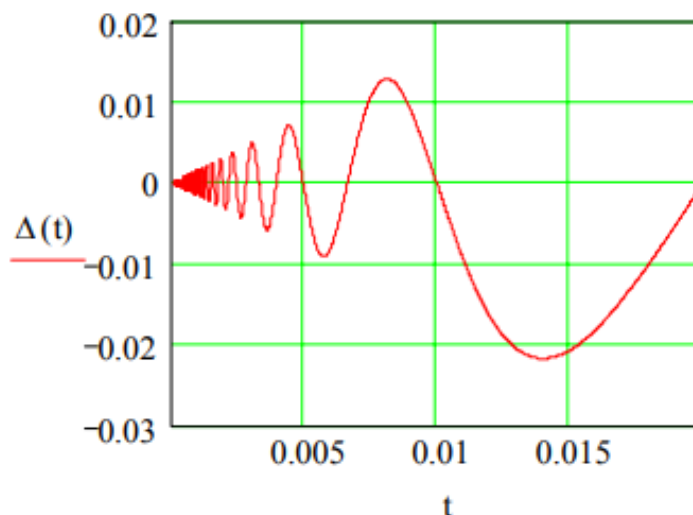


Рисунок 4. Полученный график функции.

2. Построение параметрически заданной кривой.

Необходимо построить график параметрически заданной функции. В данном случае построение происходит при задании определенных параметров. Пример приведен на рисунке 5.

$$x(t) := \frac{t^2}{1+t^2} \quad y(t) := \frac{t \cdot (1-t^2)}{1+t^2} \quad t := 0, 0.1..10$$

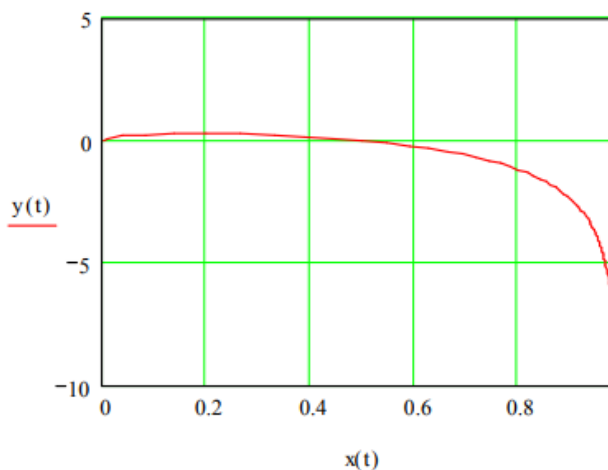


Рисунок 5. Полученный график функции.

3. Построение графиков поверхностей

Необходимо построить график поверхности заданной функции. График поверхности строится на основе данных двумерного числового массива. На графике показываются номера элементов массива.

В качестве примера построения поверхности возьмем данные задачи 1, задаем переменные в любых пределах. Последовательность введения данных и график полученной поверхности приведены на рисунке 6.

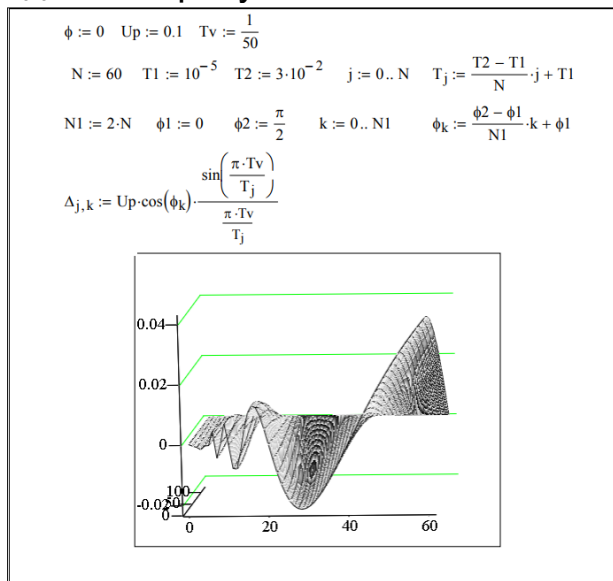


Рисунок 6. Полученный график поверхности.

Второй способ получения поверхностей малоизвестен, но очень удобен. В данном случае используется функция **CreateMesh**. При этом создается сетка поверхности: если прописать **CreateMesh (F)**, то будет построена поверхность функции F.

Построение поверхностей математическим способом и с помощью функции **CreateMesh** показано на рисунке 7, где i и j – целочисленные индексы.

1 способ

```

f(x,y) := sin(x^2 + y^2)
i := 0..20   j := 0..20
x_i := -1.5 + i*0.15
y_j := -1.5 + j*0.15
MT_{i,j} := f(x_i, y_j)

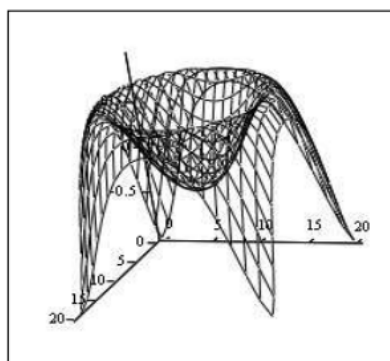
```

2 способ

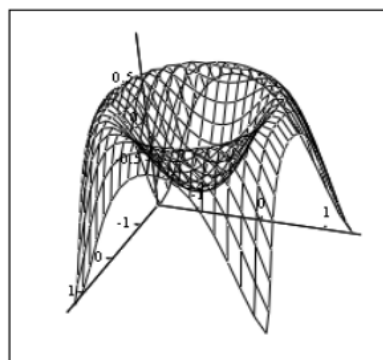
```

f(x,y) := sin(x^2 + y^2)
MM := CreateMesh (f, -1.5, 1.5, -1.5, 1.5, 20, 20)

```



MT



MM

Рисунок 7. Построение графика поверхности двумя способами.

Таблица 1. Варианты заданий

Вариант	Функция одной переменной	Параметрически заданная кривая	Поверхность
1	Значения расположены в диапазоне $3 \cdot 10^{-4} \leq T \leq 2 \cdot 10^{-2}$	$x = \frac{3t}{1+t^3}$ $y = \frac{3t^2}{1+t^3}$	$F(x, y) := -\sin(x^2 + y^2)$ $M_{x,y} := F\left(\frac{i-10}{10}, \frac{j-10}{10}\right)$
2	Значения расположены в диапазоне $5 \cdot 10^{-4} \leq T \leq 10^{-2}$	$x = t \cdot \exp(t)$ $y = t \cdot \exp(-t)$	$F(x, y) := \sin x + \cos x$ $M_{x,y} := F\left(\frac{i-10}{10}, \frac{j-10}{10}\right)$
3	Значения расположены в диапазоне $10^{-4} \leq T \leq 3 \cdot 10^{-2}$	$x = \frac{t}{t+3}$ $y = \frac{2t^2 + 3t}{(t+3)^2}$	$F(x, y) := \cos(x - y)$ $M_{x,y} := F\left(\frac{i-10}{5}, \frac{j-20}{5}\right)$
4	Значения расположены в диапазоне $10^{-4} \leq T \leq 5 \cdot 10^{-2}$	$x = 6 \cos(t) - 3 \cos(2t)$ $y = 6 \sin(t) - 3 \sin(2t)$	$F(x, y) := -\sin(x^2 + y^2)$ $M_{x,y} := F\left(\frac{i-5}{5}, \frac{j-10}{10}\right)$
5	Значения расположены в диапазоне $10^{-4} \leq T \leq 10^{-2}$	$x = t\sqrt{t^2 + 1}$ $y = \ln\left(\frac{1 + \sqrt{1 + t^2}}{t}\right)$	$F(x, y) := \sin x + \cos x$ $M_{x,y} := F\left(\frac{i-5}{5}, \frac{j-10}{10}\right)$
6	Значения расположены в диапазоне $10^{-4} \leq T \leq 10^{-2}$	$x = t + \exp(-t)$ $y = 2t + \exp(-2t)$	$F(x, y) := \cos(x - y)$ $M_{x,y} := F\left(\frac{i-5}{5}, \frac{j-10}{5}\right)$
7	Значения расположены в диапазоне $6 \cdot 10^{-4} \leq T \leq 10^{-2}$	$x = 2 \cos(t)$ $y = 3 \sin(t)$	$F(x, y) := -\sin(x^2 + y^2)$ $M_{x,y} := F\left(\frac{i-10}{10}, \frac{j-5}{5}\right)$
8	Значения расположены в диапазоне $10^{-4} \leq T \leq 6 \cdot 10^{-2}$	$x = 1 + \lg(2t + \sqrt{t^3})$ $y = 5(t^{2/3} - 1)$	$F(x, y) := \sin x + \cos x$ $M_{x,y} := F\left(\frac{i-10}{10}, \frac{j-5}{5}\right)$
9	Значения расположены в диапазоне $8 \cdot 10^{-4} \leq T \leq 8 \cdot 10^{-2}$	$x = 4 \cos^2(t)$ $y = 4 \sin^2(t)$	$F(x, y) := \cos(x - y)$ $M_{x,y} := F\left(\frac{i-10}{3}, \frac{j-3}{5}\right)$
10	Значения расположены в диапазоне $6 \cdot 10^{-4} \leq T \leq 6 \cdot 10^{-2}$	$x = 3t + 1$ $y = t^3 + 2t$	$F(x, y) := \cos(x \cdot y)$ $M_{x,y} := F\left(\frac{i-10}{3}, \frac{j-3}{5}\right)$

Содержание отчета

1. Цель работы.
2. Подробное описание методики построения графиков в программе согласно варианту.
3. Построение необходимых зависимостей.
4. Выводы.

Лабораторная работа 2. Исследование переходных процессов схем в Mathcad

В данной работе необходимо решить задачу получения и анализа импульсных переходных характеристик, и получения времени переходного процесса и перерегулирования для разных режимов нагрузки (нагруженный режим (10R), нормальный режим (100R), ненагруженный режим (1000R)) с помощью обратного преобразования Лапласа согласно варианту (табл. 2). В качестве объекта исследования выступает LC-фильтр низких частот (рис. 8), со следующими заданными параметрами: $L = 10$ мГн, $C = 100$ мкФ; $R = 1$ Ом.

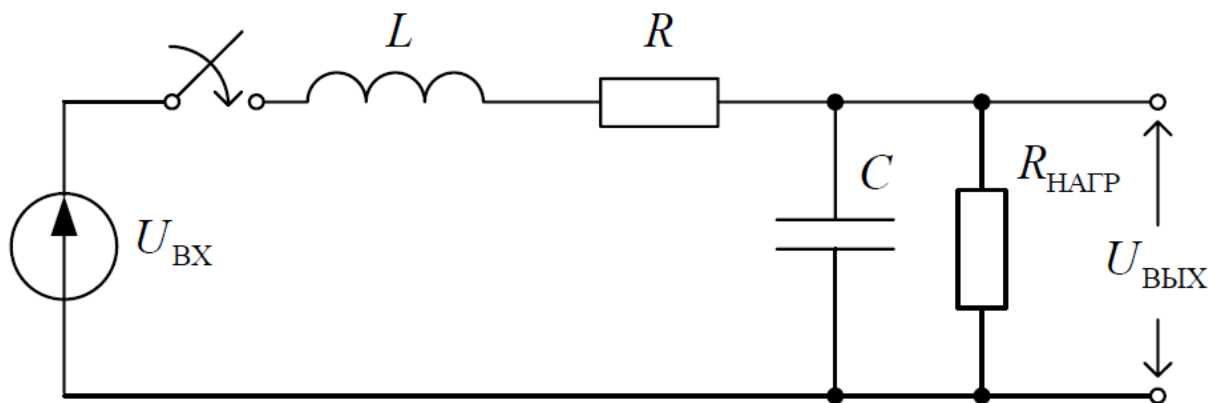


Рисунок 8. Исследуемая схема.

Для начала необходимо проанализировать схему для составления системы дифференциальных уравнений состояния схемы. Как видим, на вход схемы подается гармонический сигнал входного напряжения.

Применяя закон Кирхгофа можно записать выражение для входного напряжения схемы следующим образом (1):

$$U_{ВХ}(t) = L \frac{di_L(t)}{dt} + i_L(t) \cdot R + U_{ВЫХ}(t), \quad (1)$$

где $i_L(t) = i_C(t) + i_H(t)$ – ток катушки, $i_C(t) = C \frac{dU_{ВЫХ}(t)}{dt}$ – ток конденсатора, $i_H(t) = \frac{U_{ВЫХ}(t)}{R_{НАГР}}$ – ток нагрузки.

Тогда система уравнений состояния схемы будет выглядеть, как (2):

$$\begin{cases} U_{BX} = L \cdot \frac{di_L(t)}{dt} + i_L(t) \cdot R + U_{BbIX}(t) \\ i_L(t) = C \cdot \frac{dU_{BbIX}(t)}{dt} + \frac{U_{BbIX}(t)}{R_{HAГP}} \end{cases} \quad (2)$$

После преобразования Лапласа система будет иметь вид (3):

$$\begin{cases} U_{BX}(s) = (L \cdot s + R) \cdot i_L(s) + U_{BbIX}(s) \\ i_L(s) = \left(C \cdot s + \frac{1}{R_{HAГP}} \right) \cdot U_{BbIX}(s) \end{cases} \quad (3)$$

где s – оператор Лапласа.

Подставим нижнее уравнение из (3) в верхнее:

$$U_{BX}(s) = (L \cdot s + R) \left(C \cdot s + \frac{1}{R_{HAГP}} \right) \cdot U_{BbIX}(s) + U_{BbIX}(s), \quad (4)$$

Преобразуем (4х):

$$U_{BX}(s) = \left((L \cdot s + R) \left(C \cdot s + \frac{1}{R_{HAГP}} \right) + 1 \right) \cdot U_{BbIX}(s).$$

Как известно, операторной функцией являемся отношение изображения выходного сигнала $U_{BbIX}(p)$ к входному $U_{BX}(p)$, тогда передаточная функция схемы $W(p) = U_{BbIX}(p) / U_{BX}(p)$.

Тогда передаточная функция схемы в операторной форме выглядит следующим образом (5):

$$W(s) = \frac{1}{(L \cdot s + R) \left(C \cdot s + \frac{1}{R_{HAГP}} \right) + 1}, \quad (5)$$

Присваиваем в Mathcad заданные параметры схемы:

$L := 10 \cdot 10^{-3}$	$C := 100 \cdot 10^{-6}$	$R := 1$	
$U_{BX} := 1$	$R_{HAГP1} := 1000$	$R_{HAГP2} := 100$	$R_{HAГP3} := 10$

С учетом (5) выходной напряжение схемы выглядит так:

$$U_{BbIX}(s) = \frac{U_{BX}}{s} \cdot \frac{1}{(L \cdot s + R) \left(C \cdot s + \frac{1}{R_{HAГP}} \right) + 1}, \quad (6)$$

где $\frac{U_{BX}}{s}$ – операторная форма входного ступенчатого напряжения при коммутации ключа.

Используем встроенную функцию Mathcad для упрощения выражения (6) для $R_{HAГP1}=1000R$:

$$\frac{U_{\text{ВХ}}}{s} \cdot \frac{1}{1 + (L \cdot s + R) \cdot \left(C \cdot s + \frac{1}{R_{\text{нагр1}}} \right)} \text{ simplify } \rightarrow \frac{1000000}{\left[s \cdot (1001000 + s^2 + 110 \cdot s) \right]}$$

Применим преобразование Лапласа к полученному выражению (с учетом нулевых условий-до коммутации токи и напряжения схемы равны 0).

$$\frac{1000000}{\left[s \cdot (1001000 + s^2 + 110 \cdot s) \right]} \text{ invlaplace , s } \rightarrow$$

Получаем следующее выражение для переходной функции при ненагруженном режиме (7):

$$h_1(t) = \frac{1000}{1001} - \frac{1000}{1001} \cdot \exp(-55 \cdot t) \cdot \cos(5 \cdot \sqrt{39919} \cdot t) - \frac{1000}{3632629} \cdot \exp(-55 \cdot t) \cdot \sqrt{39919} \cdot \sin(5 \cdot \sqrt{39919} \cdot t) \cdot t \quad (7)$$

Аналогично получаем переходные функции для значений $h_2(t)$ и $h_3(t)$, а также строим временные зависимости в Mathcad (рис. 9).

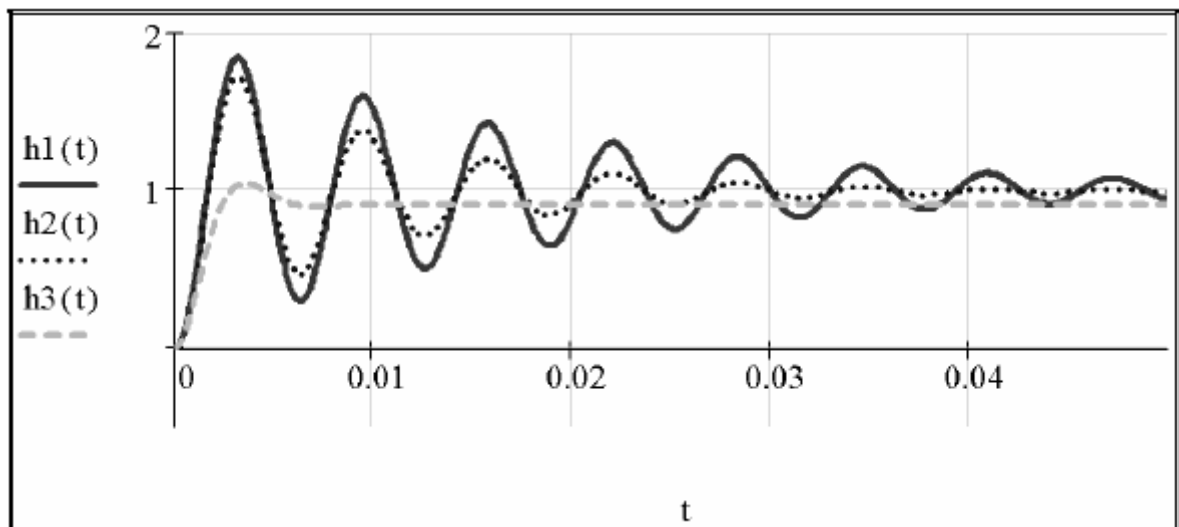


Рисунок 9. Переходные характеристики схемы при различных режимах работы.

Как видно, график переходного процесса имеет колебательный характер с перерегулированием. Для установившегося режима переходного процесса характерно, что переходный процесс заканчивается при $h_1(t) \leq 1.05h_1(\infty)$ и $h_1(t) \geq 0.95h_1(\infty)$, то есть когда функция в последний раз входит в зону допустимых отклонений $(h_1(\infty) \pm 0.05h_1(\infty))$ при $t \rightarrow \infty$ (при 100% отработки сигнала) (рис. 10).

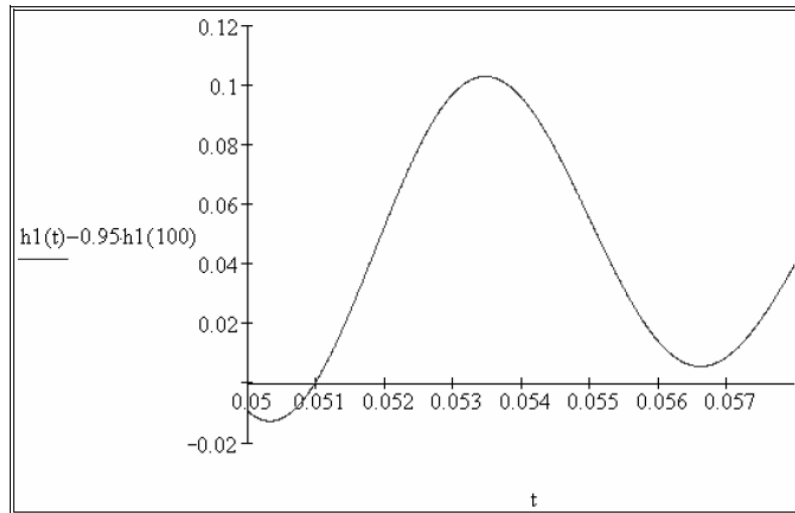


Рисунок 10. Переходные характеристики схемы при различных режимах работы.

Согласно рисунку 10, функция $h_1(t)$ в последний раз пересекает нижнюю границу области допустимых значений при 0.051 с.

Для определения времени переходного процесса необходимо построить график зависимости $(h_1(t) - 1.05h_1(t))$ от времени (рис. 11). Переходный процесс заканчивается, когда функция в последний раз входит в зону допустимых отклонений $(h_1(\infty) \pm 0.05h_1(\infty))$. Как видно, функция последний раз пересекает верхнюю границу допустимых значений при 0,054 с; это есть время переходного процесса $t_{пп1}$.

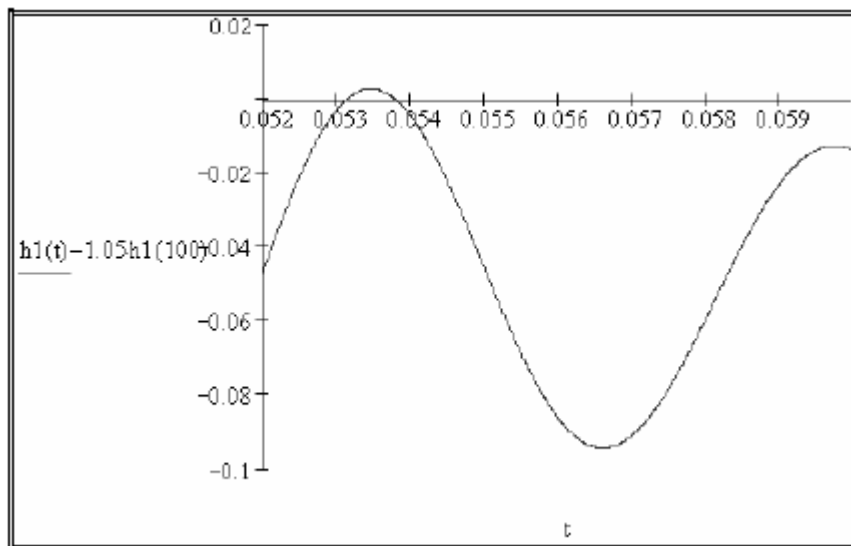


Рисунок 11. Время переходного процесса.

Аналогично определяем значения времени переходного процесса для функций $h_2(t)$ и $h_3(t)$.

Перерегулирование для переходной характеристики $h_1(t)$ определяется, согласно (12):

$$\Delta h_1 = \frac{h_1(t_{m1}) - h_1(\infty)}{h_1(\infty)} \cdot 100\% \approx \frac{h_1(t_{m1}) - h_1(100)}{h_1(100)} \cdot 100\%, \quad (12)$$

где t_{m1} – время достижения первого максимума функцией $h_1(t)$.

В программе Mathcad определим производную от $h_1(t)$, чтобы определить t_{m1} :

$$\frac{d}{dt} h_1(t) \rightarrow \frac{200000}{39919} \cdot \exp(-55 \cdot t) \cdot \sqrt{39919} \cdot \sin(5 \cdot \sqrt{39919} \cdot t)$$

Присвоим функцию $dh_1(t)$, как показано ниже, и построим ее график (рис. 12).

$$dh_1(t) := \frac{200000}{39919} \cdot \exp(-55 \cdot t) \cdot \sqrt{39919} \cdot \sin(5 \cdot \sqrt{39919} \cdot t)$$

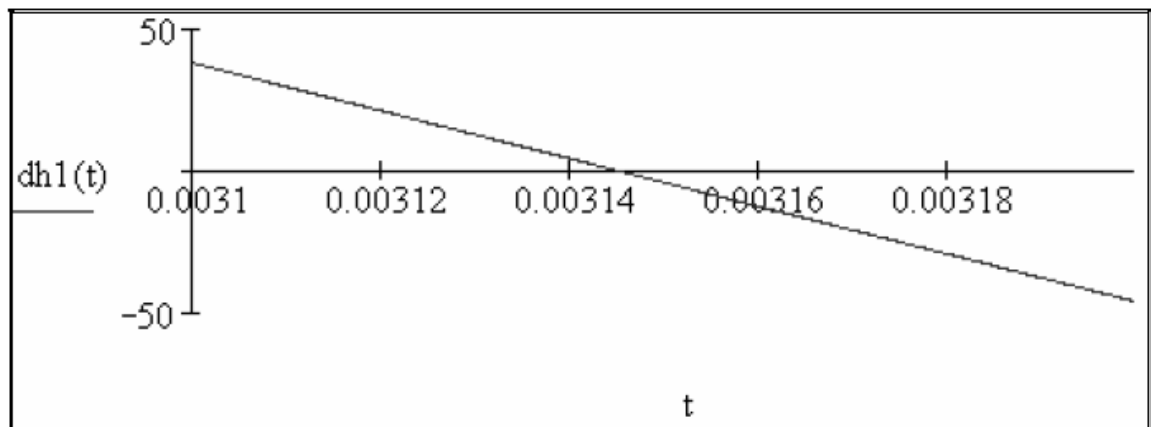


Рисунок 12. Функция $dh_1(t)$.

Графически время $t_{t1}=0.00314$ с, это соответствует t_{m1} в первом приближении.

Далее с помощью оператора **Find** прописываем процедуру определения значения перерегулирования переходной функции (рис. 13).

Как видим, значение перерегулирования переходной функции $h_1(t)$ составляет $\Delta h_1 = 84, 117\%$.

Аналогично определяем значения перерегулирования для переходных функций Δh_2 и Δh_3 .

Полученные результаты внесены в таблицу 2. Результаты показывают, что при увеличении загруженности фильтра время переходного процесса и перерегулирование уменьшаются.

$$\begin{aligned}
&tt1 := 0.00314 \\
&\text{Given} \\
&dh1(tt1) = 0 \\
&tm1 := \text{Find}(tt1) \\
&tm1 = 3.145 \times 10^{-3} \\
&\frac{h1(tm1) - h1(100)}{h1(100)} \cdot 100 = 84.117
\end{aligned}$$

Рисунок 13. Определение значения перерегулирования переходной функции.

Таблица 3. Таблица результатов работы

Показатель	Значение нагрузки		
	$R_{\text{НАГР1}}=1000R$	$R_{\text{НАГР2}}=100R$	$R_{\text{НАГР3}}=10R$
Время переходного процесса	$t_{\text{ПП1}}=0,054$ с	$t_{\text{ПП2}}=0,029$ с	$t_{\text{ПП3}}=0,005$ с
Перерегулирование	$\Delta h_1=84,1\%$	$\Delta h_2=73\%$	$\Delta h_3=13\%$

Таблица 2. Варианты заданий к рисунку 8.

Вариант	Параметры схемы
1	$L=10$ мГн, $C=100$ мкФ; $R=1$ Ом.
2	$L=20$ мГн, $C=200$ мкФ; $R=10$ Ом.
3	$L=30$ мГн, $C=300$ мкФ; $R=20$ Ом.
4	$L=40$ мГн, $C=100$ мкФ; $R=30$ Ом.
5	$L=50$ мГн, $C=200$ мкФ; $R=1$ Ом.
6	$L=15$ мГн, $C=300$ мкФ; $R=10$ Ом.
7	$L=25$ мГн, $C=100$ мкФ; $R=20$ Ом.
8	$L=35$ мГн, $C=200$ мкФ; $R=30$ Ом.
9	$L=45$ мГн, $C=300$ мкФ; $R=1$ Ом.
10	$L=50$ мГн, $C=100$ мкФ; $R=10$ Ом.

Содержание отчета

1. Цель работы.
2. Подробное описание методики построения графиков переходных процессов.
3. Обработка графиков переходных процессов и получение необходимых характеристик для исследуемых режимов.
4. Сведение результатов в таблицу 4, выводы.

Таблица 4. Таблица результатов работы

Показатель	Значение нагрузки		
	$R_{\text{НАГР1}}=1000R$	$R_{\text{НАГР2}}=100R$	$R_{\text{НАГР3}}=10R$
Время переходного процесса	$t_{\text{П1}}=$	$t_{\text{П2}}=$	$t_{\text{П3}}=$
Перерегулирование	$\Delta h_1=$	$\Delta h_2=$	$\Delta h_3=$

Лабораторная работа 3. Проектирование простейших моделей в программе MatLab

Задание: ознакомиться с основами работы в программе MatLab на примере моделирования простейшей схемы (рис. 14) согласно варианту (табл. 5).

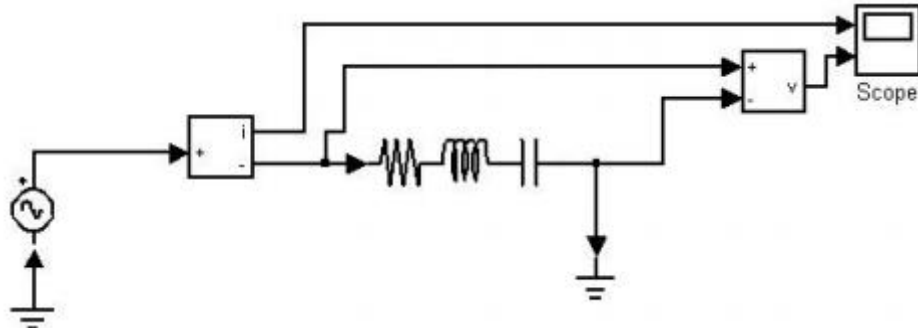


Рисунок 14. Простая электрическая цепь постоянного тока.

Пусть источник напряжения генерирует переменное напряжение величиной $U = 10$ В, частотой 50 Гц, фаза = 0, параметры RLC-цепочки: $R = 0,1$ Ом, $L = 0,001$ Гн, $C = 0,15$ Ф, время моделирования 1 с, метод расчета *ode23s*, максимальный шаг интегрирования 0,0001.

После запуска программы MatLab создаем новую модель путем выбора следующих вкладок в среде **New** \Rightarrow **Simulink Model** (рис. 15).

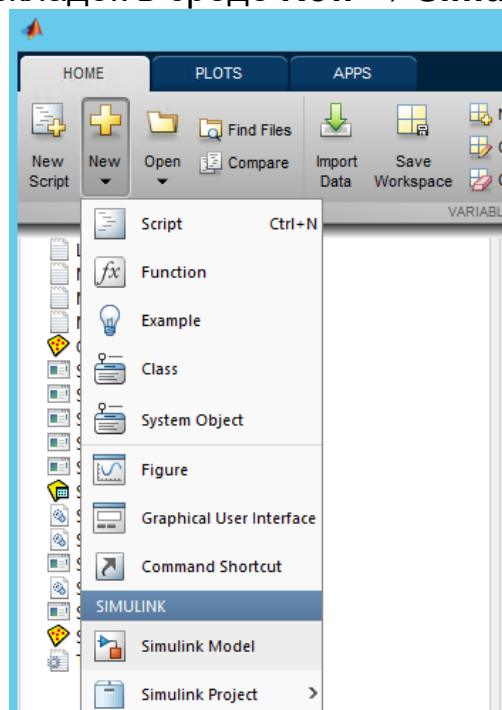


Рисунок 15. Создание новой модели Simulink Model в среде MatLab.

Далее выбираем в открывшемся окне **Library Browser** (рис. 16).

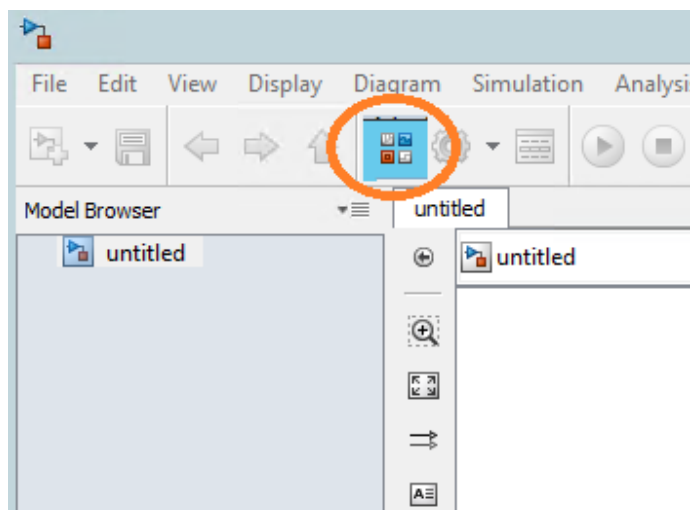


Рисунок 16. Создание новой модели Simulink Model в среде MatLab.

Тем самым мы активизировали библиотеку компонентов Simulink (рис. 17).

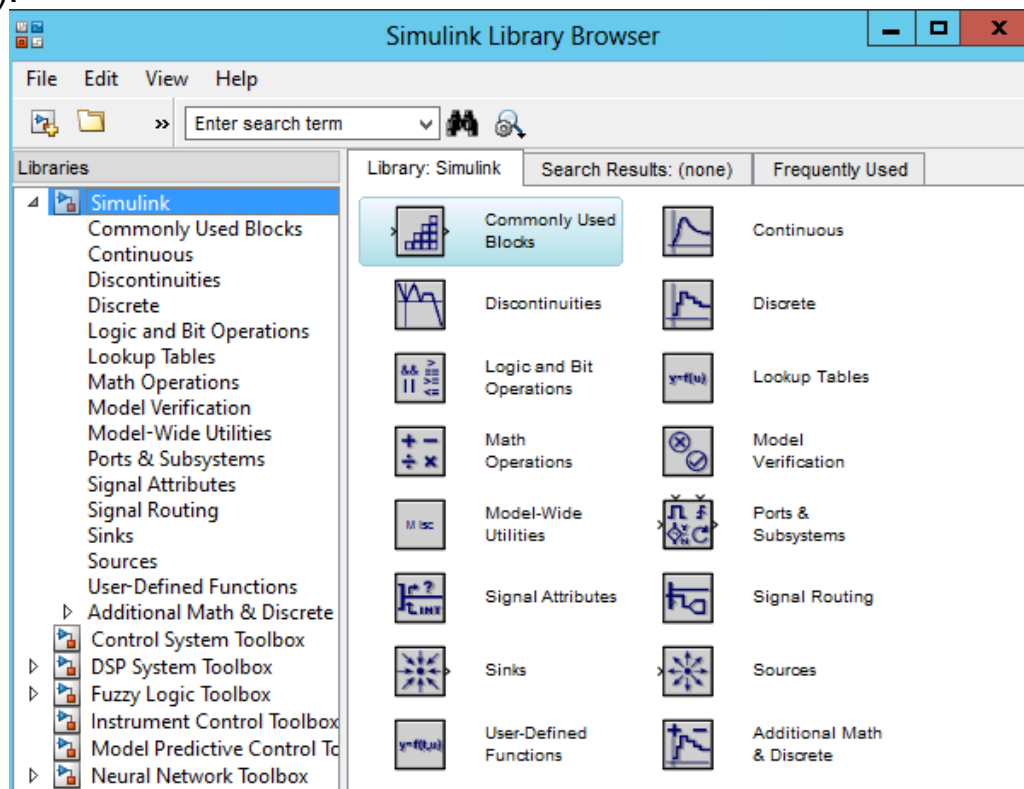


Рисунок 17. Библиотека компонентов Simulink.

Библиотека компонентов содержит элементы (блоки), необходимые для создания моделей.

Чтобы найти в библиотеке элементы, нужно в строке поиска набрать название искомого элемента и нажать **Enter**. После успешного поиска элемент отражается в списке справа (рис. 18). Его необходимо переместить на рабочее поле программы, где происходит сборка элементов схемы.

Другой способ – это последовательно открывать древо библиотек. К примеру, чтобы найти **Parallel RLC Branch** надо выполнить последовательность **SimScape/ SimPowerSystems / Elements / Parallel RLC Branch**.

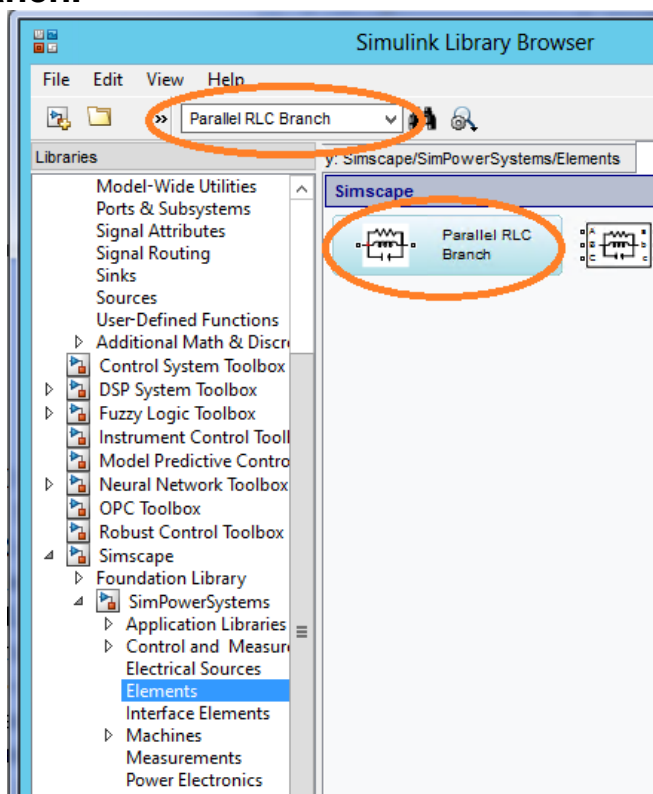


Рисунок 18. Поиск элемента схемы в библиотеке.

К элементу применяются привычные функции редактирования:

- Font** – формат текста названия блока;
- Hide Name (Show Name)** – убрать название (показать название);
- Flip Name** – переместить название на противоположную сторону блока;
- Flip Block** – зеркально отразить блок;
- Rotate Block** – повернуть блок по часовой стрелке (Clockwise) или против часовой стрелки (Counterclockwise);
- Show Drop Shadow (Hide Drop Shadow)** – показать тень (убрать тень).

Когда все элементы расположены на рабочем поле, необходимо их соединить. Соединение происходит протяжкой проводника от одного элемента к другому.

Также для соединения можно выделить первый элемент и нажать клавишу **Ctrl**. Не отпуская этой клавиши, нажать на второй элемент. При этом будет проведена линия связи между выходом первого элемента и входом второго элемента.

После сборки схемы рекомендуется ее сохранить: **File**→**Save As**.

Перед запуском программы необходимо задать параметры расчета исследуемого процесса **Simulation**→**Model Configuration Parameters** (рис. 19).

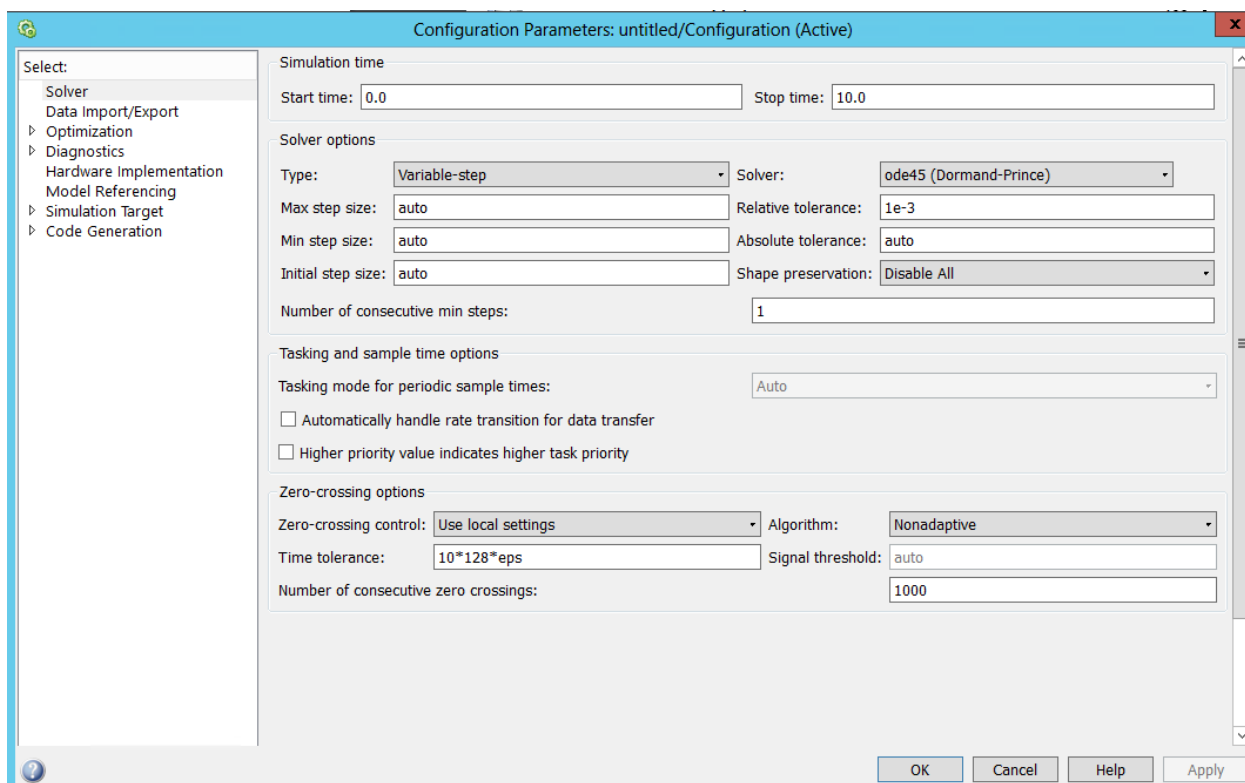


Рисунок 19. Настройка параметров расчета модели в Simulink.

В левой части окна находится список групп настраиваемых параметров.

Ввод параметров моделирования выполняется с помощью панели **Solver**. Элементы этой панели разделены на две группы: **Simulation time** (интервал времени моделирования) и **Solver options** (параметры расчёта).

Интервал времени моделирования задаётся путём указания начального (**Start time**) и конечного (**Stop time**) значений времени расчёта. Как правило, начальное время указывается **0**, а конечное время зависит от условий поставленной задачи.

В параметрах расчёта необходимо выбрать метод интегрирования **Type**: с фиксированным шагом (**Fixed-step**) или с переменным шагом (**Variable-step**). Максимальный шаг интегрирования задаётся в графе **Max Step Size**.

Также в параметрах расчёта необходимо указать один из следующих методов:

ode45 – одношаговые явные методы Рунге-Кутты 4-го и 5-го порядка;

ode23 – одношаговые явные методы Рунге-Кутты 2-го и 3-го порядка;

ode113 – многошаговый метод Адамса-Башворта-Мултона переменного порядка;

ode15s – многошаговый метод переменного порядка (от 1 до 5), использующий формулы численного дифференцирования;

ode23s – одношаговый метод, использующий модифицированную формулу Розенброка 2-го порядка;

ode23t – метод трапеций с интерполяцией;

ode23tb – неявный метод Рунге-Кутты в начале решения и метод, использующий формулы обратного дифференцирования 2-го порядка в последующем.

После редактирования параметров следует нажать клавишу **OK** или **Apply**.

Запуск расчёта (моделирования) выполняется нажатием кнопки **Start** на панели инструментов.

За графическое представление результатов моделирования в программе отвечает прибор **Scope** – осциллограф, который отражает временные зависимости.

Для просмотра графика необходимо дважды щелкнуть левой кнопкой мыши по прибору. Программа позволяет задавать необходимое количество входов осциллографа. Для этого необходимо в параметрах осциллографа указать количество входов (**Number of axes**).

Результаты моделирования схемы представлены на рисунке 20.

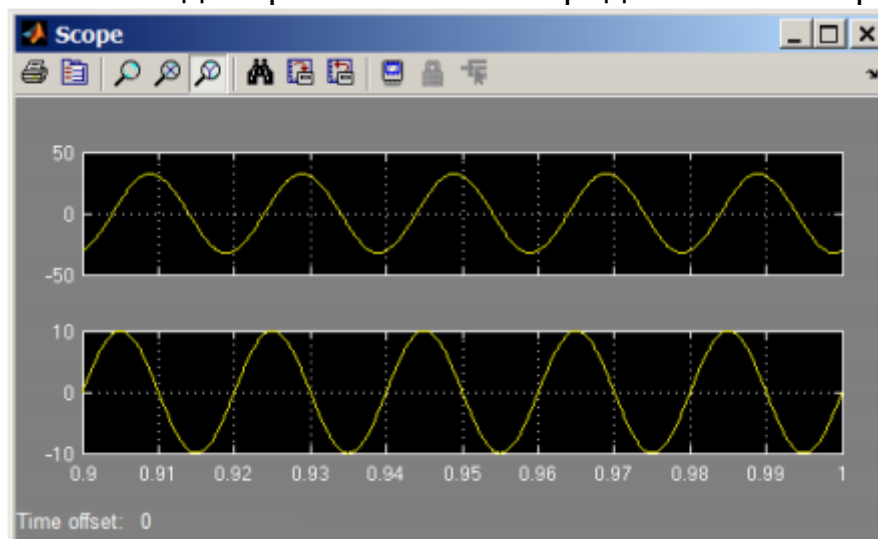


Рисунок 20. Результаты расчета модели в Simulink.

Таблица 5. Варианты заданий к рисунку 14.

Вариант	Параметры исследуемой схемы
1	Параметры RLC-цепочки: $R = 0,1$ Ом, $L = 0,005$ Гн, $C = 0,25$ Ф, время моделирования 1 с, метод расчета <i>ode23s</i> , максимальный шаг интегрирования 0,0001.
2	Параметры RLC-цепочки: $R = 0,3$ Ом, $L = 0,01$ Гн, $C = 0,05$ Ф, время моделирования 1 с, метод расчета <i>ode23s</i> , максимальный шаг интегрирования 0,0001.
3	Параметры RLC-цепочки: $R = 0,5$ Ом, $L = 0,05$ Гн, $C = 0,15$ Ф, время моделирования 1 с, метод расчета <i>ode23s</i> , максимальный шаг интегрирования 0,0001.
4	Параметры RLC-цепочки: $R = 0,1$ Ом, $L = 0,001$ Гн, $C = 0,15$ Ф, время моделирования 0,9 с, метод расчета <i>ode23s</i> , максимальный шаг интегрирования 0,0001.
5	Параметры RLC-цепочки: $R = 0,3$ Ом, $L = 0,002$ Гн, $C = 0,25$ Ф, время моделирования 0,9 с, метод расчета <i>ode23s</i> , максимальный шаг интегрирования 0,0001.
6	Параметры RLC-цепочки: $R = 0,5$ Ом, $L = 0,004$ Гн, $C = 0,1$ Ф, время моделирования 0,9 с, метод расчета <i>ode23s</i> , максимальный шаг интегрирования 0,0001.
7	Параметры RLC-цепочки: $R = 0,2$ Ом, $L = 0,006$ Гн, $C = 0,5$ Ф, время моделирования 1 с, метод расчета <i>ode23s</i> , максимальный шаг интегрирования 0,0001.
8	Параметры RLC-цепочки: $R = 0,4$ Ом, $L = 0,01$ Гн, $C = 0,25$ Ф, время моделирования 1 с, метод расчета <i>ode23s</i> , максимальный шаг интегрирования 0,0001.
9	Параметры RLC-цепочки: $R = 0,6$ Ом, $L = 0,05$ Гн, $C = 0,3$ Ф, время моделирования 1 с, метод расчета <i>ode23s</i> , максимальный шаг интегрирования 0,0001.
10	Параметры RLC-цепочки: $R = 0,8$ Ом, $L = 0,02$ Гн, $C = 0,35$ Ф, время моделирования 1 с, метод расчета <i>ode23s</i> , максимальный шаг интегрирования 0,0001.

Содержание отчета

1. Цель работы.
2. Подробное описание методики построения модели исследуемого устройства в программе.
3. Исследование объекта, построение необходимых зависимостей, анализ результатов.
4. Выводы.

Лабораторная работа 4. Виртуальное проектирование технических устройств в программе MatLab

Задание 1. Изучить работу заданной принципиальной схемы устройства на основании анализа ее переходных процессов в среде MatLab согласно варианту (см. табл. 6).

Рассмотрим создание имитационной модели на примере фильтра низких частот со следующими параметрами $L_1= 50$ мГн; $R_1= 1$ Ом; $C_1= 10$ мкФ; $L_2= 5$ мГн; $R_2= 100$ Ом (рис. 21).

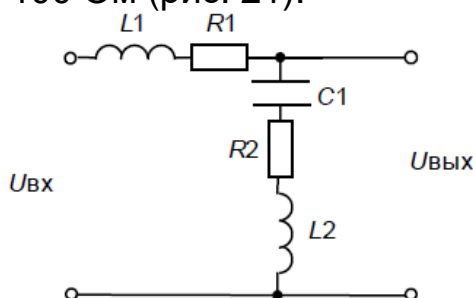


Рисунок 21. Схема исследуемого фильтра.

После запуска программы MatLab создаем модель исследуемой схемы.

Для создания подобных схем в среде MatLab предусмотрена последовательная и параллельная $R-L-C$ цепочки (рис. 22).



Рисунок 22. Последовательная и параллельная $R-L-C$ цепочки.

В программе существует возможность отключения ненужного элемента $R-L-C$ цепочки без какого-либо ущерба для работы схемы.

В **последовательной** $R-L-C$ цепочке для исключения **емкости** в окне задания параметров цепочки устанавливаем **inf** (от слова бесконечность), для исключения **сопротивления и индуктивности** в окне задания параметров цепочки устанавливаем **0** (рис. 23). В **параллельной** $R-L-C$ цепочке для исключения **емкости** в окне задания параметров цепочки устанавливаем **0**, для исключения **сопротивления и индуктивности** в окне задания параметров цепочки устанавливаем **inf**.

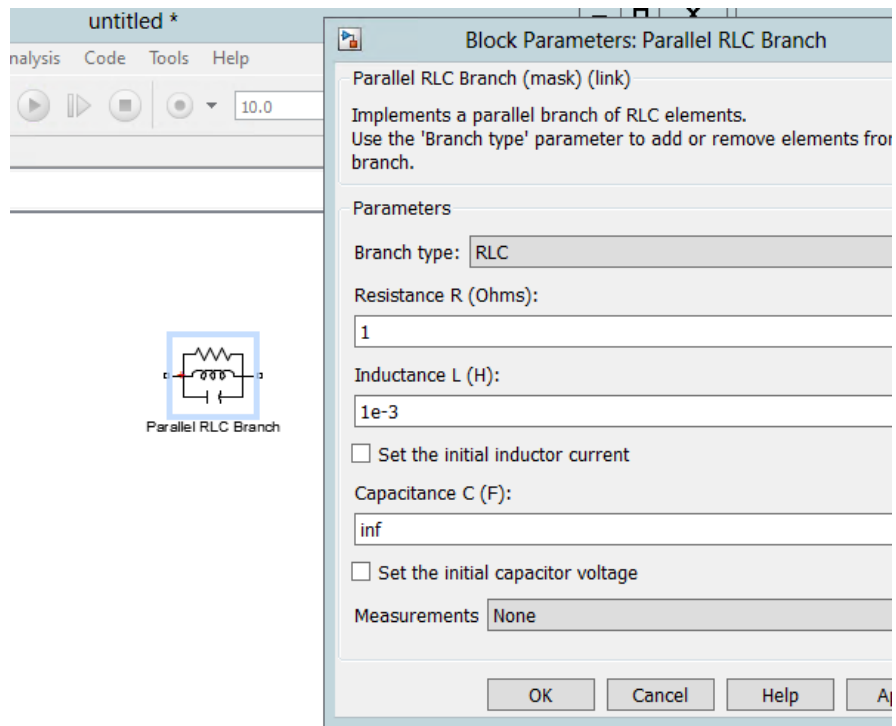


Рисунок 23. Окно ввода параметров R - L - C цепочки.

В нашем случае нужно использовать последовательную R - L - C цепочку.

На вход схемы подаем переменное напряжение. В схеме за это отвечает источник переменного напряжения (**AC Voltage Source**). Собираем схему, соединяя проводниками элементы, вводим заданные параметры (рис. 24).

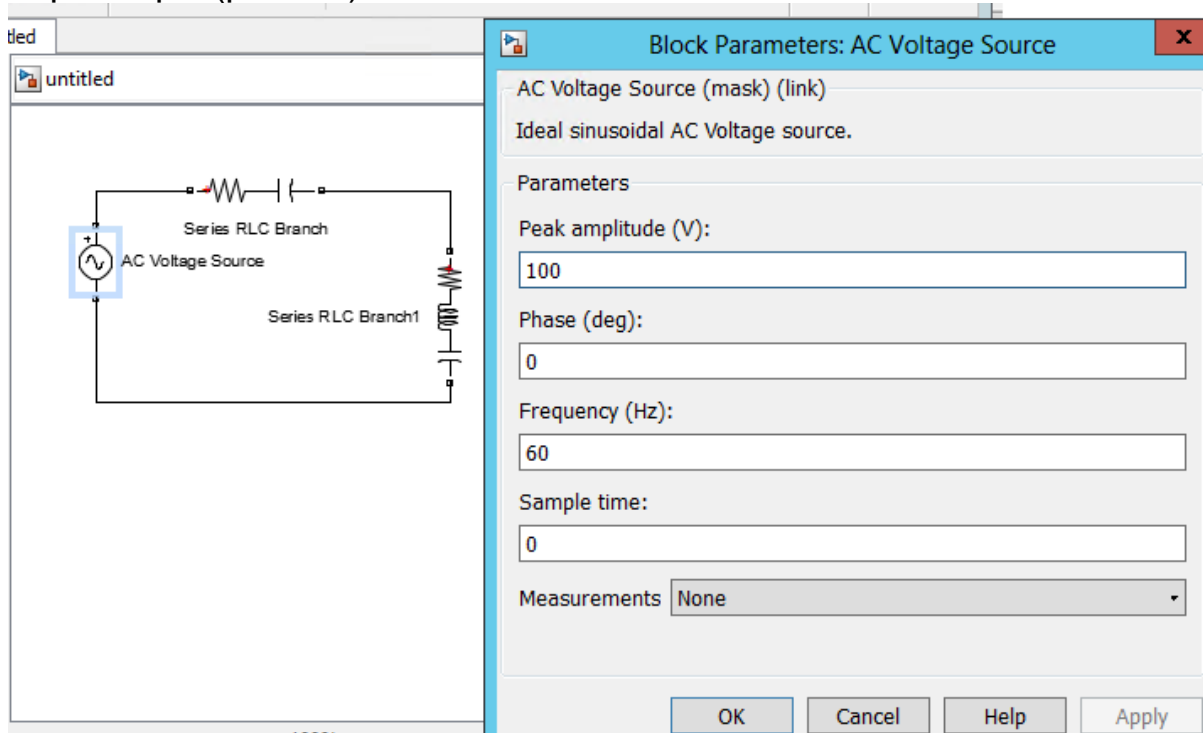


Рисунок 24. Исследуемая схема.

Для получения переходных процессов схемы необходимо подключить измерительный прибор для снятия напряжения – Вольтметр (**Voltage Measurement**), и далее - Осциллограф (**Scope**). Обратите внимание, что осциллограф в данной программе имеет возможность настройки количества входов. В нашем случае в настройках мы выбрали 2 входа.

Исследуем схему в двух режимах: без нагрузки и с нагрузкой (рис. 25).

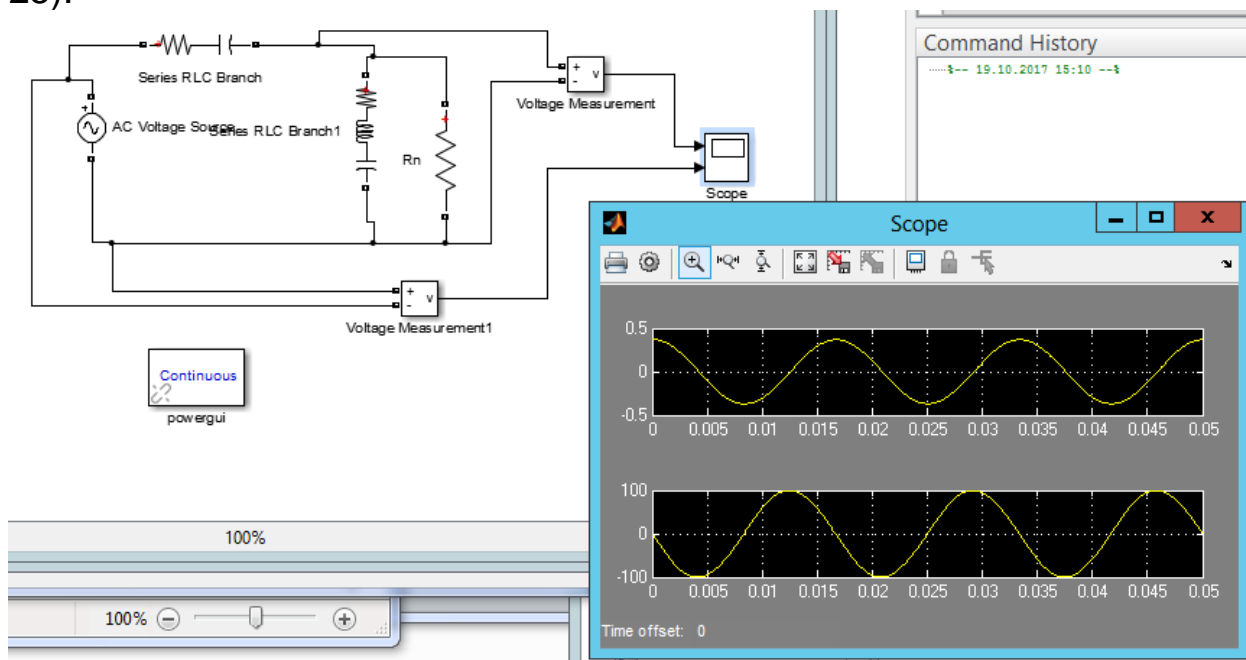


Рисунок 25. Характеристики исследуемой схемы с нагрузкой.

Как вы видите, на схеме еще должен присутствовать еще один дополнительный блок – **powergui**.

Этот блок является инструментом графического интерфейса пользователя и обеспечивает решение следующих задач:

- расчет схемы комплексным методом,
- расчет установившегося режима,
- дискретизация модели,
- задание начальных условий,
- инициализация трехфазных схем содержащих электрические машины, таким образом, чтобы расчет начался с установившегося режима,
- анализ схемы с помощью инструмента Simulink LTI-Viewer,
- определение полного сопротивления (импеданса) цепи,
- выполнение гармонического анализа,
- создание отчета,
- создание файла характеристик намагничивания для модели нелинейного трансформатора.

Переходим к получению графиков переходных процессов исследуемой схемы.

Для этого необходимо описать переходные процессы, происходящие в схеме с помощью дифференциальных уравнений по второму закону Кирхгофа.

Для исследуемой схемы система дифференциальных уравнений будет выглядеть так (13):

$$\begin{cases} U_{BX} = L_1 \cdot \frac{di(t)}{dt} + i(t) \cdot R_1 + U_{ВЫХ}(t) \\ U_{ВЫХ} = L_2 \cdot \frac{di(t)}{dt} + i(t) \cdot R_2 + \frac{1}{C_1} \cdot \int i(t) dt \end{cases} \quad (13)$$

Продифференцируем второе уравнение из (13), получим (14)

$$\begin{cases} U_{BX} = L_1 \cdot \frac{di(t)}{dt} + i(t) \cdot R_1 + U_{ВЫХ}(t) \\ U_{ВЫХ} = L_2 \cdot \frac{d^2 i(t)}{dt^2} + \frac{di(t)}{dt} \cdot R_2 + \frac{1}{C_1} \cdot i(t) \end{cases} \quad (14)$$

Переходим от временной к операторной форме записи системы уравнений для получения передаточной функции (отношение изображения по Лапласу выходной координаты к изображению входной). Такое преобразование называется преобразованием Лапласа, при этом происходит замена в уравнениях $p \Rightarrow \frac{d}{dt}$.

Система (14) после преобразований Лапласа будет выглядеть так (15):

$$\begin{cases} U_{BX}(p) = L_1 \cdot p \cdot i(p) + i(p) \cdot R_1 + U_{ВЫХ}(p) \\ p \cdot U_{ВЫХ}(p) = L_2 \cdot p^2 \cdot i(p) + i(p) \cdot R_2 \cdot p + \frac{1}{C_1} \cdot i(p) \end{cases} \quad (15)$$

В (3) выразим $i(p)$ во втором уравнении, подставим в первое уравнение.

После преобразований получим передаточную функцию заданной схемы (16):

$$W(p) = \frac{1 + p \cdot R_2 \cdot C_1 + L_2 \cdot C_1 \cdot p^2}{L_1 \cdot C_1 \cdot p^2 + p \cdot R_1 \cdot C_1 + 1 + p \cdot R_2 \cdot C_1 + L_2 \cdot C_1 \cdot p^2} \quad (16)$$

или

$$W(p) = \frac{1 + p \cdot R_2 \cdot C_1 + L_2 \cdot C_1 \cdot p^2}{p^2 (L_1 \cdot C_1 + L_2 \cdot C_1) + p (R_1 \cdot C_1 + R_2 \cdot C_1) + 1} \quad (17)$$

Выразим (17) через постоянные времени электрической цепи: $T_1^2 = L_2 \cdot C_1$, $T_2 = R_2 \cdot C_1$, $T_3^2 = L_1 \cdot C_1$, $T_4 = R_1 \cdot C_1$. Получим (18):

$$W(p) = \frac{T_1^2 \cdot p^2 + T_2 \cdot p + 1}{(T_3^2 + T_1^2) \cdot p^2 + (T_4 + T_2) \cdot p + 1} \quad (18)$$

Как известно, схемы фильтров собираются по Т и П-образной схемам. Их общий вид представлен на рисунке 26.

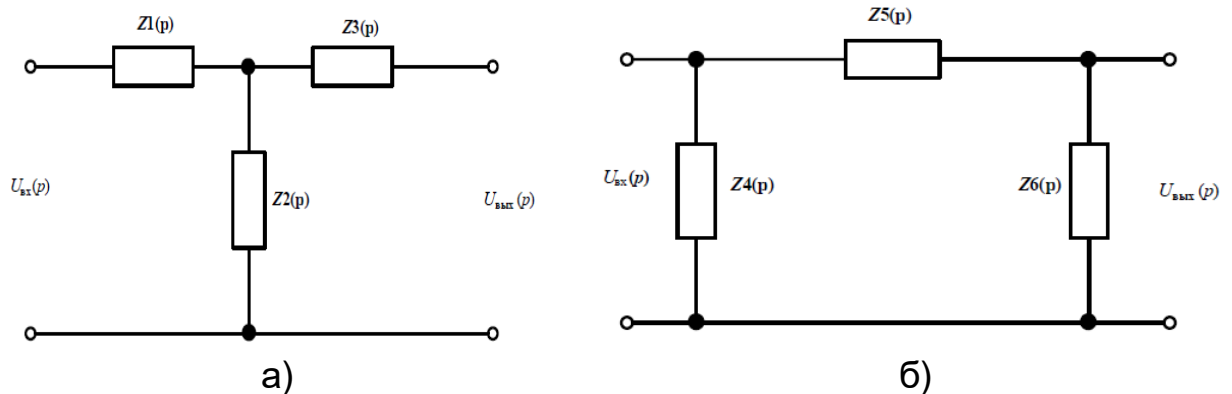


Рисунок 26. Т (а) и П-образная (б) схемы фильтров.

Для дальнейшей работы необходимо определить к какому типу схемы относится исследуемая схема (рис. 21). Эта схема имеет Т-тип.

Выведем общую формулу расчета передаточных функций для схемы по Т-типу.

При работе электрической схемы на холостом ходу через $Z_2(p)$ ток не протекает, поэтому передаточная функция для данной схемы будет иметь вид (19):

$$K(p) = \frac{Z_2(p)}{Z_1(p) + Z_2(p)}. \quad (19)$$

В реальных условиях эксплуатации Т-схема работает на активную нагрузку; передаточная функция рассчитывается по формуле (20):

$$K(p) = \frac{Z_2(p) \cdot R}{Z_1(p) \cdot R - Z_1(p) \cdot Z_2(p) + Z_1(p) \cdot Z_2(p)}, \quad (20)$$

где $Z_{21}(p) = Z_2(p)$; $Z_{11}(p) = Z_1(p) + Z_2(p)$; $Z_{22}(p) = -(Z_2(p) + Z_3(p))$; $Z_{12}(p) = -Z_{21}(p)$.

Выведем общую формулу расчета передаточных функций для схемы по П-типу.

При работе электрической схемы на холостом ходу передаточная функция для данной схемы будет иметь вид (21):

$$K(p) = \frac{Z_4(p)}{Z_3(p)}, \quad (21)$$

где $Z_4(p) = \frac{Z_4(p) \cdot Z_6(p)}{Z_4(p) + Z_5(p) + Z_6(p)}$; $Z_3(p) = \frac{Z_4(p) \cdot Z_6(p)}{Z_4(p) + Z_5(p) + Z_6(p)}$.

При работе П-схемы на активную нагрузку; передаточная функция рассчитывается по формуле (22):

$$W(p) = \frac{Z_4(p) \cdot R}{Z_3(p) \cdot R - Z_3(p) \cdot Z_{44}(p) + Z_{14}(p) \cdot Z_4(p)}, \quad (22)$$

где $Z_{44}(p) = \frac{(Z_4(p) + Z_5(p)) \cdot Z_6(p)}{Z_4(p) + Z_5(p) + Z_6(p)}$; $Z_{14}(p) = Z_{41}(p)$.

Для ускорения расчетов можно использовать MathCad.

Имея передаточную функцию звена, которым представлена исследуемая схема, можно построить график переходного процесса и частотные характеристики схемы (рис. 27).

Передаточная функция системы представляется в виде (23):

$$W(p) = \frac{a_m s^m + a_{m-1} s^{m-1} + \dots + a_1 s + a_0}{b_n s^n + b_{n-1} s^{n-1} + \dots + b_1 s + b_0}. \quad (23)$$

В Matlab передаточная функция задается с помощью звена **Transfer Fcn** из библиотеки **Simulink / Continuous**.

Ввод коэффициентов передаточной функции (11) осуществляем следующим образом. Коэффициенты a_i следует вводить в поле **Numerator**, начиная с коэффициента a_m при старшей производной. Аналогично заполняется поле знаменателя передаточной функции **Denominator**, начиная с коэффициента b_n .

Входное воздействие на передаточную функцию подается с помощью элемента **Constant** из библиотеки **Simulink / Commonly Used Blocks**, а выходной сигнал регистрируется осциллографом **Simulink / Sinks / Scope**.

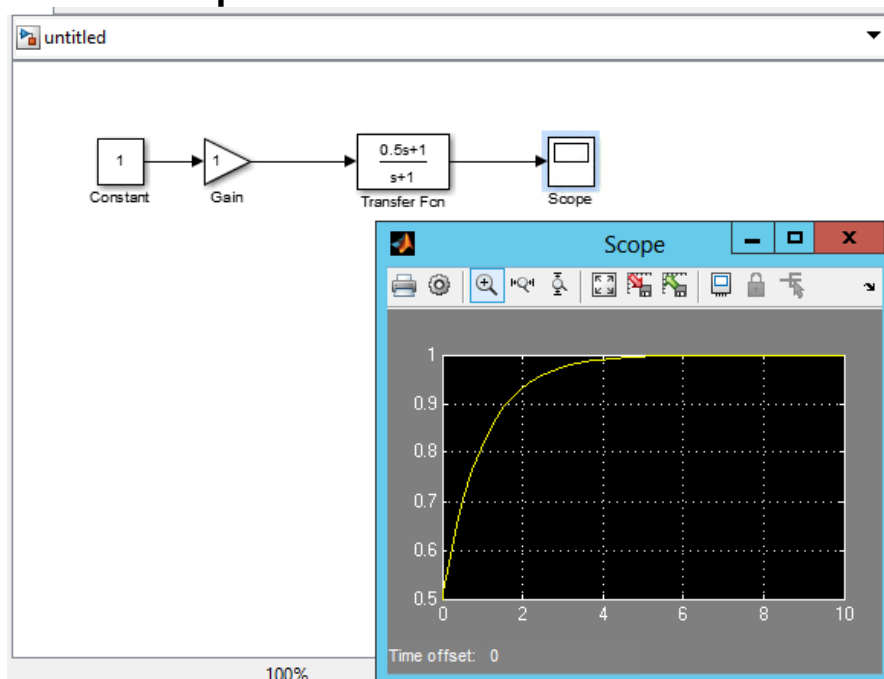


Рисунок 27. Получение переходной характеристики системы.

Рекомендуемые параметры моделирования показаны на рисунке 28.

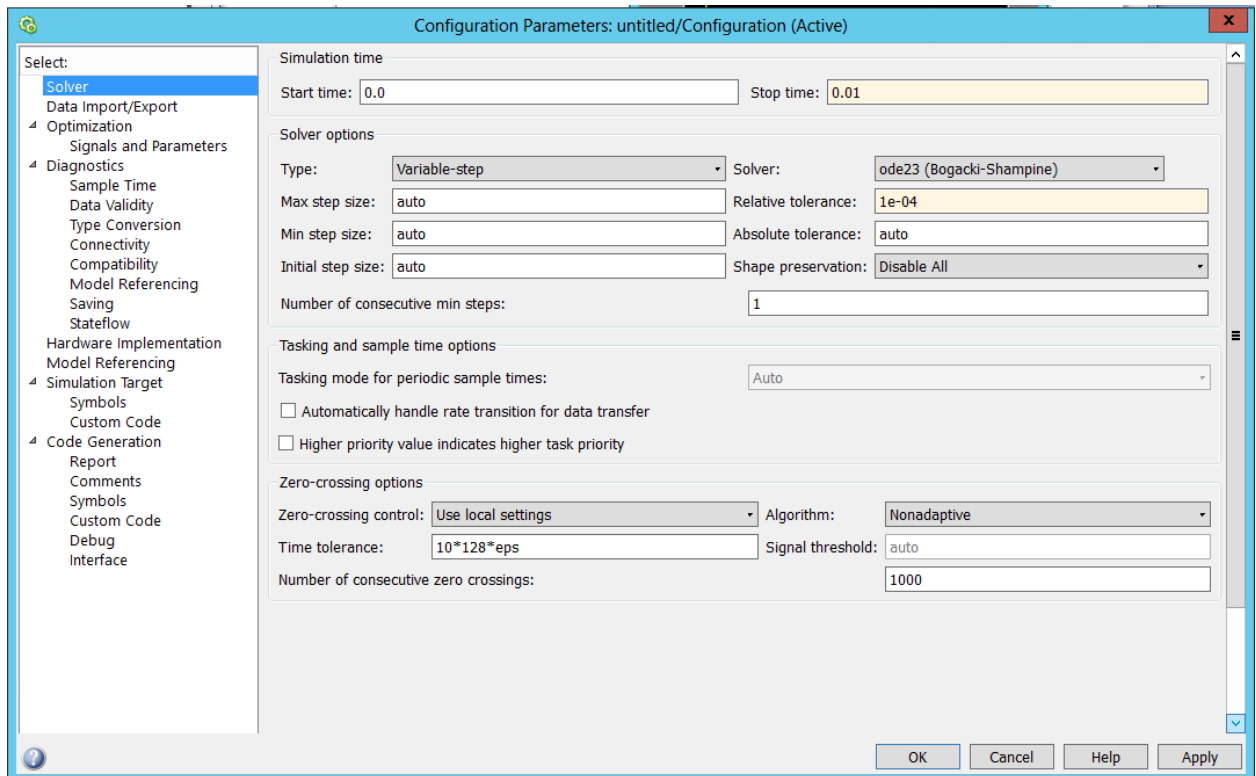


Рисунок 28. Параметры построения графика переходного процесса.

Получение графика переходного процесса и графиков частотных характеристик возможно при подключении к выходу схемы компонента **To Workspace** (рис. 29).

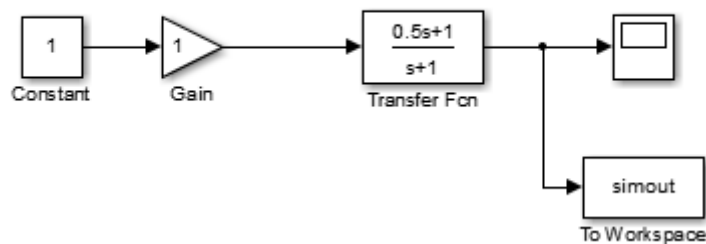


Рисунок 29. Исследование параметров переходного процесса.

В настройках этого компонента следует обязательно указать **Array** (Массив) в поле **Save Format** (Формат сохранения) (рис. 30).

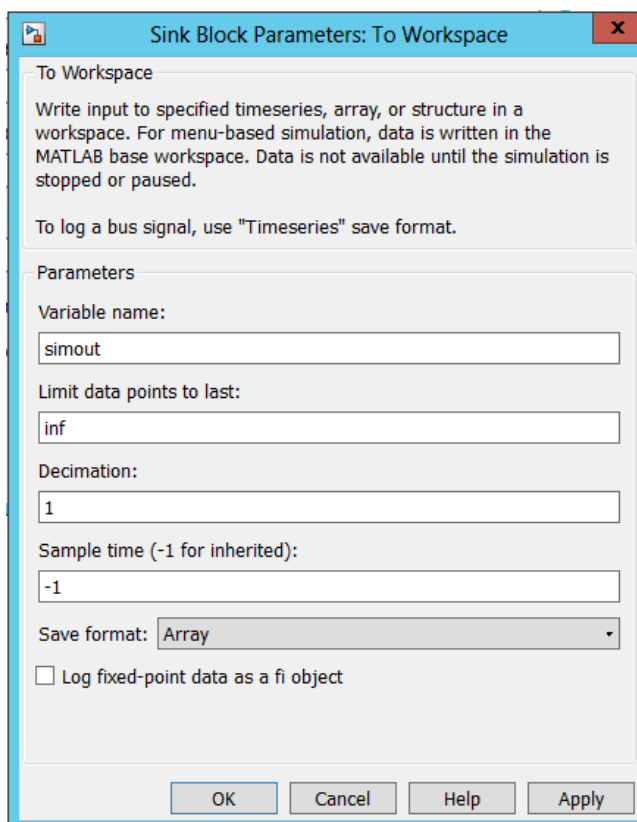


Рисунок 30. Параметры компонента **To Workspace**.

Повторяем расчёт переходных процессов, после чего переходим в окно Workspace, где отображаются переменные текущего проекта (рис. 31).

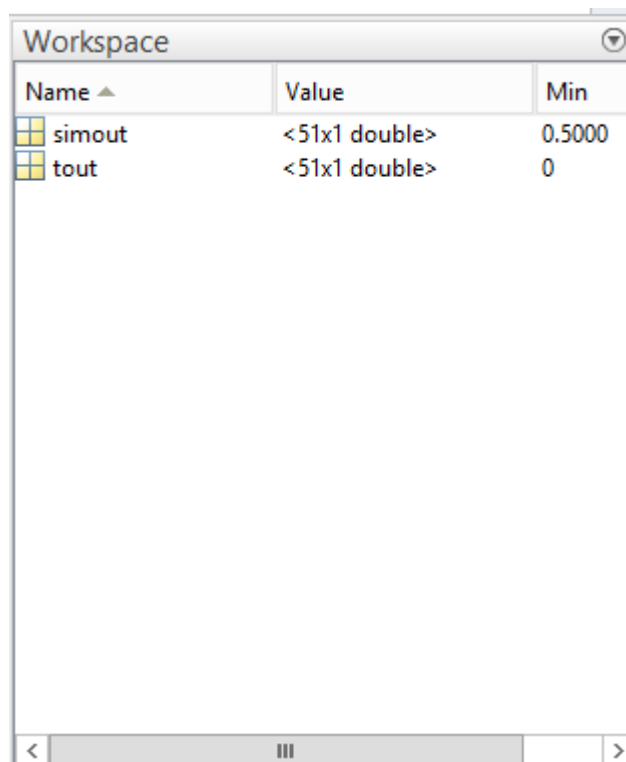


Рисунок 31. Переменные текущего проекта.

В окне **Workspace** выделяем переменную **Simout** и выполняем построение графика с помощью команды **Plot Simout**. В окне можно наблюдать переходный процесс при обработке входного единичного воздействия (рис. 32). С помощью команды **Edit\Save as** сохранить график в любом удобном формате.

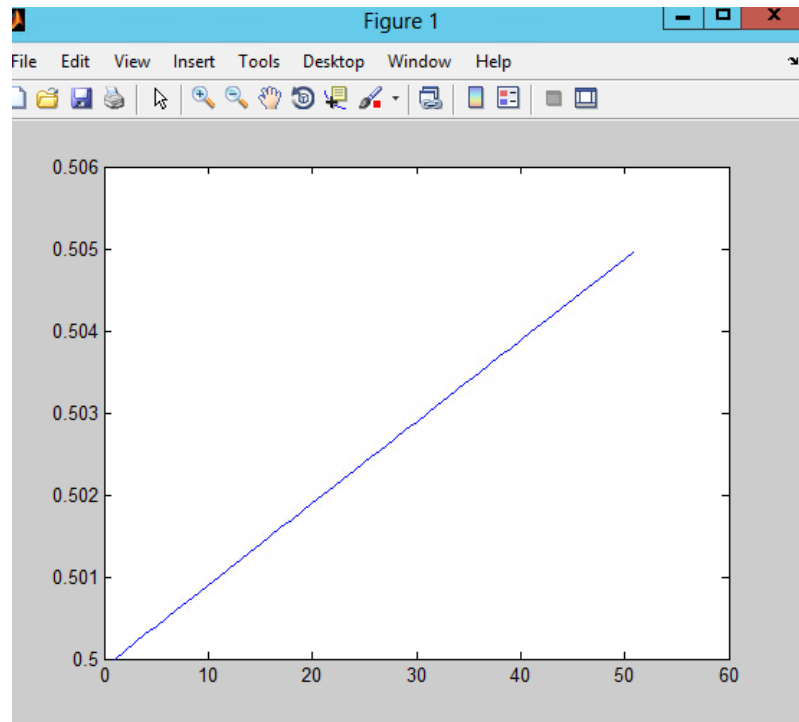


Рисунок 32. Переходная характеристика.

Для построения частотных характеристик нужно задать вход и выход передаточной функции с помощью команды **Linear Analysis/Input Point** и **Linear Analysis/Output Point** из выпадающего меню щелчком правой кнопки мыши на входе и выходе передаточной функции (рис. 33).

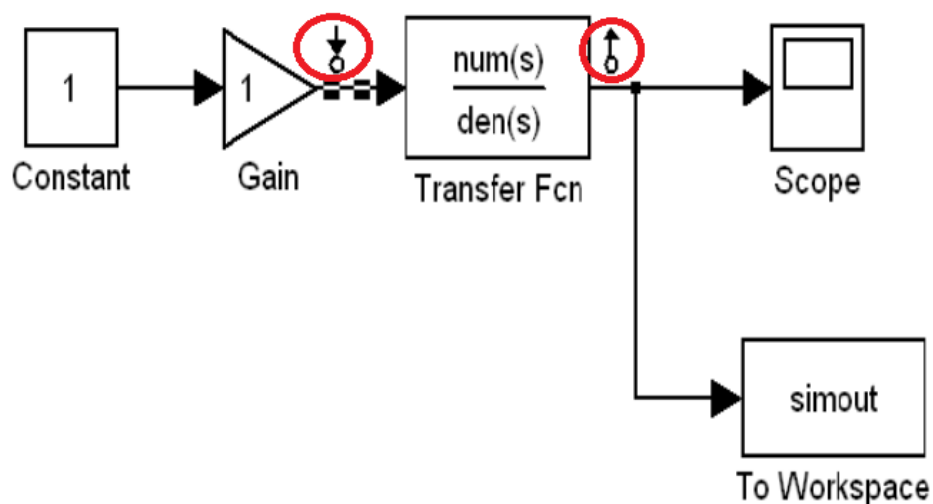


Рисунок 33. Назначение входа и выхода передаточной функции.

В открывшемся окне настройки **Control and Estimation Tools Manager** выбираем **Bode Response Plot** для построения частотных характеристик и нажимаем на **Linearize Model** (рис. 34).

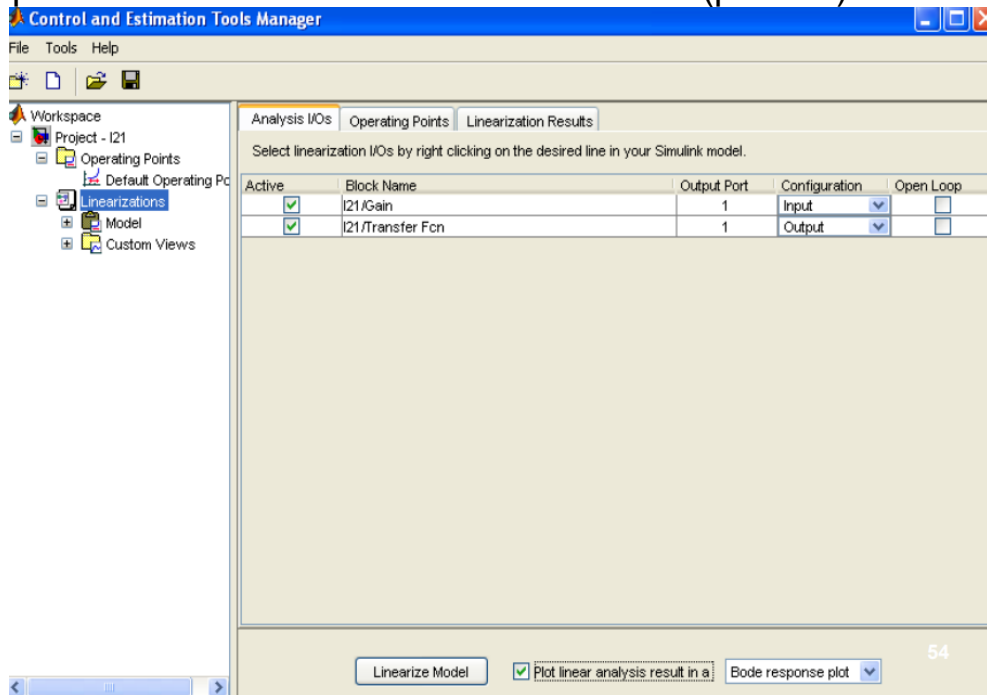


Рисунок 34. Настройки Control and Estimation Tools Manager.

В окне построения в верхней части изображена амплитудно-частотная характеристика, а в нижней части - фазо-частотная характеристика (рис. 35).

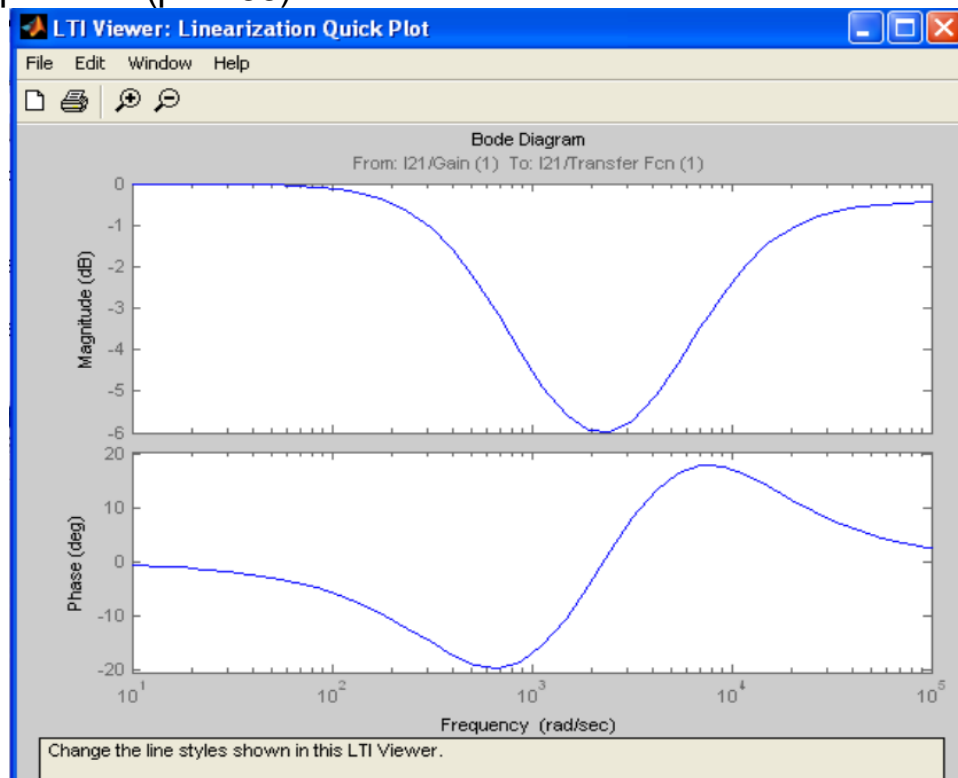
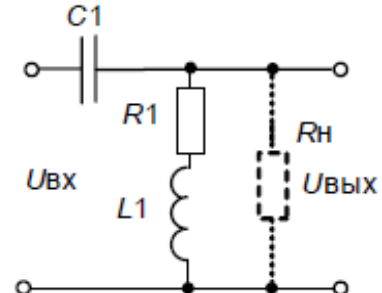
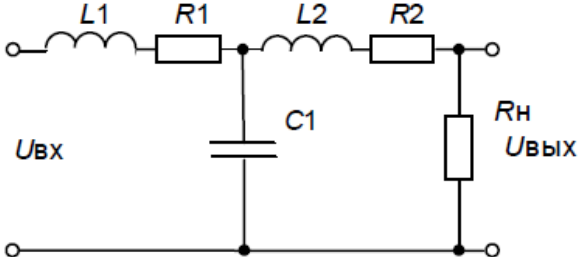
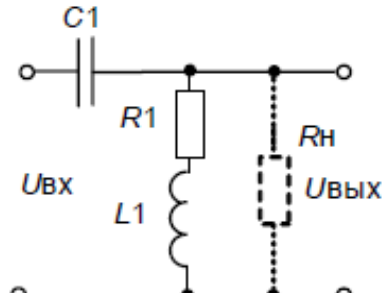
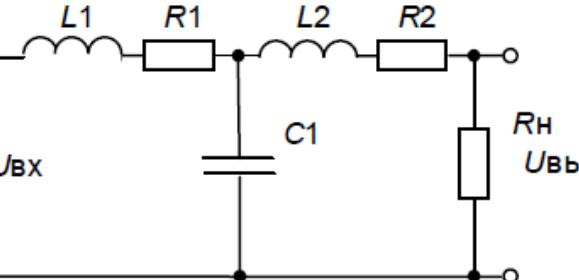
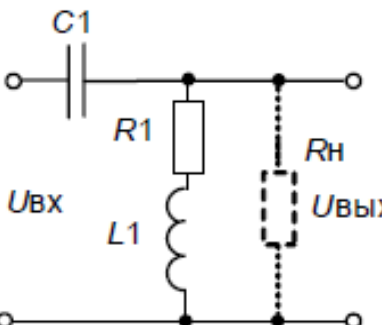


Рисунок 35. Полученные частотные характеристики.

Таблица 6. Варианты заданий.

Вариант	Принципиальная схема	Параметры схемы
1		$L_1 = 5 \text{ мГн};$ $R_1 = 1 \text{ Ом};$ $C_1 = 1 \text{ мкФ};$ $L_2 = 10 \text{ мГн};$ $R_2 = 1 \text{ Ом};$ $R_H = 4 \text{ кОм}.$
2		$L_1 = 10 \text{ мГн};$ $R_1 = 1 \text{ Ом};$ $C_1 = 10 \text{ мкФ};$ $R_H = 8 \text{ кОм}.$
3		$L_1 = 50 \text{ мГн};$ $R_1 = 1 \text{ Ом};$ $C_1 = 10 \text{ мкФ};$ $L_2 = 5 \text{ мГн};$ $R_2 = 100 \text{ Ом};$ $R_H = 2 \text{ кОм}.$
4		$L_1 = 8 \text{ мГн};$ $R_1 = 10 \text{ Ом};$ $C_1 = 1 \text{ мкФ};$ $R_H = 6 \text{ кОм}.$
5		$L_1 = 5 \text{ мГн};$ $R_1 = 1 \text{ Ом};$ $C_1 = 1 \text{ мкФ};$ $L_2 = 5 \text{ мГн};$ $R_2 = 100 \text{ Ом};$ $R_H = 2 \text{ кОм}.$

6		$L_1 = 5 \text{ мГн};$ $R_1 = 1 \text{ Ом};$ $C_1 = 1 \text{ мкФ};$ $R_H = 2 \text{ кОм}.$
7		$L_1 = 50 \text{ мГн};$ $R_1 = 1 \text{ Ом};$ $C_1 = 10 \text{ мкФ};$ $L_2 = 50 \text{ мГн};$ $R_2 = 10 \text{ Ом};$ $R_H = 6 \text{ кОм}.$
8		$L_1 = 15 \text{ мГн};$ $R_1 = 1 \text{ Ом};$ $C_1 = 10 \text{ мкФ};$ $R_H = 6 \text{ кОм}.$
9		$L_1 = 5 \text{ мГн};$ $R_1 = 1 \text{ Ом};$ $C_1 = 1 \text{ мкФ};$ $L_2 = 50 \text{ мГн};$ $R_2 = 10 \text{ Ом};$ $R_H = 6 \text{ кОм}.$
10		$L_1 = 10 \text{ мГн};$ $R_1 = 10 \text{ Ом};$ $C_1 = 10 \text{ мкФ};$ $R_H = 8 \text{ кОм}.$

Содержание отчета

1. Цель работы.
2. Подробное описание методики построения модели исследуемого устройства в программе.
3. Исследование объекта, построение необходимых зависимостей, анализ результатов.
4. Выводы.

Лабораторная работа 5. Реализация процессов широтно-импульсной модуляции в MatLab

Задание 1. Изучить параметры широтно-импульсной модуляции и влияние этих параметров на формирование сигналов в MatLab согласно варианту (см. табл. 7).

Создаем имитационную модель широтно-импульсного модулятора в MatLab (рис. 36).

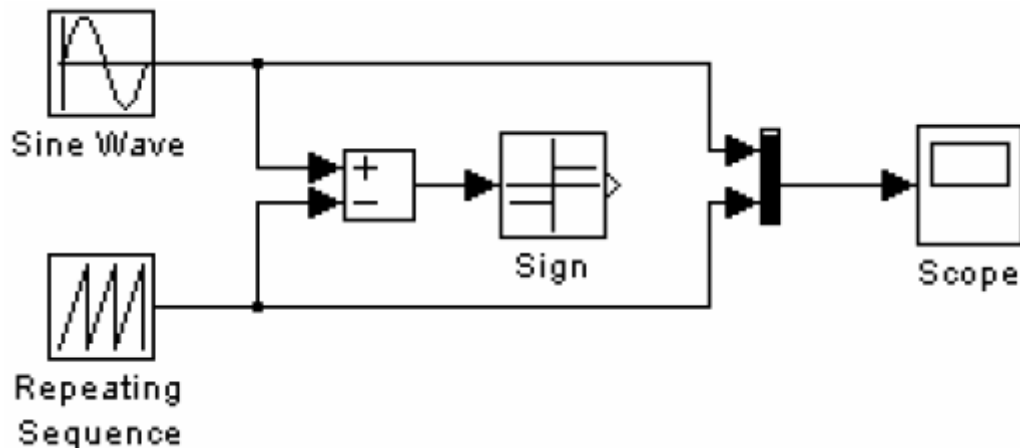


Рисунок 36. Исследуемая модель.

Рассмотрим особенности настройки блоков исследуемой модели.

Генерация сигналов пилообразной формы происходит в блоке **Repeating Sequence**, где имеется 2 параметра настройки (рис. 37):

Time values (временные значения) – числовой массив, где нужно задать два значения настройки: первое – начальное время генерации, второе – период генерации;

Output values (выходные значения) – числовой массив, где нужно задать два значения настройки: первое – значение на выходе блока в начальный момент времени, второе – в конце периода генерации.

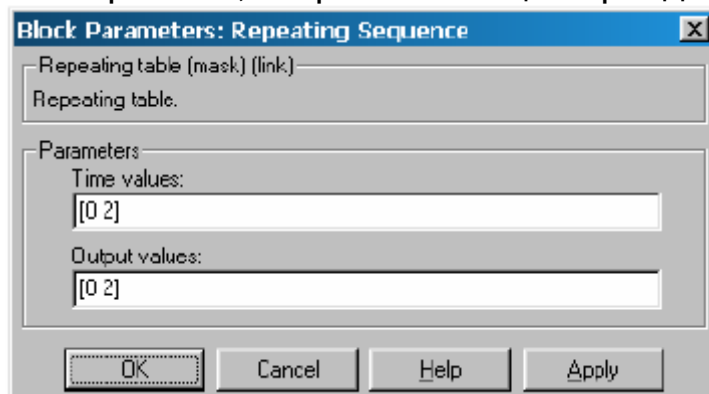


Рисунок 37. Настройка параметров блока **Repeating Sequence**.

Генерация гармонических колебаний с заданными параметрами происходит в блоке **Sine Wave**, где имеется четыре параметра настройки:

Amplitude (амплитуда) – амплитуда синусоиды;

Frequency (rad/sec) (частота (рад/сек)) – частота синусоиды;

Phase (rad) (фаза (рад)) – начальная фаза колебания;

Sample time (модельное время) – служебный параметр.

Этот блок моделирует напряжение, подаваемое на модулятор.

Суммирование входных сигналов происходит в блоке **Sum**. Этот блок использует, когда необходимо суммировать:

- 1) входные сигналы (даже в случае разных знаков сигнала);
- 2) элементы вектора, поступающего на вход.

В блоке имеется один параметр настройки: **List of Signs** (список знаков). Его значения могут задаваться в виде *последовательности знаков «+» и «-»* (число знаков - это число входов и их полярность), в виде *положительной константы* (больше 1, здесь задается число входов положительных блоков) и в виде *символа «1»* (указывает на использование блоков во втором режиме).

В нашем случае используем первый вариант задания параметра этого блока.

Блок **Sign** определяет знак входного сигнала. На выходе блока появляется значение, равное «1», если значение входного сигнала больше 0, «-1» - если меньше, и «0» - если сигнал на входе равен нулю.

Блок **Mux** выполняет объединение входных величин в один линейный вектор. Размерность результирующего вектора равна суммарному количеству элементов, поступающих на входные порты блока. Блок имеет один параметр настройки – **Numbers of inputs** (число входов). Установим это значение 2.

Блок **Scope** был описан ранее.

В качестве параметров моделирования установим значения полей на вкладке **Solver**: **Start time** = 0, **Stop time** = 4, **Max Step size** = 0.05.

После запуска процесса моделирования, получаем графики процесса моделирования (рис. 38).

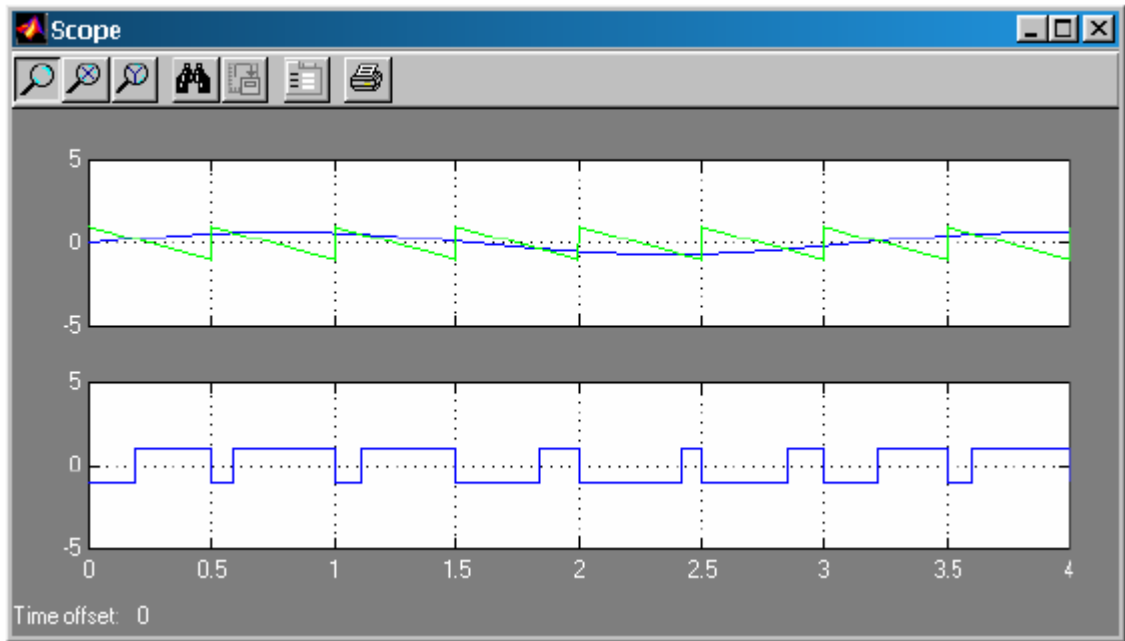


Рисунок 38. Итоговые результаты моделирования.

Таблица 7. Варианты заданий к рисунку 36.

Вариант	Параметры схемы
1	<p>Блок Repeating Sequence Time values = [0 0.5]; Output values = [1 -1].</p> <p>Блок Sine Wave Amplitude = 0.7; Frequency = 2; Phase = 0.</p>
2	<p>Блок Repeating Sequence Time values = [0 0.6]; Output values = [2 -2].</p> <p>Блок Sine Wave Amplitude = 0.8; Frequency = 2; Phase = 0.</p>
3	<p>Блок Repeating Sequence Time values = [0 0.7]; Output values = [3 -3].</p> <p>Блок Sine Wave Amplitude = 0.9; Frequency = 2; Phase = 0.</p>
4	<p>Блок Repeating Sequence Time values = [0 0.8]; Output values = [4 -4].</p> <p>Блок Sine Wave Amplitude = 1; Frequency = 2; Phase = 0.</p>
5	<p>Блок Repeating Sequence Time values = [0 0.9]; Output values = [1 -1].</p> <p>Блок Sine Wave Amplitude = 1.1; Frequency = 2; Phase = 0.</p>

6	<p>Блок Repeating Sequence Time values = [0 0.1]; Output values = [2 -2].</p> <p>Блок Sine Wave Amplitude = 1.2; Frequency = 2; Phase = 0.</p>
7	<p>Блок Repeating Sequence Time values = [0 0.15]; Output values = [3 -3].</p> <p>Блок Sine Wave Amplitude = 1.3; Frequency = 2; Phase = 0.</p>
8	<p>Блок Repeating Sequence Time values = [0 0.2]; Output values = [4 -4].</p> <p>Блок Sine Wave Amplitude = 1.4; Frequency = 2; Phase = 0.</p>
9	<p>Блок Repeating Sequence Time values = [0 0.25]; Output values = [1 -1].</p> <p>Блок Sine Wave Amplitude = 1.5; Frequency = 2; Phase = 0.</p>
10	<p>Блок Repeating Sequence Time values = [0 0.3]; Output values = [2 -2].</p> <p>Блок Sine Wave Amplitude = 1.8; Frequency = 2; Phase = 0.</p>

Содержание отчета

1. Цель работы.
2. Подробное описание методики построения модели исследуемого устройства в программе.
3. Исследование объекта, построение необходимых зависимостей, анализ результатов.
4. Выводы.

Лабораторная работа 6. Расчет электростатических полей методом конечных элементов с помощью программного продукта ELCUT

Задание 1. Рассчитать электростатические характеристики конденсатора с параллельными обкладками с помощью ELCUT.

Для исследования задан конденсатор с двумя параллельными обкладками со следующими характеристиками конструкции: разность потенциалов между обкладками 100 В , $l = 3\text{ мм}$, $d = 0,7\text{ мм}$ (рис. 39).

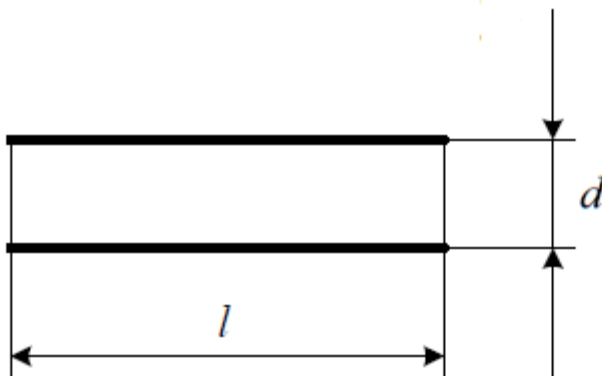


Рисунок 39. Исследуемый объект.

Этап 1. Приступаем к созданию исследуемого объекта в программе ELCUT.

Запускаем программу **ELCUT**. Выбираем пункт **Файл** в меню и далее **Создать задачу** (рис. 40).

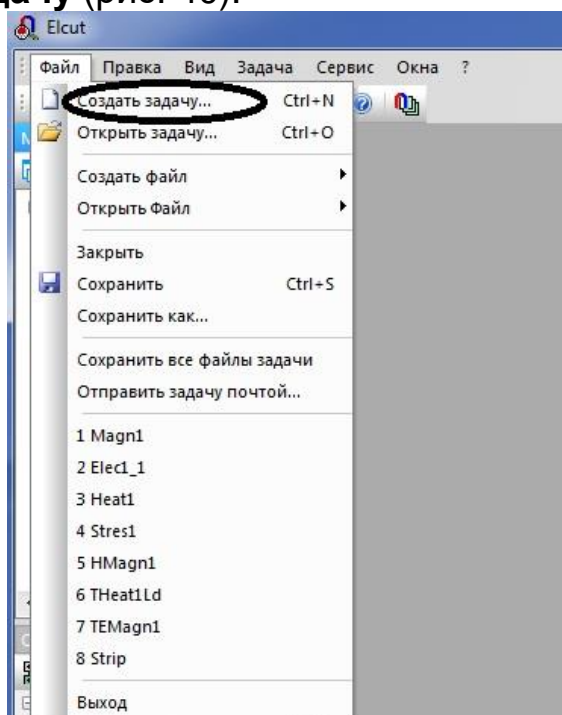


Рисунок 40. Создание задачи.

При создании файла задачи необходимо ввести **Имя файла задачи** в соответствующей строке и указать директорию для сохранения файла (рис. 41). Мы ввели имя **Расчет конденсатора**, вы можете ввести любое имя файла.

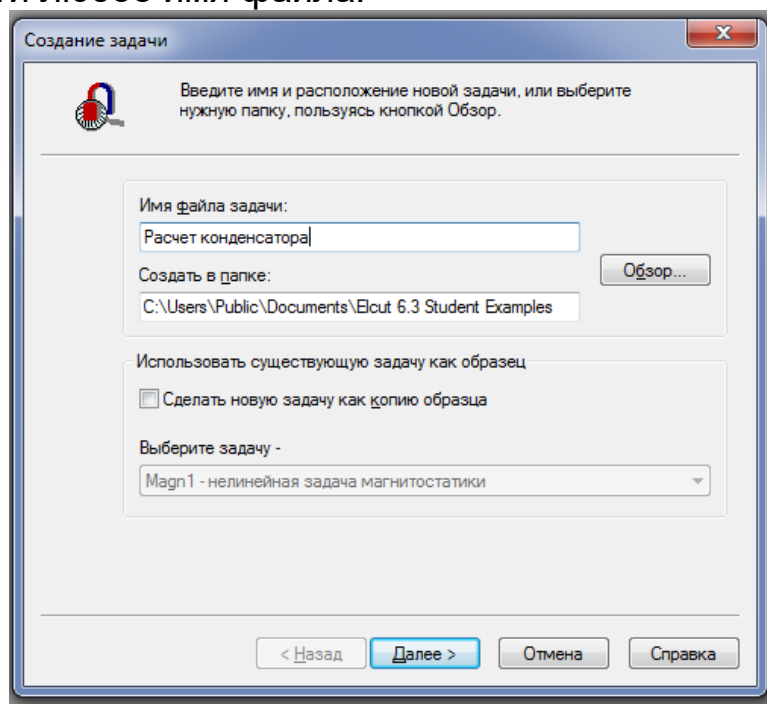


Рисунок 41. Создание файла задачи.

Нажать **Далее**. В следующем окне выбрать тип задачи **Электростатическое поле**, класс модели **Плоская** и параметр расчета **Обычный**. Здесь же можно выбрать единицы длины **Миллиметры**, систему координат **Декартовы** (рис. 42).

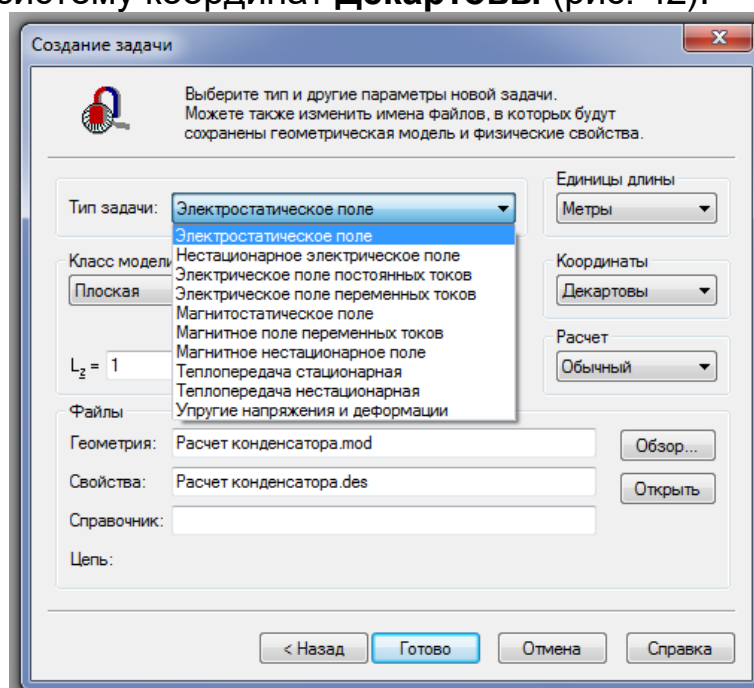


Рисунок 42. Выбор параметров файла задачи.

Как видно из рисунка 4, мы создали файл геометрии с расширением **.mod** и файл свойств с расширением **.des**. Нажать **Готово**.

Интерфейс программы выглядит следующим образом (рис. 43).

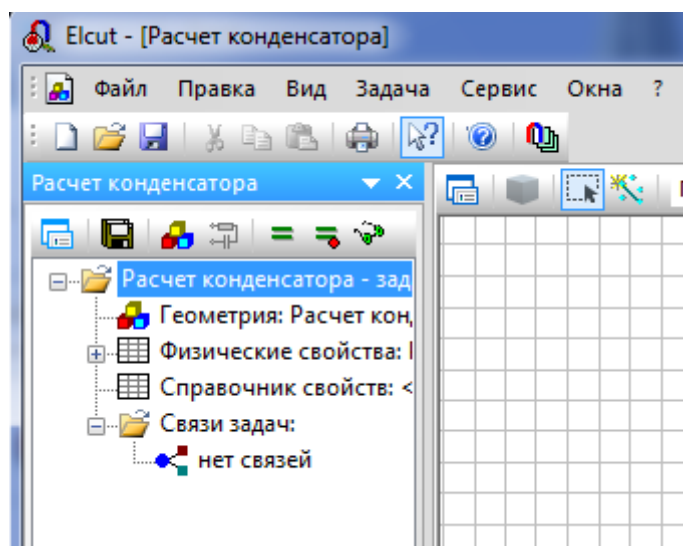


Рисунок 43. Интерфейс программы ELCUT.

Далее можно просмотреть пункты списка в окне программы при помощи левой кнопки мыши (рис. 44).

Мы видим выпадающие параметры в пункте **Физические свойства:** **Метки блоков, ребер и вершин**. Пока свойства не заданы, список пуст.

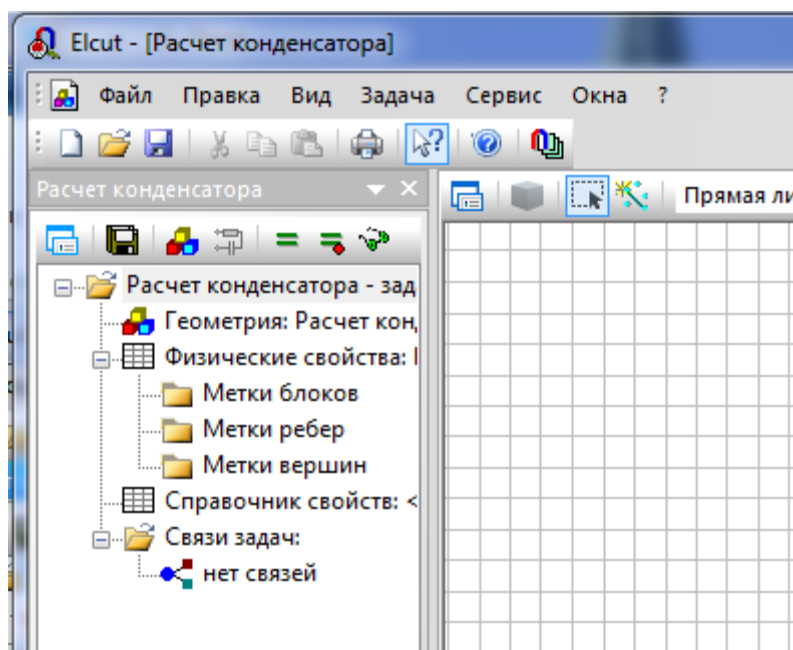


Рисунок 44. Физические свойства объекта.

Этап 2. Приступаем к созданию геометрической модели объекта.

Итак, в окне программы выделим строку **Геометрия: Расчет конденсатора** и приступаем к рисованию плоской модели конденсатора. Рисование происходит при помощи инструментов с панели рисования (рис. 45). Для перехода в режим рисования и наоборот мы переключаемся между кнопками **Выделение объектов** и **Вставлять вершины и ребра**.



Рисунок 45. Панель рисования модели.

Изображаем конструкцию конденсатора (две обкладки без выводов). Для этого протягиваем при помощи левой кнопки мыши необходимые линии конструкции. Геометрические размеры можно определить по текущим координатам курсора (они указываются в нижней части файла модели). Изменить масштаб можно кнопками **Увеличить** или **Уменьшить** на панели.

На этом построение геометрической модели не заканчивается. Понятно, что в конструкцию конденсатора входит **2 обмотки**, сама конструкция находится в **воздухе**.

Для дальнейшего исследования необходимо применить **метод конечных элементов (МКЭ)**.

Как известно, поля в пространстве распространяются бесконечно. Для их исследования необходимо наличие локальной области, а создание локальной области заключается в ограничении области исследования объекта. В программе ELCUT это реализуется путем создания ограниченной области вокруг конденсатора любой формы с помощью такого элемента, как **ЭКРАН**, т.е. необходимо охватить конденсатор экраном.

Окончательный вид геометрической модели конденсатора представлен на рисунке 46.

Этап 3. Переходим к введению свойств объекта.

Согласно рисунку 44, необходимо ввести свойства следующих элементов модели: блоков, ребер и вершин. Рассмотрим эти понятия.

Для этого необходимо описать любой изучаемый объект. Он может находиться в разных средах, иметь различное количество составляющих конструкцию элементов.

Изучаемая конструкция согласно геометрической модели (рис. 8) содержит следующие элементы: **2 обкладки и экран**, т.е. конструкция имеет **3 основных ребра**.



Рисунок 46. Геометрическая модель объекта (две обкладки и экран).

Далее, исследуемая конструкция находится в воздухе, между обкладками нет никаких других материалов, поэтому в этой конструкции **один блок – воздух**.

Теперь вводим свойства ребер и блоков. Для этого необходимо перейти из режима рисования и два раза щелкнуть левой клавишей по элементу и назвать его. Вместо **Нет** вводим название **Верхняя**. Так вводятся все ребра конструкции (рис. 47).

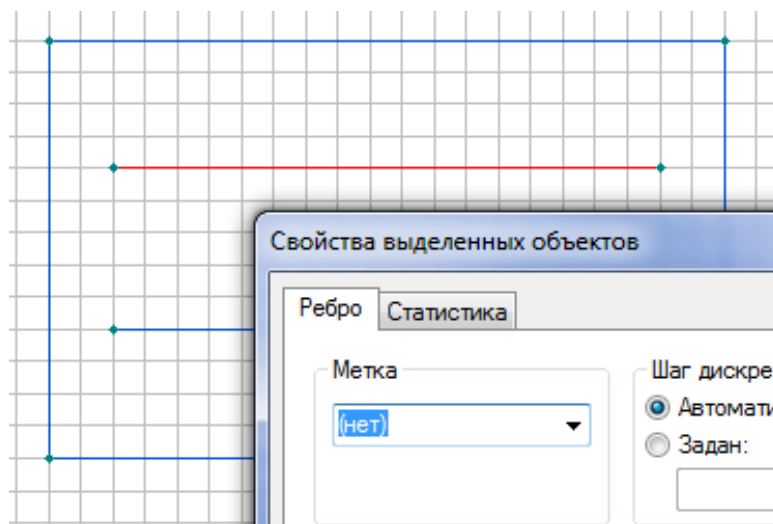


Рисунок 47. Ввод ребер конструкции.

Получившийся результат после введения ребер конструкции представлен на рисунке 48.

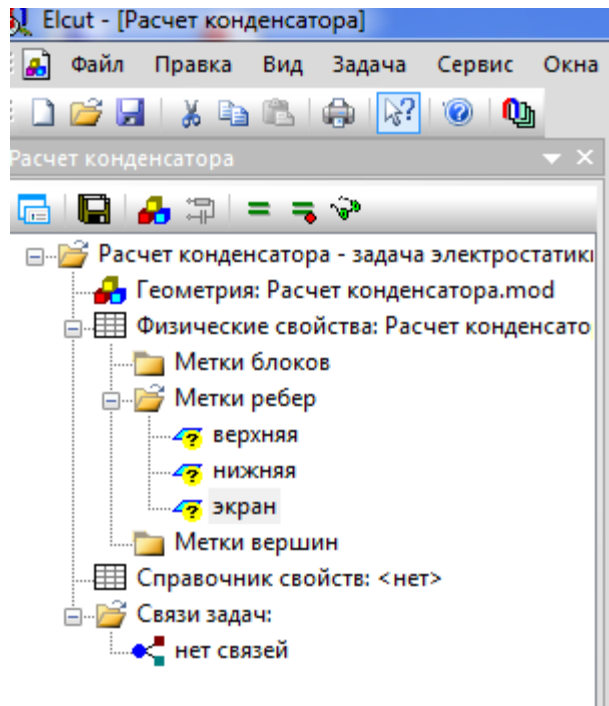


Рисунок 48. Ввод ребер конструкции.

Как видим из рисунка 48, количество ребер три, то есть можно сделать следующие выводы:

1) количество ребер при анализе конструкции три; количество введенных ребер три; то есть все ребра конструкции введены;

2) желтый знак вопроса рядом с позицией ребра означает, что можно приступить к введению его числовых параметров. После того, как параметры будут введены, знаки вопросы исчезнут. Пока есть хоть один знак вопроса после введения параметров, приступить к анализу полей нельзя. **Отсутствие знаков вопроса является индикатором для продолжения анализа конструкции.**

Теперь необходимо ввести информацию о блоках (рис. 49).

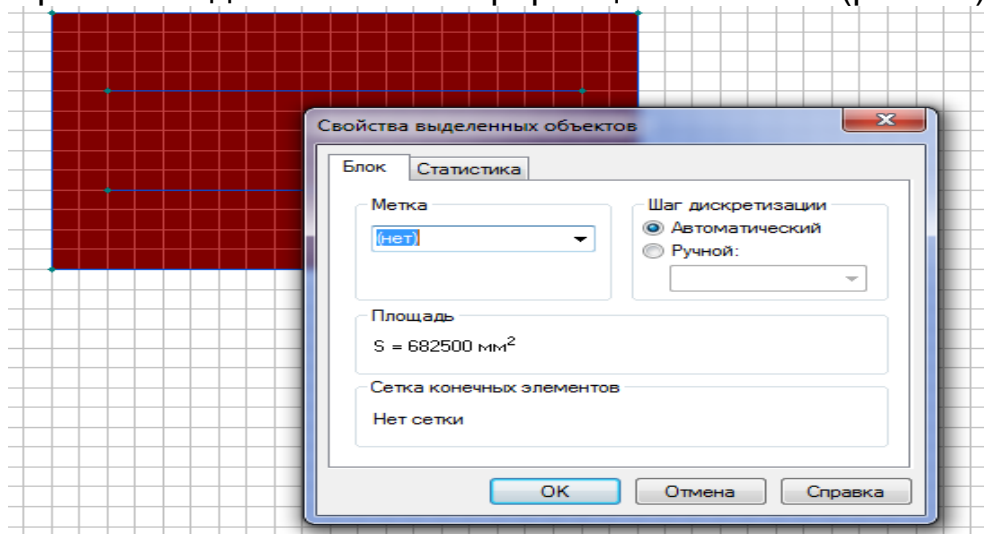


Рисунок 49. Выделение блока конструкции.

Как говорилось ранее, в конструкции один блок. Двойным щелчком левой клавишей мыши отмечаем область на геометрической модели, где находится воздух, вместо слова **Нет** вводим метку блока **Воздух** (рис. 50).

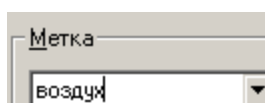


Рисунок 50. Метка блока.

На данном этапе прописаны ребра и блоки конструкции.

Теперь переходим к введению свойств блоков и ребер. Для введения параметров необходимо два раза левой клавишей мыши щелкнуть по строке с названием блока, и в диалоговом окне появится поле для введения параметров блока, а именно для введения значения относительной диэлектрической проницаемости. Как известно, воздух является изотропным диэлектриком, его свойства не зависят от координаты, его относительная диэлектрическая проницаемость равна 1 (рис. 51).

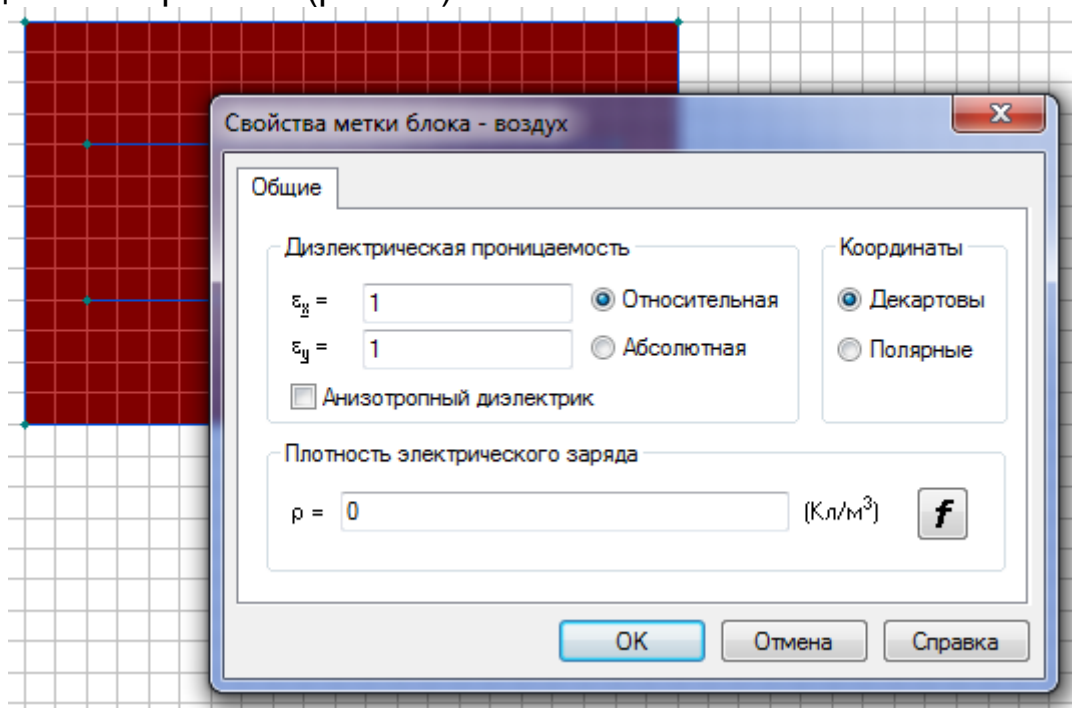


Рисунок 51. Ввод параметров блока.

Аналогичным способом вводим параметры ребер (согласно заданию есть значение потенциала).

Теперь модель готова для анализа. Приступаем к построению сетки в области расчета. Для этого надо использовать кнопку панели инструментов **Построение сетки** (рис. 45).

Правильность выполнения дальнейшей работы зависит от того, что построенная сетка области расчета охватывает область расчета (рис. 52).

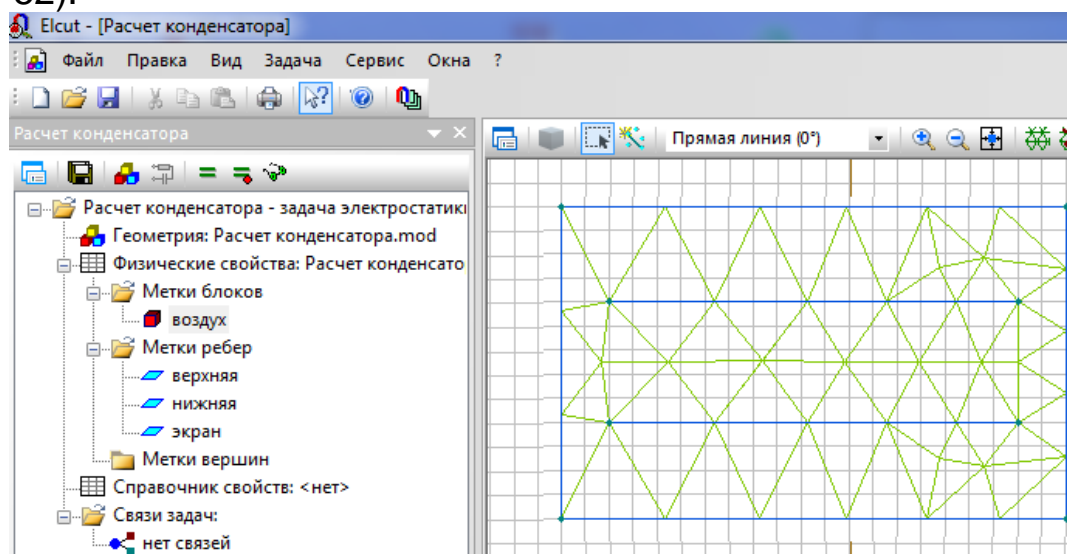


Рисунок 52. Построение сетки в области расчета.

Если все этапы выполнены правильно, то получится картина, аналогичная картине на рисунке 52 (признаки правильности выполнения исследования: нет знаков вопроса, область расчета полностью охвачена).

Этап 4. Переходим к получению результатов моделирования.

Далее можно приступить к расчету. Для этого необходимо выделить файл задачи и выбрать **Решить задачу** (знак = в меню программы) (рис. 52). Далее на вопрос об изменении файлов и сохранении необходимо ответить **Да**.

Таким способом были получены результаты моделирования. Изолинии потенциала представлены на рисунке 53.

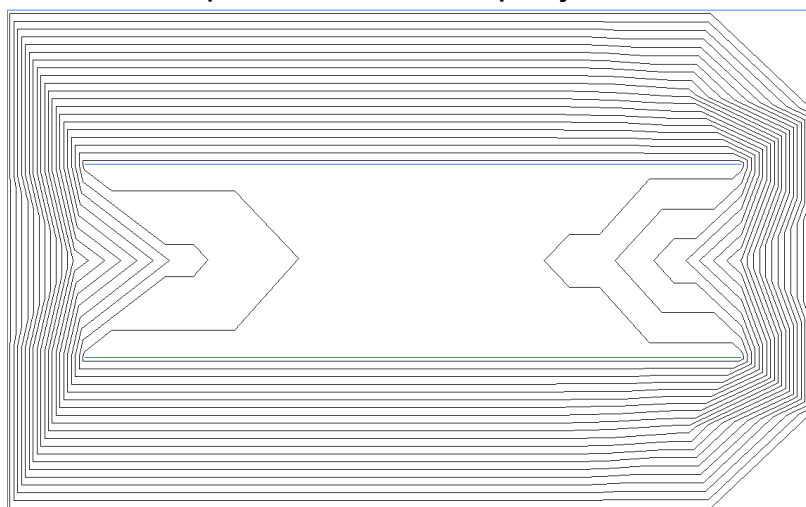


Рисунок 53. Результаты моделирования.

Далее правой клавишей щелкаем по получившейся картине и выбираем картины полей исследуемого объекта. Как пример, приведем картину векторного поля электрического смещения (рис. 54).

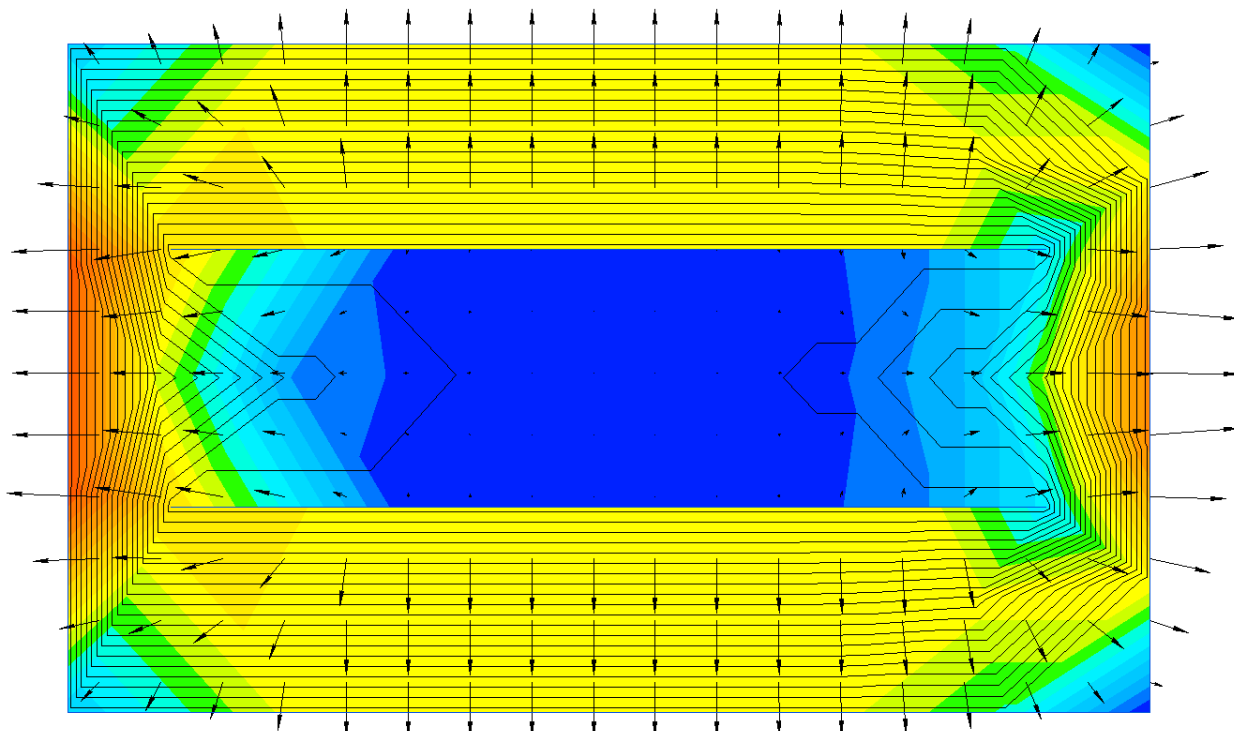


Рисунок 54. Векторное поле электрического смещения.

На рисунке 55 приведен перечень всех полученных картин полей.

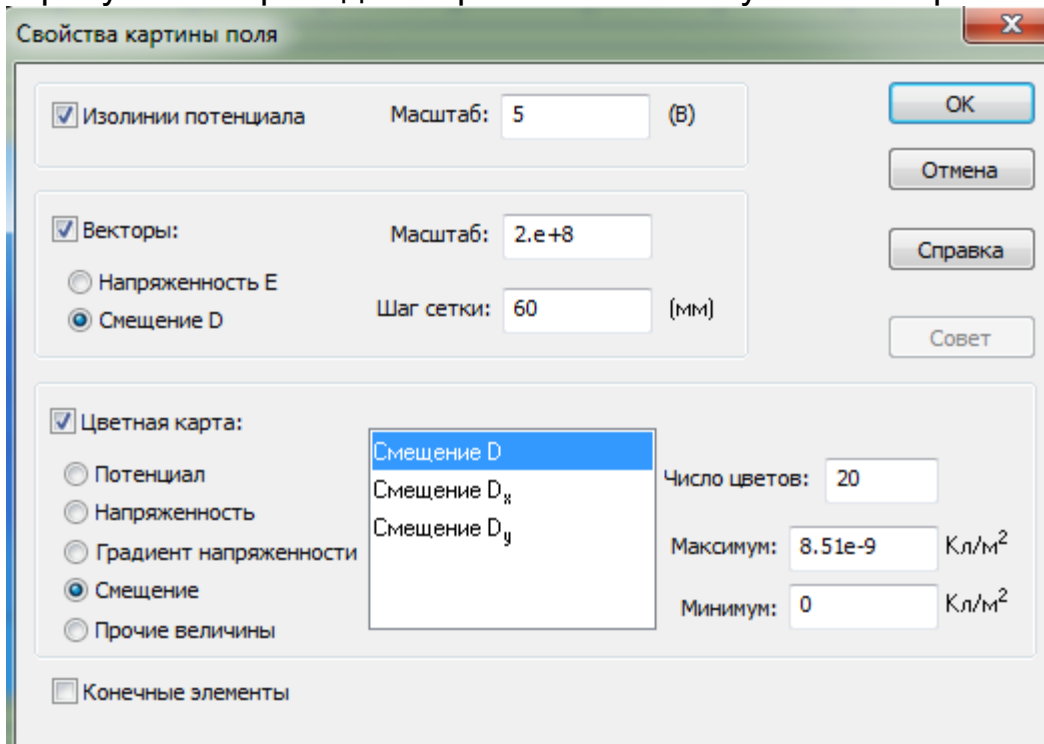


Рисунок 55. Векторное поле электрического смещения.

При анализе результатов моделирования можно сравнить такие параметры электрического поля исследуемого электротехнического устройства, как напряжённость, электрическое смещение, плотность энергии и др. (рис. 56).

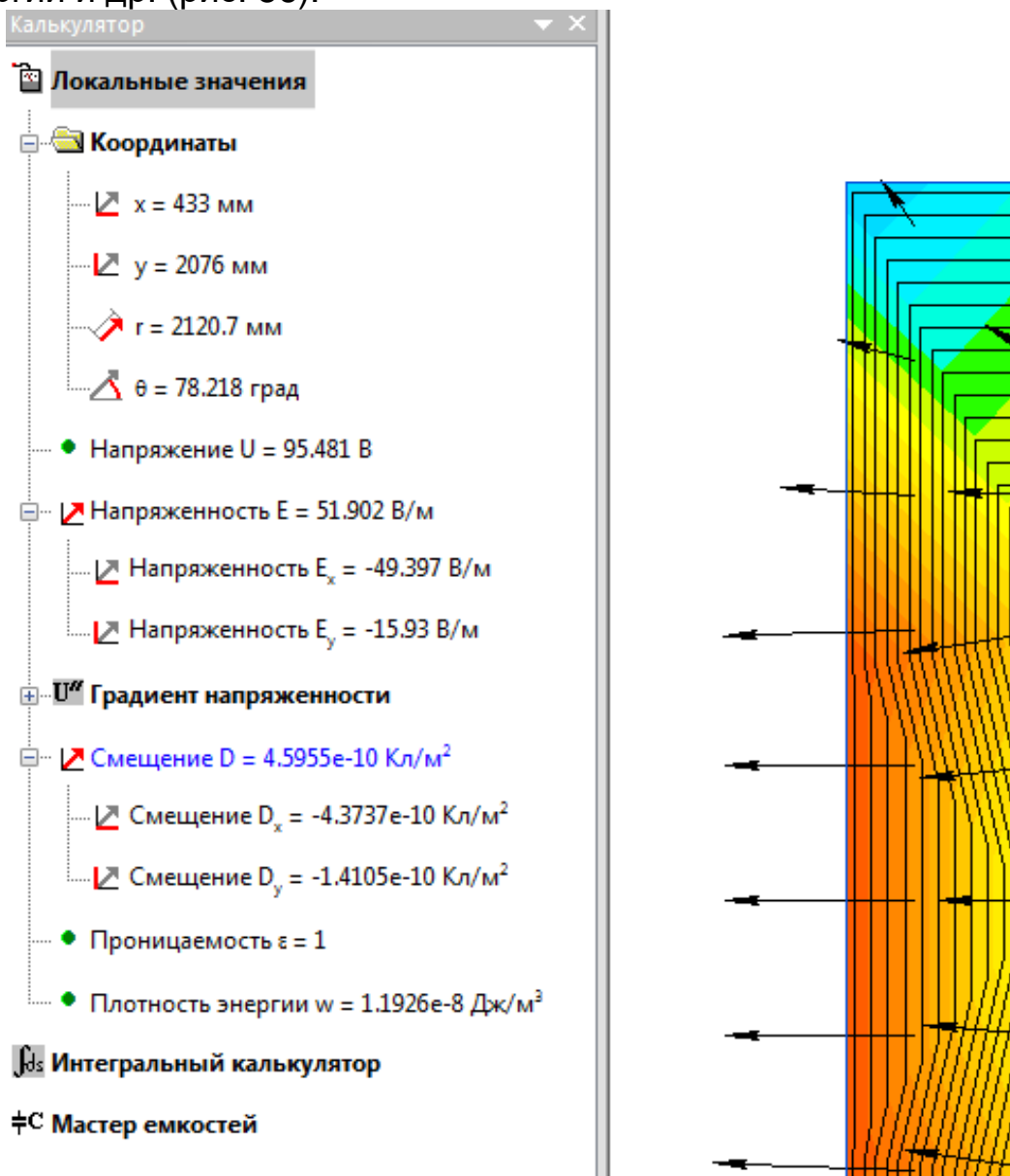


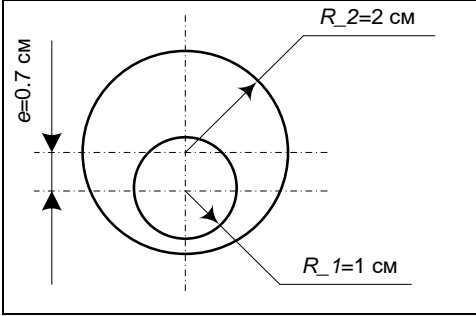
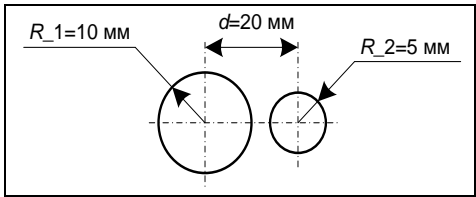
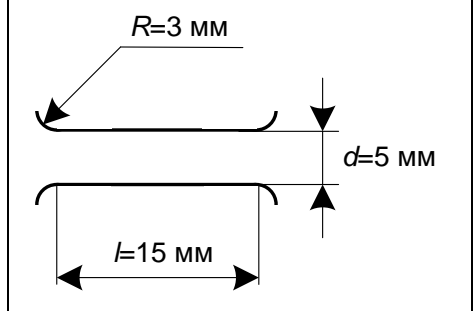
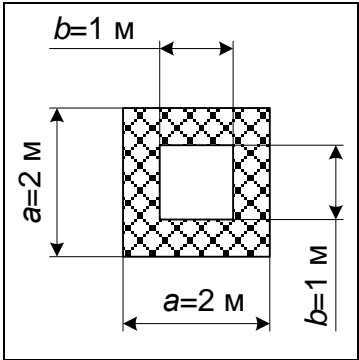
Рисунок 56. Локальные значения величин.

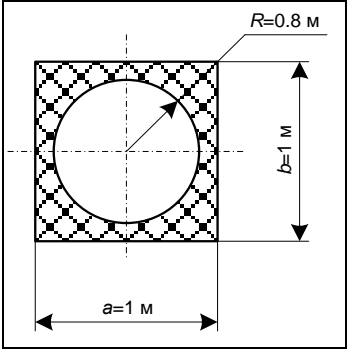
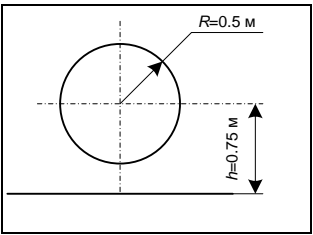
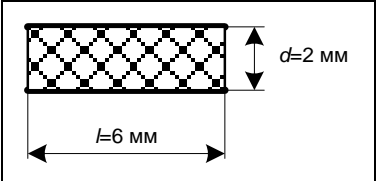
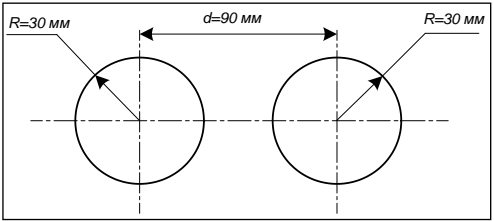
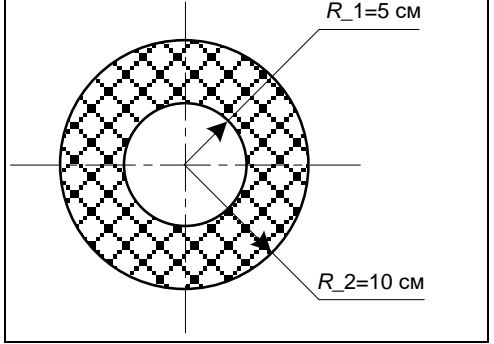
На основании моделирования электрических полей можно сделать выводы и рекомендации по конструктивной оптимизации технических устройств.

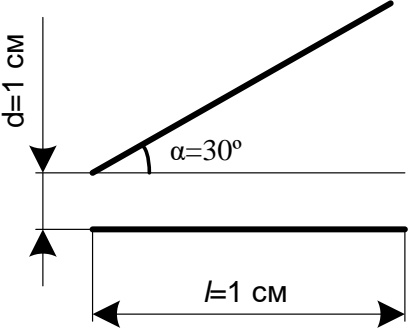
Задание 2. Рассчитать электростатические характеристики объекта (согласно варианту) с помощью ELCUT.

Согласно методике, описанной в задании 1 этой работы, необходимо провести моделирование процессов конструкции согласно варианту (табл. 8).

Таблица 8. Варианты заданий.

№ варианта	Конструкция устройства	Краткое описание
1		<p>Между несоосно расположенными цилиндрическими проводниками расположен диэлектрик – фторопласт-4 (ПТФЭ). Потенциал внутреннего цилиндра 300 В, внешнего – 100 В.</p>
2		<p>Два цилиндрических проводника различного диаметра пространство имеют потенциал внутреннего цилиндра 200 В, внешнего – 50 В.</p>
3		<p>В конденсаторе со скругленными параллельными обкладками подается 100 В на верхнюю обкладку, - 100 В на нижнюю.</p>
4		<p>В системе из двух соосно расположенных проводников подается потенциал 300 В на внутренний проводник. Второй проводник имеет свойства экрана. Материал диэлектрика между проводниками – полипропилен.</p>

№ варианта	Конструкция устройства	Краткое описание
5		<p>В системе из двух проводников подается потенциал 100 В на цилиндрический проводник. Он покрыт изоляцией из полиэтилентерефталата (ПЭТФ), внешний проводник служит экраном.</p>
6		<p>В системе «цилиндрический проводник – плоская обкладка» проводник имеет потенциал 100 В, обкладка – 200 В, система находится в воздухе.</p>
7		<p>Плоский конденсатор на основе титаната бария в качестве диэлектрика (при относительной проницаемости 600). Имеет потенциал верхней обкладки 200 В, нижней – -200 В.</p>
8		<p>В воздушной среде находятся два цилиндрических проводника с потенциалами 200 В и -200 В (выбрать любой случай).</p>
9		<p>В системе из двух соосно расположенных проводников подается потенциал 200 В на внутренний проводник, внешний - 0. Материал диэлектрика между проводниками - стекловолокно.</p>

10		<p>Конденсатор, обкладки которого не параллельны. В качестве диэлектрика используется поликарбонат (относительная диэлектрическая проницаемость 2,6), $\alpha = 30^\circ$</p>
----	---	--

Содержание отчета

1. Цель работы.
2. Конструктивный эскиз исследуемого устройства.
3. Подробное описание методики построения модели исследуемого устройства в программе (согласно примеру).
4. Картины полученных полей.
5. Характеристики поля в любых трёх точках, анализ результатов.
6. Выводы.

Лабораторная работа 7. Решение задач электромагнитной дефектоскопии с применением программы ELCUT

Задание. Для исследуемой магнитной цепи (рис. 57) согласно варианту (табл. 9) решить задачи дефектоскопии (задачу магнитостатики), где блок CDHG – материал магнита АЛНИКО+; блок EIJF – материал магнита АЛНИКО-, оценить влияние щели на конструкцию (построить картины полей с ней и без нее) с помощью ELCUT.

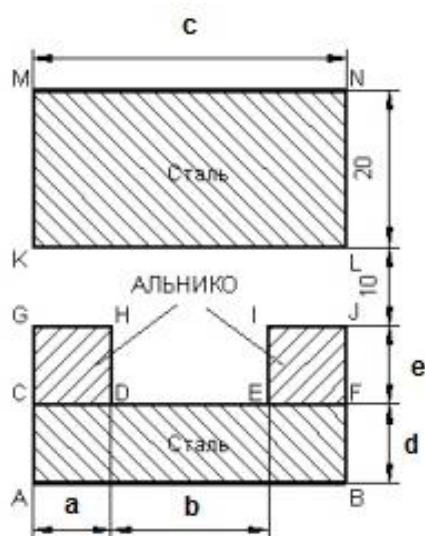


Рисунок 57. Схема магнитной цепи.

Этап 1. Согласно методике построить геометрическую модель конструкции в программе ELCUT.

В данном задании мы наблюдаем, что параметры конструкции кратны 5 или 10 мм. Для упрощения создания объектов можно использовать такой инструмент, как **Сетка привязки**.

В меню **Вид** надо выбрать **Сетка привязки**, далее Вы увидите окно (рис. 58). Устанавливаем указанный шаг сетки.

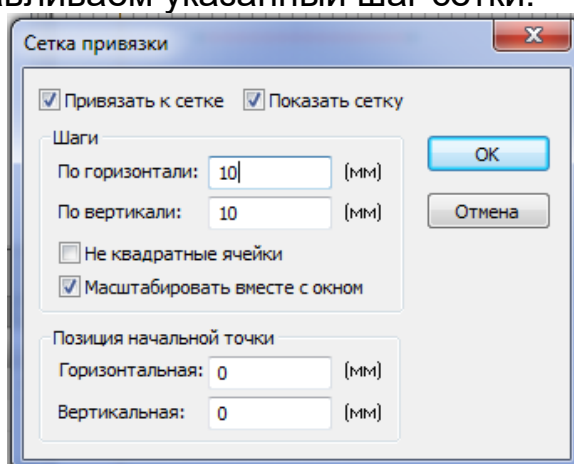


Рисунок 58. Окно сетки привязки.

Этап 2. Введение свойств блоков нелинейных материалов.

В отличие от предыдущей работы конструкция имеет в своем составе нелинейные свойства материалов. Для введения свойств нелинейных блоков в диалоговом окне необходимо пометить **Нелинейный материал**.

После этого появится диалоговое окно для введения кривой намагничивания материала. К примеру, в таблице 10 приведены параметры кривой намагничивания $B(H)$ стали.

Таблица 10. Кривая намагничивания стали.

H (А/м)	400	600	800	1000	1400	2000	3000	4000	6000
B (Тл)	0,73	0,92	1,05	1,15	1,28	1,42	1,52	1,58	1,60

После введения параметров кривой намагничивания получим следующее диалоговое окно (рис. 59).

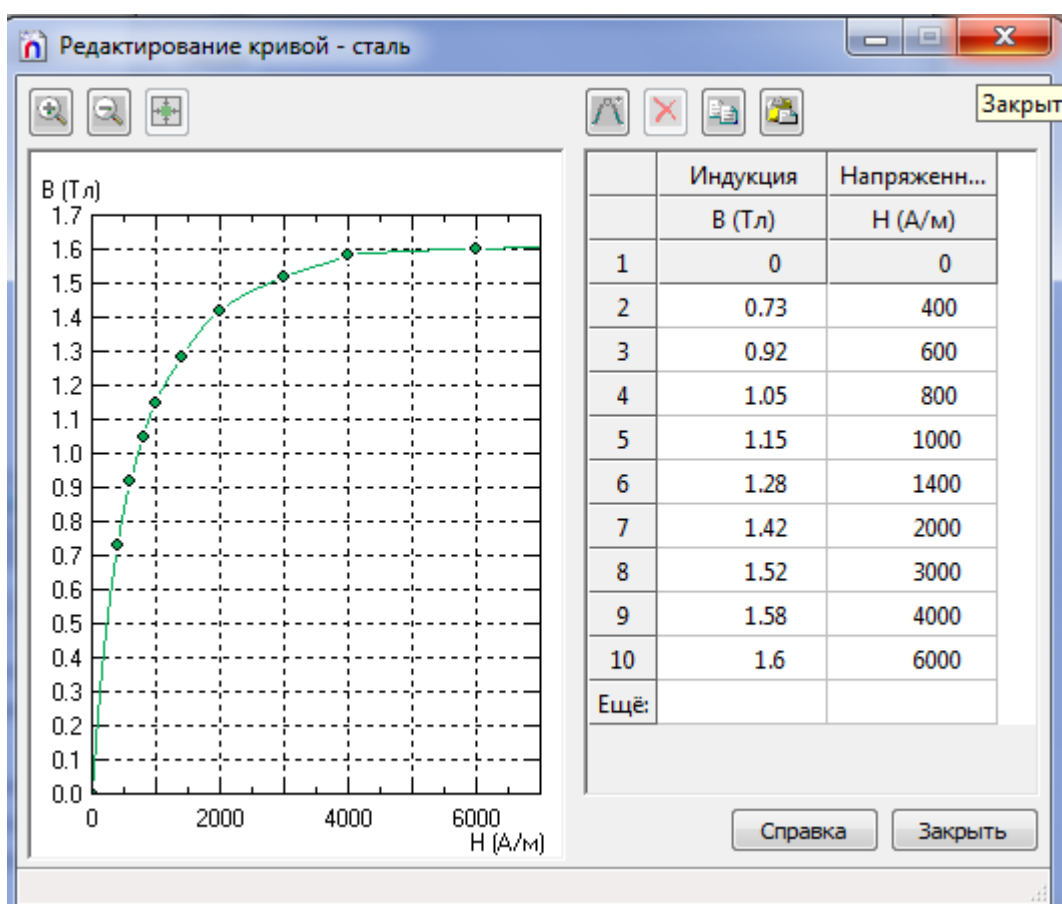


Рисунок 59. Окно заданий свойств нелинейных материалов (сталь).

Этап 3. Введение свойств блоков для постоянного магнита.

Для такого типа блоков кроме кривой размагничивания необходимо задать направление намагничивания (то есть компоненты вектора коэрцитивной силы). Переходим к построению характеристики размагничивания для блока АЛНИКО+.

Привычно выбираем строку блока для ввода свойств, открываем диалоговое окно.

Для активации ввода параметров блока для постоянного магнита необходимо ввести **1** в любую позицию ввода параметра магнитной проницаемости.

Далее вводим в поле **Величина** значение коэрцитивной силы **147218** (первая точка кривой размагничивания задана и имеет координаты (-147218; 0)).

В поле **Направление** вводим **90** (так как направление принято отсчитывать от горизонтальной оси против часовой стрелки), отмечаем Нелинейный материал, вводим остальные значения кривой размагничивания сплава АЛНИКО+ (табл. 11).

Таблица 11. Кривая размагничивания сплава АЛНИКО.

H (А/м)	400	600	800	1000	1400	2000	3000	4000	6000
B (Тл)	0,73	0,92	1,05	1,15	1,28	1,42	1,52	1,58	1,60

Для введения свойств блока АЛНИКО- повторять введение всех параметров кривой размагничивания не нужно. Так как параметры кривой размагничивания блока АЛНИКО- и АЛНИКО+ отличны только направлением, то делаем следующее.

В дереве меток выбираем метку АЛНИКО+, копируем метку и вставляем копируемую метку в дерево меток (она появится там и будет дублировать имя копируемой метки). Далее необходимо переименовать полученную метку на АЛНИКО-, потом в диалоговом окне ввода свойств поменять направление (с 90 на - 90). Таким способом изменится направление коэрцитивной силы, данные кривой размагничивания для блока АЛНИКО- автоматически изменятся.

В отличии задач электростатики, мы вводим только параметры **Экрана** аналогично предыдущей лабораторной работе.

Этап 4. Получение результатов моделирования.

После получения файла задачи, файла геометрии, построения сетки выбираем пункт **Решить задачу** и получаем картину с результатами расчета задачи магнитостатики (привести картины всех полей).

Переходим к расчету интегральных значений. Для этого необходимо обозначить (построить) **контур интегрирования**. С помощью мыши выбираем в меню команду **Контур/Добавить**. Выбираем тип линии контура на панели инструментов **Панель анализа результатов**. Для замыкания контура выполнить команду **Контур/Замкнуть**.

После создания контура нужно перейти к пункту меню **Вид/Интегральные значения**. Теперь в программе появился список рассчитанных интегральных характеристик. Выбрав необходимый параметр и щелкнув по нему, получаем его значение (рис. 60).

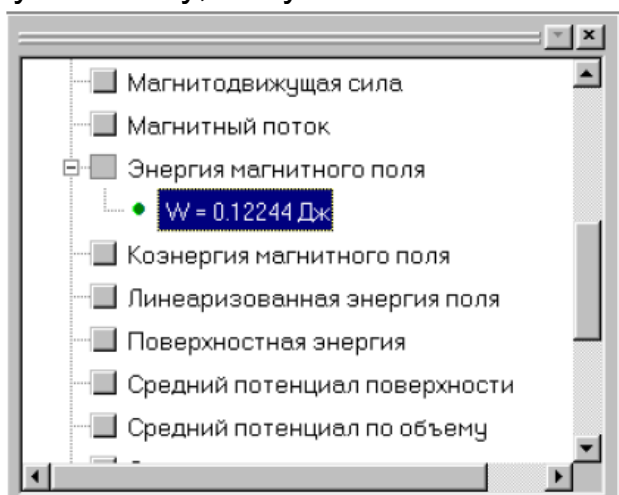


Рисунок 60. Окно со списком характеристик.

Для получения графиков распределения переменных в программе необходимо создать контур построения (графики строятся вдоль линии построения контура, в качестве контура может быть просто отрезок). Когда контур определен, переходим к пункту меню **Вид/График**.

На рисунке 61 представлена полученная зависимость. Для выбора другой переменной для построения ее зависимости, необходимо выбрать команду **Вид/Кривые** и выбрать переменную.

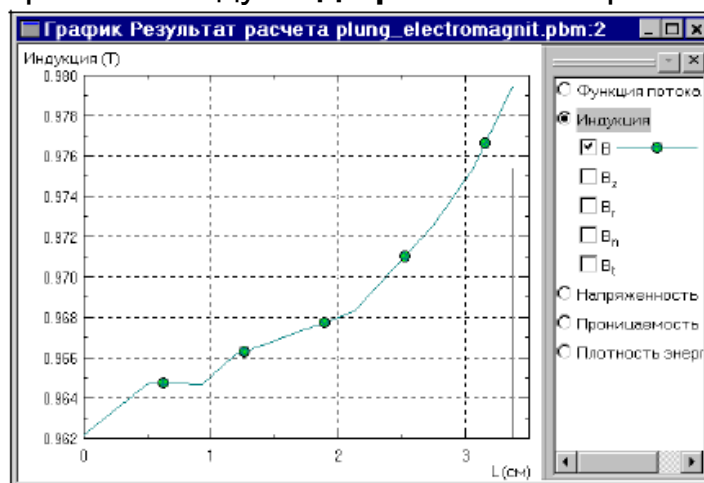


Рисунок 61. Построение зависимости.

Таблица 9. Варианты заданий к рисунку 57.

№ варианта	Краткое описание
1	a=5 мм; b=10 мм; c=40 мм; d=10 мм; e=10 мм.
2	a=10 мм; b=20 мм; c=40 мм; d=15 мм; e=15 мм.
3	a=10 мм; b=5 мм; c=40 мм; d=10 мм; e=15 мм.
4	a=10 мм; b=15 мм; c=50 мм; d=15 мм; e=10 мм.
5	a=10 мм; b=20 мм; c=50 мм; d=15 мм; e=15 мм.
6	a=20 мм; b=5 мм; c=50 мм; d=10 мм; e=10 мм.
7	a=5 мм; b=30 мм; c=60 мм; d=15 мм; e=15 мм.
8	a=30 мм; b=5 мм; c=60 мм; d=10 мм; e=15 мм.
9	a=15 мм; b=15 мм; c=60 мм; d=15 мм; e=15 мм.
10	a=10 мм; b=15 мм; c=70 мм; d=10 мм; e=20 мм.

Содержание отчета

1. Цель работы.
2. Конструктивный эскиз исследуемого устройства.
3. Подробное описание методики построения модели исследуемого устройства в программе.
4. Исследование объекта без щели и со щелью.
5. Построение необходимых зависимостей, полей, анализ результатов.
6. Выводы.

Лабораторная работа 8. Тепловая дефектоскопия жилого помещения с помощью ELCUT

Задание. Для исследуемого жилого помещения (рис. 61) согласно варианту (табл. 12) решить задачу тепловой дефектоскопии (теплопередачи) с помощью ELCUT. Заданные условия: коэффициент теплопроводности кирпича $\lambda=0,67$ Вт/(м·К), дерева $\lambda_x=0,2$ Вт/(м·К), огнеупорной глины $\lambda=1,64$ Вт/(м·К), $\lambda_y=0,35$ Вт/(м·К), температура вне помещения = 243 К, температура камина и газовой конфорки = 500 К. Таблица 13 показывает зависимость коэффициента теплопроводности воздуха от температуры. В помещении находятся отопительные приборы (камин, газовая плита).

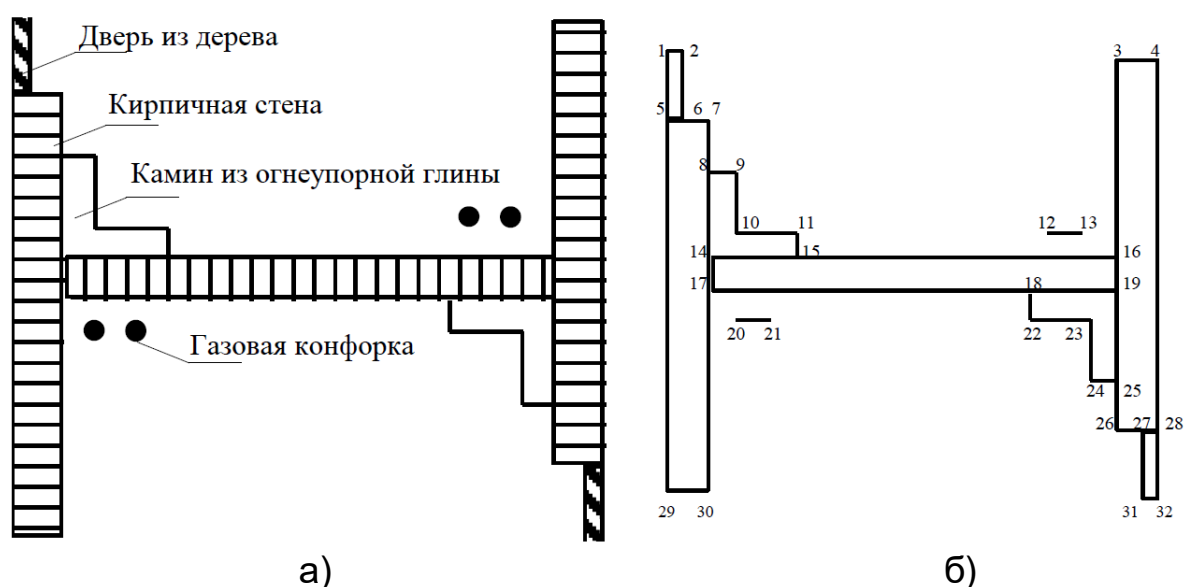


Рисунок 62. Схема жилого помещения.

Этап 1. Согласно методике построить геометрическую модель конструкции в программе ELCUT.

В данном случае тип задачи: нелинейная плоскопараллельная задача теплопередачи, то есть вводим свойства задачи следующим образом: Тип задачи – *Теплопередача стационарная*, Класс модели – *Плоская*.

Изображаем геометрическую модель объекта, схематически она представлена на рисунке 62б.

Аналогично предыдущим работам прописываем свойства блоков, ребер модели объекта.

После решения задачи мы видим результат расчета температурного поля исследуемого помещения (рис. 63).

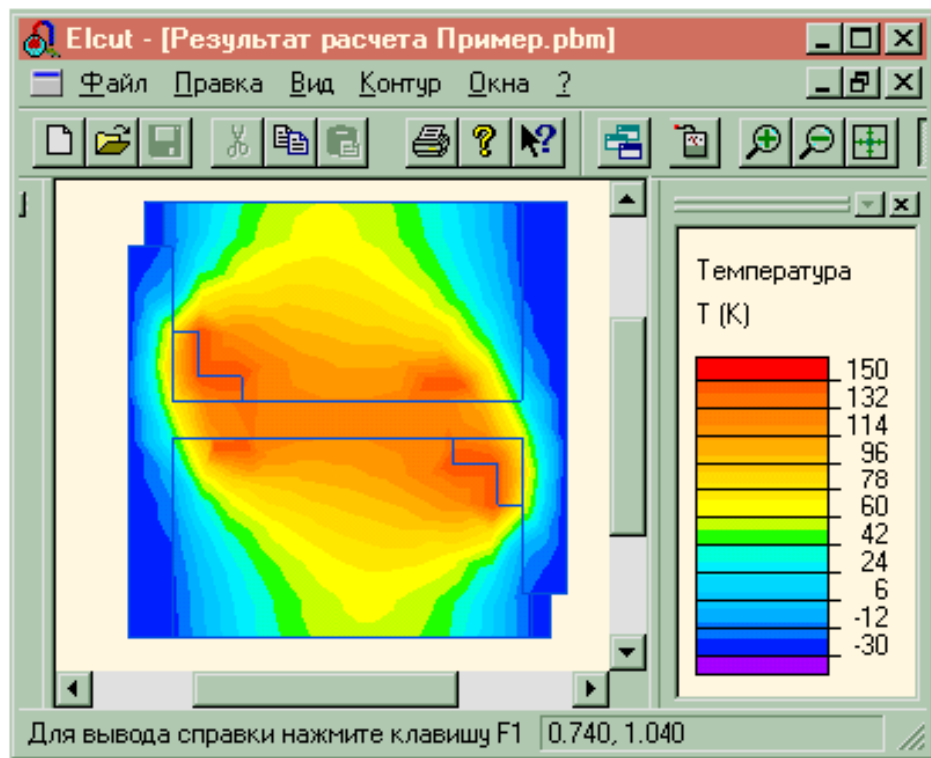


Рисунок 63. Результат решения задачи.

Значение температуры в этой исследуемого объекта можно получить через меню **Вид**→**Локальные значения** (рис. 64).

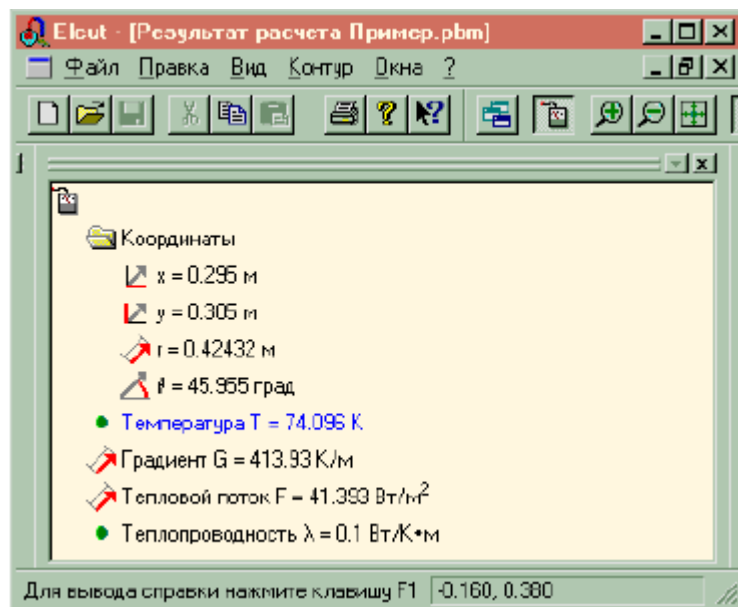


Рисунок 64. Локальные значения температуры.

В качестве дополнительного задания, необходимо определить значения температуры для любых трех исследуемых точек, внести результаты в отчет по работе.

Таблица 12. Варианты заданий.

Вариант	Помещение	
1		<p><i>Условные обозначения:</i></p> <ol style="list-style-type: none"> 1. Наружняя дверь из дерева $\lambda_x=0,2$; $\lambda_y=0,35$ 2. Кирпичная перегородка $\lambda=0,81$ 3. Кирпичная кладка $\lambda=0,81$ 4. Обшивка из ДВП $\lambda=0,72$ 5. Кирпичная перегородка $\lambda=0,81$ 6. Дверь из дерева $\lambda_x=0,2$; $\lambda_y=0,35$ 7. Камин из огнеупорной глины $\lambda=1,04$ 8. Газовая конфорка 9. Прослойка из бетона $\lambda=1,04$
2		<p><i>Условные обозначения:</i></p> <ol style="list-style-type: none"> 1. Обшивка из дерева $\lambda_x=0,2$; $\lambda_y=0,35$ 2. Наружняя дверь из дерева $\lambda_x=0,2$; $\lambda_y=0,35$ 3. Кирпичная кладка $\lambda=0,81$ 4. Дверь из дерева $\lambda_x=0,2$; $\lambda_y=0,35$ 5. Кирпичная перегородка $\lambda=0,81$ 6. Стекло $\lambda=0,96$ 7. Воздушная прослойка 8. Газовая конфорка
3		<p><i>Условные обозначения:</i></p> <ol style="list-style-type: none"> 1. Наружняя дверь из дерева $\lambda_x=0,2$; $\lambda_y=0,35$ 2. Перегородка из дерева $\lambda_x=0,2$; $\lambda_y=0,35$ 3. Штукатурка $\lambda=1,2$ 4. Стекло $\lambda=0,96$ 5. Кирпичная кладка $\lambda=0,81$ 6. Газовая конфорка
4		<p><i>Условные обозначения:</i></p> <ol style="list-style-type: none"> 1. Газовая конфорка 2. Перегородка из дерева $\lambda_x=0,2$; $\lambda_y=0,35$ 3. Штукатурка $\lambda=1,2$ 4. Камин из огнеупорной глины $\lambda=1,04$ 5. Кирпичная кладка $\lambda=0,81$ 6. Наружняя дверь из дерева $\lambda_x=0,2$; $\lambda_y=0,35$

5		<p><i>Условные обозначения</i></p> <ol style="list-style-type: none"> 1. Обшивка из дерева $\lambda_x=0,2$; $\lambda_y=0,35$ 2. Кирпичная перегородка $\lambda=0,81$ 3. Кирпичная кладка $\lambda=0,81$ 4. Прослойка из бетона $\lambda=1,28$ 5. Дверь из дерева $\lambda_x=0,2$; $\lambda_y=0,35$ 6. Штукатурка $\lambda=1,2$ 7. Камин из огнеупорной глины $\lambda=1,04$
6		<p><i>Условные обозначения</i></p> <ol style="list-style-type: none"> 1. Обшивка из дерева $\lambda_x=0,2$; $\lambda_y=0,35$ 2. Камин из огнеупорной глины $\lambda=1,04$ 3. Кирпичная кладка $\lambda=0,81$ 4. Стекло $\lambda=0,96$ 5. Воздушная прослойка 6. Дверь из дерева $\lambda_x=0,2$; $\lambda_y=0,35$ 7. Кирпичная перегородка $\lambda=0,81$ 8. Наружняя дверь из дерева $\lambda_x=0,2$; $\lambda_y=0,35$
7		<p><i>Условные обозначения:</i></p> <ol style="list-style-type: none"> 1. Обшивка из дерева $\lambda_x=0,2$; $\lambda_y=0,35$ 2. Кирпичная кладка $\lambda=0,81$ 3. Перегородка из ДВП $\lambda=0,72$ 4. Камин из огнеупорной глины $\lambda=1,04$ 5. Воздушная прослойка 6. Наружняя дверь из дерева $\lambda_x=0,2$; $\lambda_y=0,35$
8		<p><i>Условные обозначения:</i></p> <ol style="list-style-type: none"> 1. Обшивка из дерева $\lambda_x=0,2$; $\lambda_y=0,35$ 2. Перегородка из ДВП $\lambda=0,72$ 3. Камин из огнеупорной глины $\lambda=1,04$ 4. Кирпичная кладка $\lambda=0,81$ 5. Бетонная прослойка $\lambda=1,04$ 6. Газовая конфорка

9		<p><i>Условные обозначения:</i></p> <ol style="list-style-type: none"> 1. Кирпичная кладка $\lambda=0,81$ 2. Обшивка из ДВП $\lambda=0,72$ 3. Перегородка из дерева $\lambda_x=0,2;$ $\lambda_y=0,35$ 4. Камин из огнеупорной глины $\lambda=1,04$ 5. Газовая конфорка 6. Наружная дверь из дерева $\lambda_x=0,2;$ $\lambda_y=0,35$
10		<p><i>Условные обозначения:</i></p> <ol style="list-style-type: none"> 1. Газовая конфорка 2. Перегородка из ДВП $\lambda=0,72$ 3. Штукатурка $\lambda=0,7$ 4. Кирпичная кладка $\lambda=0,81$ 5. Дверь из дерева $\lambda_x=0,2;$ $\lambda_y=0,35$ 6. Камин из огнеупорной глины $\lambda=1,04$ 7. Наружная дверь из дерева $\lambda_x=0,2;$ $\lambda_y=0,35$

Таблица 13. Зависимость коэффициента теплопроводности воздуха от температуры

T, K	243	253	263	273	283	293	303	313	323	333	343
$\lambda \cdot 10^2, \text{Вт}/(\text{м} \cdot \text{К})$	2.20	2.28	2.36	2.44	2.51	2.59	2.67	2.76	2.83	2.90	2.96
T, K	353	363	373	393	413	433	453	473	523	573	623
$\lambda \cdot 10^2, \text{Вт}/(\text{м} \cdot \text{К})$	3.05	3.13	3.21	3.34	3.49	3.64	3.78	3.93	4.27	4.60	4.91

Содержание отчета

1. Цель работы.
2. Подробное описание методики построения модели исследуемого объекта в программе.
3. Геометрическая модель исследуемого объекта.
4. Температурное исследование объекта, построение необходимых полей, значений температур, анализ результатов.
6. Выводы.

Лабораторная работа 9. Моделирование приборных устройств с помощью программного продукта ELCUT

Задание. Провести анализ возможностей пакета ELCUT для моделирования промышленного оборудования на примере установки индукционного нагрева (рис. 65) согласно варианту (табл. 14). В качестве установки индукционного нагрева используется индукционная тигельная печь.

Устройствами индукционного нагрева (индукционными установками) называют электротермические устройства, предназначенные для нагрева или плавки материалов посредством воздействия на них переменным электромагнитным полем. Под действием переменного поля в нагреваемом материале наводятся вихревые токи (токи Фуко), которые и разогревают материал. Конструктивно индукционная установка обычно представляет собой индуктор (катушку), питаемый от источника переменного тока промышленной или повышенной частоты. Внутри индуктора помещается нагреваемый объект. Индукционный нагрев обладает рядом преимуществ, благодаря которым широко используется в промышленности. Это быстрота нагрева, выделение тепла только в нагреваемом объекте, пониженные затраты энергии, а также возможность автоматического управления процессом нагрева. Нагревательные индукционные установки применяются для подготовки материалов перед ковкой, прокаткой, прессованием, для закалки, отжига, нагрева перед сваркой, для пайки, наплавки, а также для плавки черных и цветных металлов с целью получения материалов особой чистоты, однородности и точности химического состава.

В настоящее время методики проектирования таких установок являются приближенными, основаны на экспериментальных данных, что часто приводит к большим погрешностям в расчетах. Из этого следует, что проверка характеристик любой разработанной установки должна сопровождаться ее моделированием. Для таких целей при моделировании установок, в том числе индукционных, наиболее подходят программы с возможностями расчета полей методом конечных элементов (МКЭ), в том числе программный пакет ELCUT.

Этап 1. Согласно методике построить геометрическую модель конструкции в программе ELCUT и ввести свойства блоков.

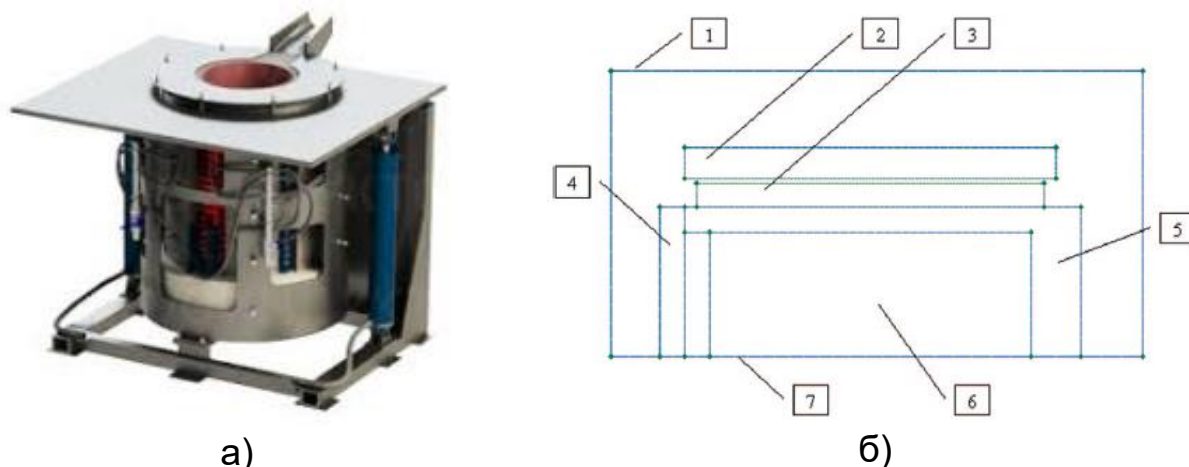


Рисунок 65. Конструкция индукционной печи (а) и геометрическая модель печи (б), где 1 – граница расчетной области; 2 - магнитопровод, 3 - обмотка; 4 - крышка тигля; 5 - тигель; 6 - металл; 7 - ось симметрии.

В отличие от предыдущих работ при создании файла задачи вводим тип задачи **Магнитное поле переменных токов**, класс модели **Осесимметричная**, величину частоты 500 Гц (рис. 66).

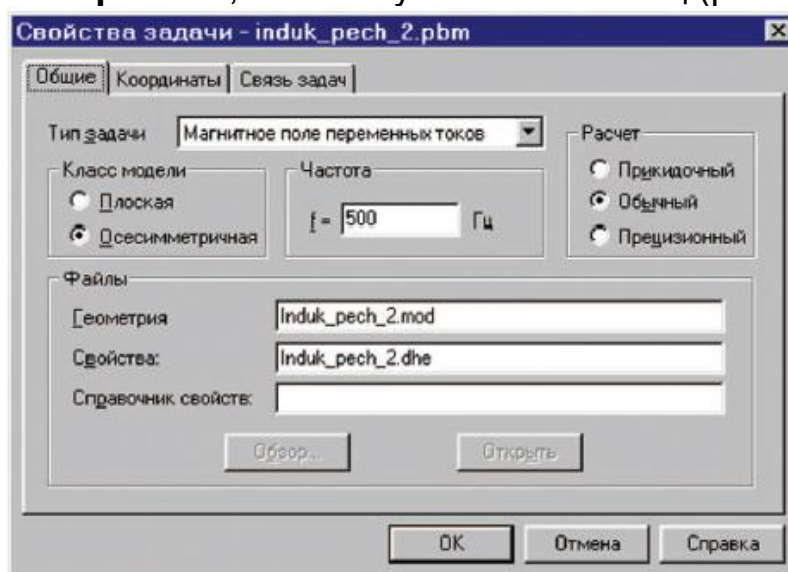


Рисунок 66. Создание файла задачи

Стоит отметить, что программа ELCUT решает задачи двухмерного типа, но если внимательно посмотреть на представленную геометрическую модель печи (рис. 65б) и ее тип, то можно сказать, что исследуемая конструкция является телом вращения, и решаемая задача является трехмерной. Это расширяет возможности использования программы.

При решении представленной задачи существует возможность определения картины распределения мощности тепловыделения в нагреваемом элементе, которая средствами пакета передается затем в подчиненную задачу нестационарной теплопередачи для расчета процесса нагрева. Передача картины распределения мощности тепловыделения в нагреваемом элементе передается в подчиненную путем вкладки **Связь задач** (см. рис. 66).

После создания геометрической модели переходим к заданию свойств конструкции.

На рисунке 67 представлено окно задания электромагнитных свойств расплавляемого материала.

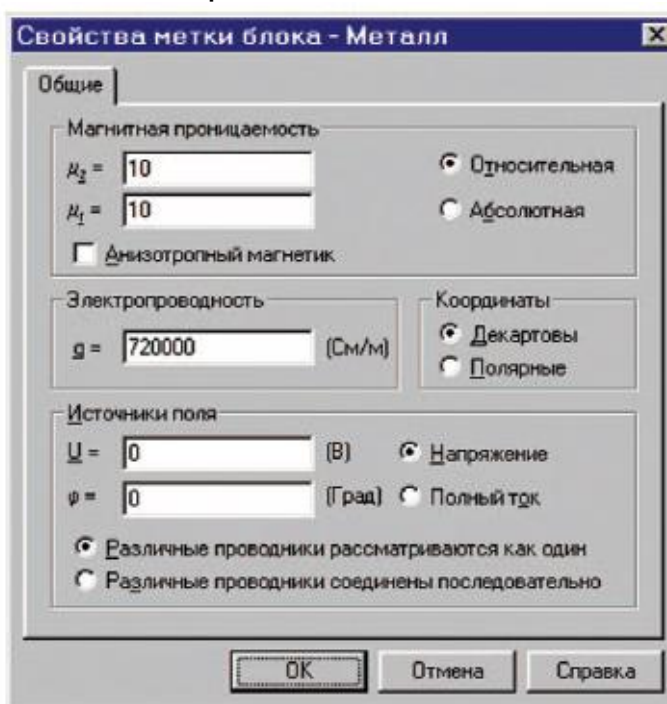


Рисунок 67. Задание свойств блока Металл.

Для тепловых задач также существует возможность задания свойств нелинейного материала (зависимость теплопроводности от температуры). Для задания зависимости необходимо установить флажок «Нелинейный материал» в окне параметров.

Далее переходим к заданию нагрузок (величины токов, мощности тепловыделения и т.д.), они также задаются в окне свойств метки блока. Величина тока в обмотке задается в **Источники поля** (см. рис. 67). В этом же окне можно задать начальную фазу тока в обмотке. Также существует возможность для моделирования многофазной обмотки. Для этого в геометрической модели необходимо создать соответствующие обмотки, присвоив им разные метки, и затем задать разные начальные фазы токов в каждой обмотке.

Далее действуем по вышеописанной методике и анализируем результаты.

В программе имеется также специальный инструмент **Мастер импеданса**, с помощью которого можно определить полное сопротивление обмотки индуктора на заданной частоте. Такой инструмент оказывается весьма полезным, если необходимо определить схему замещения печи с учетом загрузки для выбора источника питания обмотки печи.

При решении задачи нестационарной теплопередачи (динамической тепловой задачи) картина теплового поля находится для каждого заданного момента времени.

По результатам расчета необходимо построить картину поля температур (или другой рассчитанной переменной) для любого момента времени. Получив картину температурного поля в начале процесса нагрева, необходимо построить и привести графики изменения переменных во времени для любой точки модели.

На рисунке 68 в качестве примера показан график изменения температуры металла в точке, близкой к стенке тигля в его средней части.

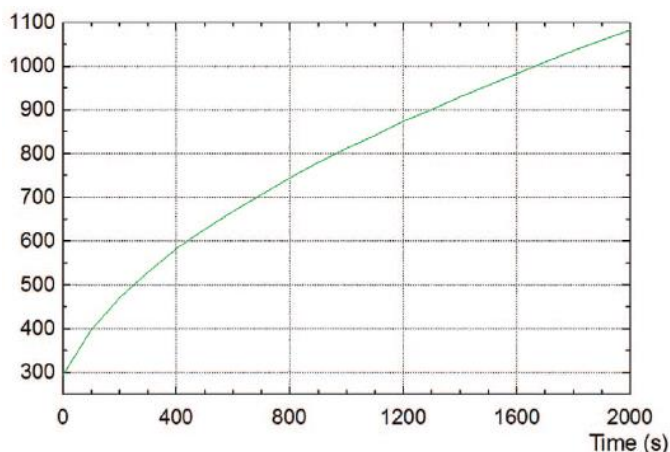


Рисунок 68. Полученная зависимость.

Таблица 14. Варианты заданий к рисунку 65.

Вариант	Относительная магнитная проницаемость материала
1	Расплавляемый металл - Алюминий
2	Расплавляемый металл - Висмут
3	Расплавляемый металл - Железо
4	Расплавляемый металл - Медь
5	Расплавляемый металл - Никель
6	Расплавляемый металл - Платина
7	Расплавляемый металл - Ферритная нержавеющая сталь (отожженная) $1.26 \cdot 10^{-3}$
8	Расплавляемый металл - Фторопласт 4
9	Расплавляемый металл - Пермаллой
10	Расплавляемый металл - Ферритная нержавеющая сталь (отожженная) $2.26 \cdot 10^{-3}$

Содержание отчета

1. Цель работы.
2. Конструктивный эскиз исследуемого устройства.
3. Подробное описание методики построения модели исследуемого устройства в программе.
4. Исследование объекта, построение необходимых зависимостей, полей, анализ результатов.
5. Выводы.

Лабораторная работа 10. Создание программ в среде разработки Turbo C++.

Задание: Ознакомиться с возможностями программного продукта Turbo C++.

Начнем работу с изучения структуры программы на языке C++. В качестве примера рассмотрим случай, когда программа запрашивает значения двух целых переменных *a* и *b*, производит анализ и далее выводит большее из них число. Листинг программы для выполнения этого задания представлен на рисунке 69.

```
01  #include <iostream.h>
02  #include <conio.h>
03  int main(void)
04  {
05      int a,b,max;
06      cout << "a=";          //приглашение ввести значение a
07      cin >> a;              //ввод значения переменной a
08      cout << "b=";          //приглашение ввести значение b
09      cin >> b;              //ввод значения переменной b
10      if(a>b) max=a;         //если a>b то max=a
11      else max=b;           //иначе max=b
12      cout << "max="<<max; //вывод максимального значения
13      getch();              //останов до нажатия клавиши
14      return 0;             //выход из функции
15  }
```

Рисунок 69. Структура программы.

Номера от 01 до 15 являются номерами строк программы, необходимы для упрощения ссылок на описание действия определенных строк, к программированию не относятся.

Строки 01 и 02 реализуют задание **Подключить** (include – включить) **заголовочные** файлы системы (h от header – заголовок) к тексту программы. В этих файлах описаны системные функции и их аргументы.

Компилятор проверяет правильность вызова системных функций, используя эти описания. В нашем случае программа использует системные функции ввода (**cin >>**) и вывода (**cout <<**), описания которых находятся в заголовочном файле **iostream.h**, а также функцию ожидания нажатия какой-либо клавиши (**getch**), описание которой находится в заголовочном файле **conio.h**. Названия заголовочных файлов зачастую образуются от каких-либо аббревиатур английских слов, их полезно научиться понимать, а не

запоминать. В нашем примере: **io** – **input/output** (ввод/вывод), **stream** (поток), **con** (**console** – пульт оператора, т.е. клавиатура и дисплей).

Строка 03 содержит заголовок функции **main**. Функция с таким названием обязана присутствовать в каждой программе на языке C, C++. Именно с нее начинается выполнение программы, она – главная (именно так переводится служебное слово **main**). Предшествующее ей служебное слово **int** (от *integer* – целый) сообщает, что результатом работы функции **main** должно быть целое число. По возвращаемому функцией значению операционная система, запустившая программу **main**, может "сообразить", правильно или неправильно завершилась работа программы. По общепринятому соглашению нулевое значение, возвращаемое функцией **main**, свидетельствует о нормальном завершении работы программы. Служебное слово **void** (дословно – пустота), указанное в круглых скобках, сообщает, что у функции **main** аргументы отсутствуют.

Текст программы (тело функции) заключается в фигурные скобки (строки 04 и 15). В большинстве последующих строк присутствует пояснительный текст на русском языке, следующий после пары символов **//** – это комментарий, на содержание которого система не обращает внимания.

В строке 05 объявлены три переменные с именами **a**, **b** и **max**, которые могут принимать только целочисленные значения (тип – **int**).

Строка 06 является первой строкой программы, которая выполняет следующее действие: она выводит на дисплей сообщение, состоящее из двух символов (**a=**). Текст сообщения заключен в двойные кавычки. Строка 07 организует приостанов работы программы до тех пор, пока пользователь не наберет на клавиатуре какое-либо число и нажмет клавишу **Enter**. Поступившее значение будет хорошо воспринято, если оно целое, и направлено в переменную **a**. Точно таким же образом в строках 08 и 09 будет организован ввод значения числовой переменной **b**.

В строке 10 сравниваются (**if** – если) текущие значения переменных **a** и **b**. Если проверяемое условие выполнено, т.е. значение переменной **a** больше, то оно присваивается переменной **max** – выполняется действие, записанное после проверки условия. В противном случае (**else** – иначе) в переменную **max** заносится значение **b**.

Строка 12 выводит на дисплей два сообщения – текстовое (**max=**) и числовое (значение переменной **max**).

Обращение к функции **getch** (строка 13) приводит к задержке на экране сообщения программы до тех пор, пока пользователь не нажмет какую-либо клавишу (**getch** – от *get character*, дай символ).

Последняя выполняемая строка с номером 14 возвращает управление операционной системе (return – вернуться) и выдает в качестве значения функции нулевой результат.

Обратите внимание!!! Если программа обращается к каким-либо системным функциям, то в первых ее строках обязательно должно стоять указание о подключении соответствующих заголовочных файлов. Программа может содержать более, чем одну функцию, но среди них обязательно должна присутствовать функция с именем main. Каждая строка программы, содержащая какое-либо объявление или выполняемое действие, оканчивается точкой с запятой. Тело функции обязательно заключается в фигурные скобки (в Паскале аналогичные функции выполняли операторные скобки – begin и end).

После запуска программы Turbo C++ в среде Windows на экране появляется окно интегрированной среды, представленное на рисунке 70.

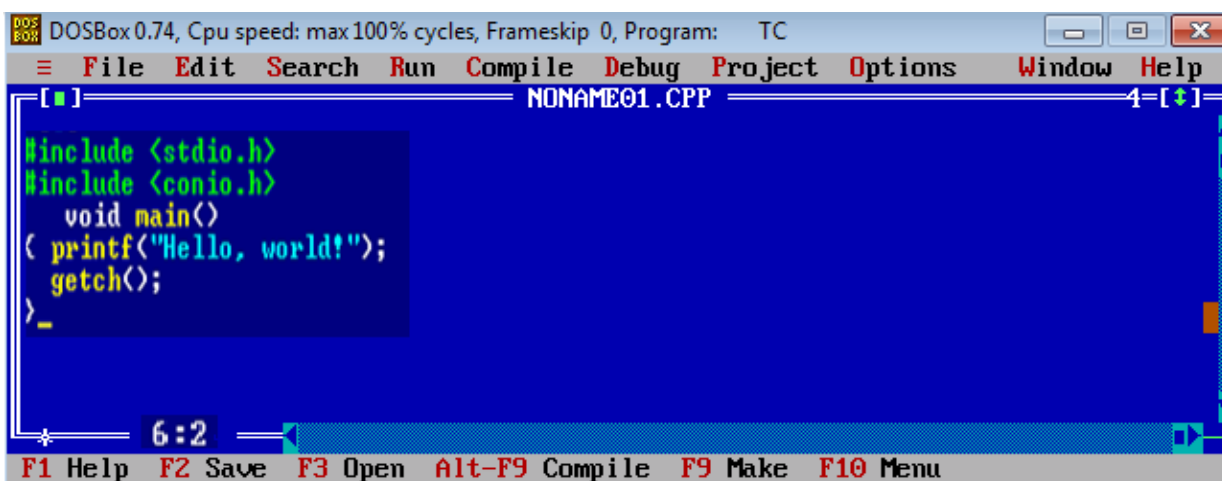


Рисунок 70. Окно интегрированной среды Turbo C++.

В верхней строке располагается главное меню, где

File – общение с файловой подсистемой;

Edit – ввод и редактирование исходной программы;

Search – поиск в тексте исходной программы;

Run – запуск программы в автоматическом или пошаговом режиме;

Compile – компиляция исходной программы;

Debug – отладка программы;

Project – управление проектом сборки программы из нескольких модулей;

Options – настройка параметров интегрированной среды;

Window – управление дополнительными окнами системы программирования;

Help – обращение к файлам помощи.

Далее более подробно рассмотрим команды некоторых подменю, которые чаще всего используются на стадии начального знакомства с этой системой программирования.

Команды меню **File** (рис. 71) используются при наборе новой программы (команда **New**), при запоминании в файле на диске набранной или измененной программы (команды **Save** и **Save as**), при вызове ранее сохраненной программы (команда **Open**). Последняя команда этого меню (**Quit**) выполняется при выходе из интегрированной среды.

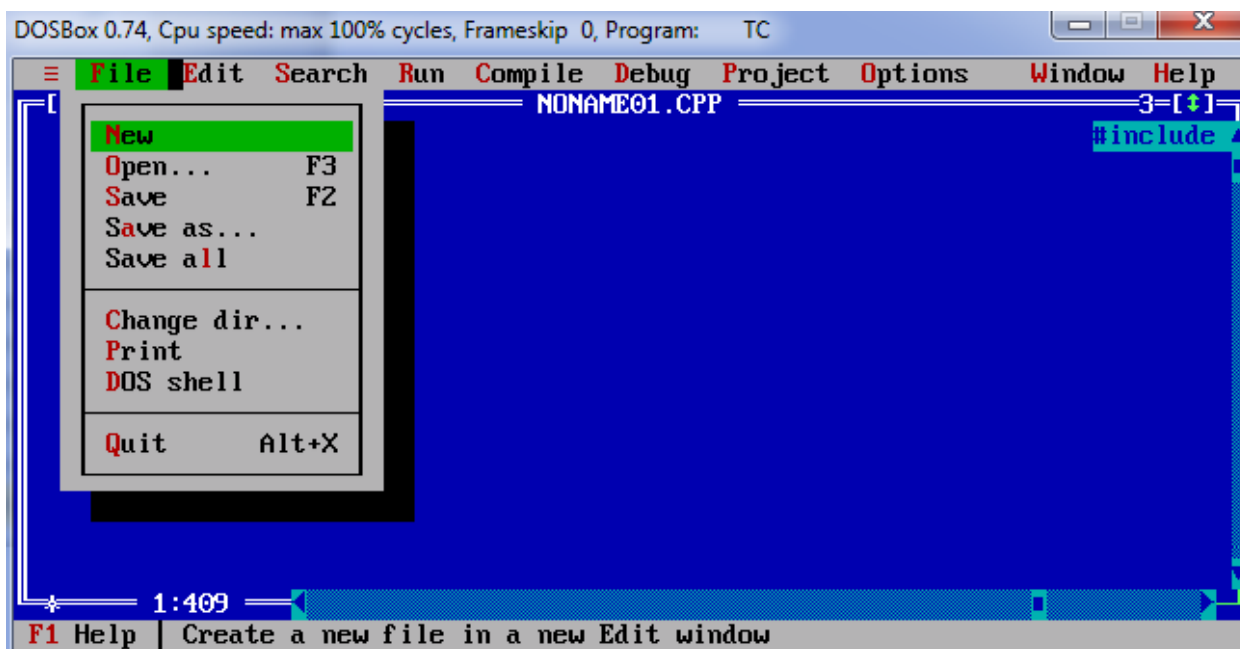


Рисунок 71. Команды меню **File**.

Вход в меню **File** происходит либо после щелчка мышью по заголовку **File**, либо после набора клавишной комбинации **Alt+F**. Выполнение наиболее употребительных команд также продублировано нажатием одной функциональной клавиши или соответствующей клавишной комбинации. Эквивалентные клавишные команды указаны справа.

Переход к той или иной команде выбранного меню сопровождается появлением в нижней строке подсказки. На рисунке 55 выделена команда **New**, а в строке подсказки находится сообщение – "Создание нового файла в новом окне редактирования".

До тех пор, пока при сохранении набранной программы соответствующему дисковому файлу не присвоено индивидуальное имя, вновь набираемая программа выступает под именем **NONAMEnn.CPP**, т.е. программа "безымянная" (здесь nn – порядковый номер безымянной программы, созданной в течение одного сеанса).

Процесс набора текста исходной программы ничем не отличается от технологии работы в любом текстовом редакторе. Разве что клавишные команды, связанные с выделением фрагментов текста, их копированием, удалением или вставкой, специфичны для каждой системы программирования. К ним довольно быстро привыкают при продолжительной работе в той или иной среде. На первых порах вы будете заглядывать в команды меню **Edit**.

Одной из наиболее часто используемых групп команд являются строки меню **Run** (рис. 72). По команде **Run** производится попытка запуска на выполнение программы, находящейся в текущем окне редактирования. При этом новая программа сначала компилируется, потом редактируются ее связи с другими модулями, затем полученный исполняемый модуль загружается в оперативную память и начинает выполняться. Если текущая программа ранее компилировалась и на диске существует соответствующий исполняемый модуль, то повторное выполнение начальных этапов не происходит. Любые ошибки, обнаруженные на стадии компиляции, выдаются в окне сообщений с указанием строки исходной программы, при обработке которой возникла исключительная ситуация. Аналогичные ошибки, обнаруженные редактором связей, выводятся в окно сообщений и процесс исполнения прерывается. Настоятельно рекомендуется запускать программу только после сохранения ее исходного текста на диске. Дело в том, что при некоторых зависаниях системы или работающей программы продолжение сеанса возможно только после перезагрузки компьютера. А в этом случае содержимое поля редактирования будет потеряно, и текст программы придется набирать заново.

Во время выполнения программы ее работа может быть прервана системой, обнаружившей исключительную ситуацию (деление на нуль, переполнение, и т.д.), или пользователем. В ряде случаев для продолжения работы необходимо восстановить первоначальное состояние программы. Для этой цели используется команда **Program reset**.

Командой **Go to cursor** пользуются для автоматического запуска программы с остановом в той строке исходного текста, где находится курсор. Обычно такой режим используется при отладке программы. Также для отладочных целей используется пошаговое выполнение программы. В этом режиме очередное нажатие функциональной клавиши **F7** или **F8** приводит к останову после выполнения очередной строки исходного текста. Разница между этими двумя режимами заключается в исполнении строк, содержащих вызов пользовательской функции. Нажатие **F7** приводит к тому, что пошаговое исполнение сохраняется и в вызываемой функции. В

отличие от этого нажатие **F8** приводит к автоматическому выполнению вызываемой функции и останову после возврата из нее.

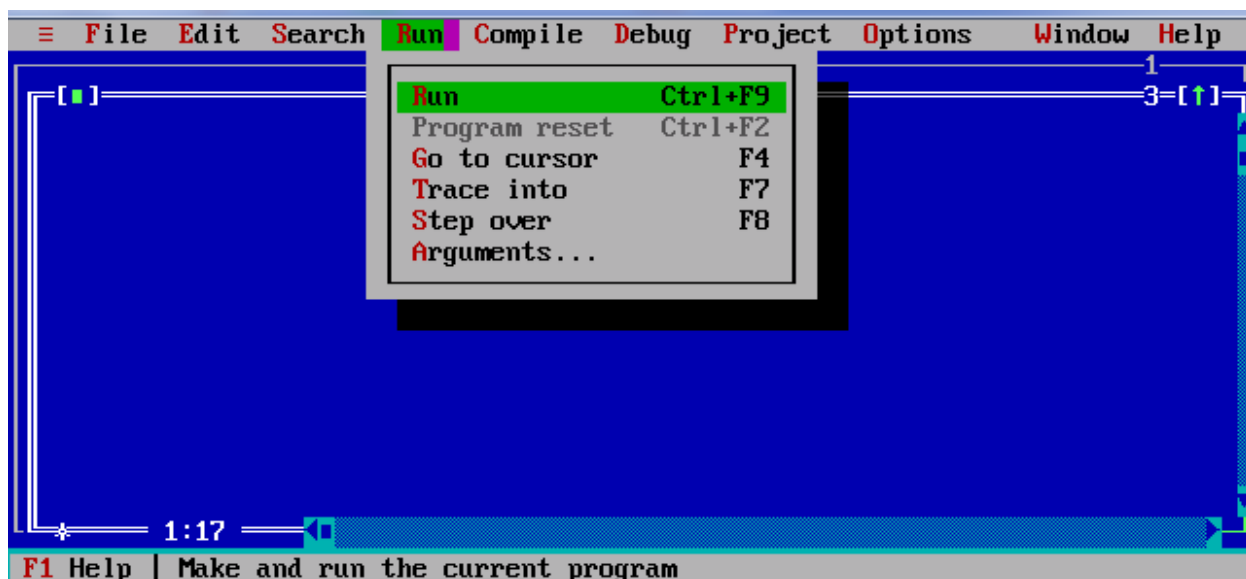


Рисунок 72. Команда меню **Run**.

Очень важно уметь пользоваться справочной системой интегрированной среды – командами меню **Help** (рис. 73).

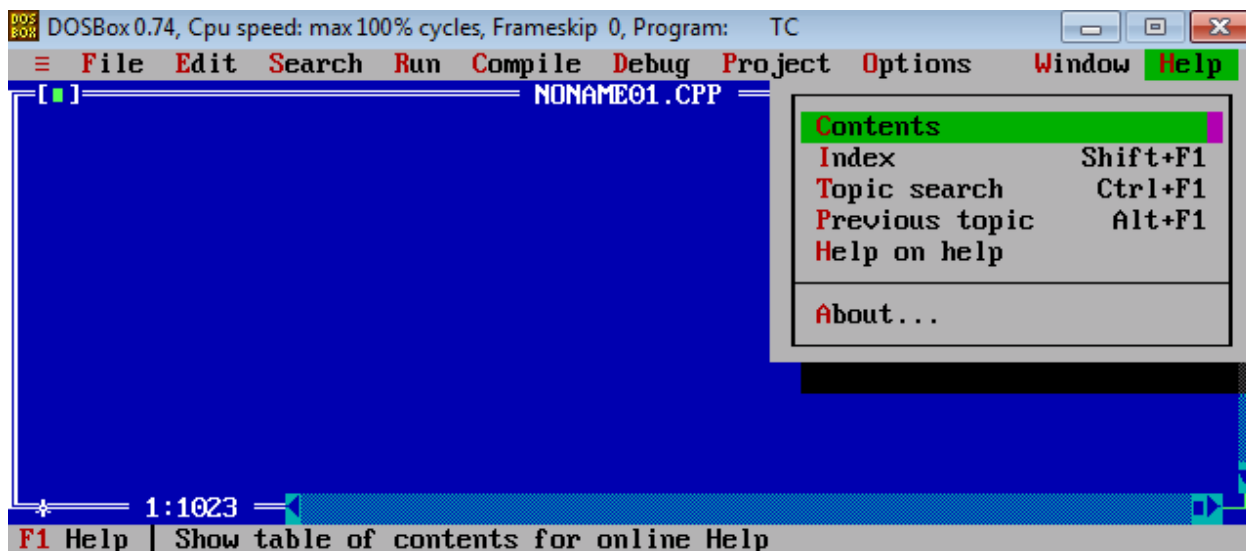


Рисунок 73. Команды меню **Help**.

Справочная система предназначена для работы в двух режимах. Первый из них, называемый контекстной помощью, срабатывает следующим образом. Курсор мыши подводится к интересующему вас служебному слову или наименованию системной функции и нажимается клавиша **F1**. Второй способ связан с использованием команд меню **Help** – **Contents** (Содержание) и **Index** (Указатель терминов, упорядоченных по алфавиту).

В меню **Contents** на экране (рис. 74) появляются названия справочных разделов:

- **How to Use Help** – Как использовать помощь;
- **Menus and Hot Keys** – Меню и "горячие" клавиши (клавишные команды);
- **Editor Commands** – Команды редактора;
- **Assembler (built-in)** – Встроенный ассемблер;
- **Command Line** – Использование командной строки;
- **Container Classes** – Классы-контейнеры;
- **Error Messages** – Сообщения об ошибках;
- **Functions** – Функции;
- **Header Files** – Заголовочные файлы;
- **IOStream Classes** – Поточковые классы ввода/вывода;
- **Turbo C++ Language** – Язык C++ среды Turbo;
- **Utilities** – Сервисные программы.

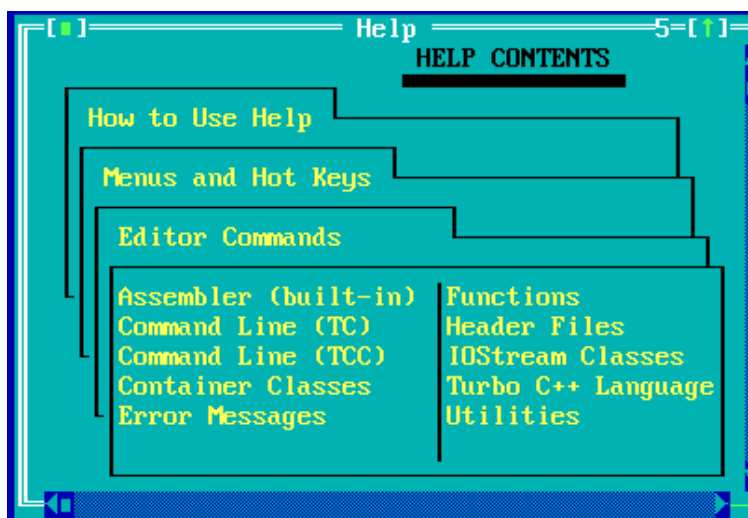


Рисунок 74. Разделы файла помощи системы Turbo C++.

По команде **Index** на экране появляется упорядоченный по алфавиту список (рис. 73), в котором можно выбрать интересующий вас термин и перейти к соответствующему кадру помощи.

Задание 1: Привести листинг программы для сравнения двух результатов $(a+b)$ и $(a \cdot b)$.

Задание 2: Привести листинг программы для перевода значений температуры в градусах по Фаренгейту в значения по Цельсию по формуле (24):

$$C = \frac{5}{9} (F - 32), \quad (24)$$

где C – температура по Цельсию, а F – температура по Фаренгейту.

Задание 3: Привести листинг программы для расчета объема конуса по формуле (25):

$$S = \frac{1}{3} \times \pi \times r^2 \times h, \quad (25)$$

где r - радиус основания, h - высота конуса.

Содержание отчета

1. Цель работы.
2. Листинг программ согласно заданиям.
3. Подробное описание методики написания программ.

Лабораторная работа 11. Построение блок-схем вычислительных процессов

Изучаемая среда многооконна, а это значит, что на экране может быть расположено одновременно несколько независимых окон, в том числе окон с текстами различных программ. Новое окно редактирования для ввода текста программы создается всякий раз командой **File**⇒**New**. Кроме окон редактирования могут быть открыты различные служебные окна (рис. 75):

Size/Move или **Ctrl+F5** переводит окно в режим перемещения и изменения размеров. В этом режиме в строке состояния выводится:

↑↓← → Move Shift-↑↓← → Resize,

то есть клавишами со **стрелками** окно можно перемещать, а при нажатой клавише **Shift** стрелками будут изменяться размеры. Отмена этого режима производится клавишей **Enter** или **Esc**;

Zoom или **F5** увеличивает размеры окна до максимального размера, повторное ее выполнение возвращает окно к прежнему размеру;

Tile распределяет равномерно пространство экрана между окнами, чтобы все они были видны;

Cascade располагает несколько окон одного типа друга над другом, так что видимым будет только верхнее окно, а у остальных будут видны только заголовки;

Переход между окнами обеспечивает **Next** или **F6**. Каждое нажатие **F6** активизирует следующее окно;

List all... выводит список всех открытых окон. В нем можно выбрать любое окно и по кнопке ОК перейти в него;

Message – окно сообщений, в которое выводятся сообщения об ошибках этапа компиляции или редактирования связей;

Output – окно для вывода результатов работы программы;

Watch – окно просмотра, в котором показываются текущие значения переменных выполняемой программы или значения выражений;

Project – окно для отображения состава проекта (проекты используются, когда исходный программный код размещен в нескольких файлах).

Project notes открывает окно примечаний к проекту. Здесь можно писать любые заметки, поясняющие смысл проекта.

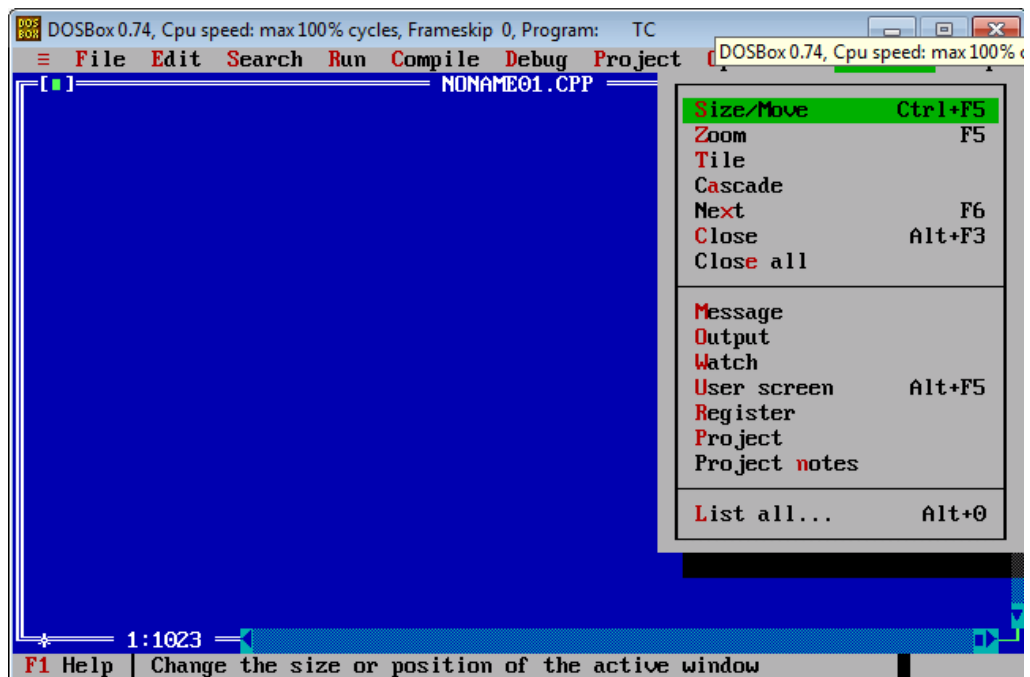


Рисунок 75. Служебные окна.

Только одно окно является активным в любой момент времени. Активное окно выделяется рамкой из двойных линий, неактивное окно ограничивается рамкой из одинарных линий.

На рисунке 76 показано типовое окно и его элементы управления.

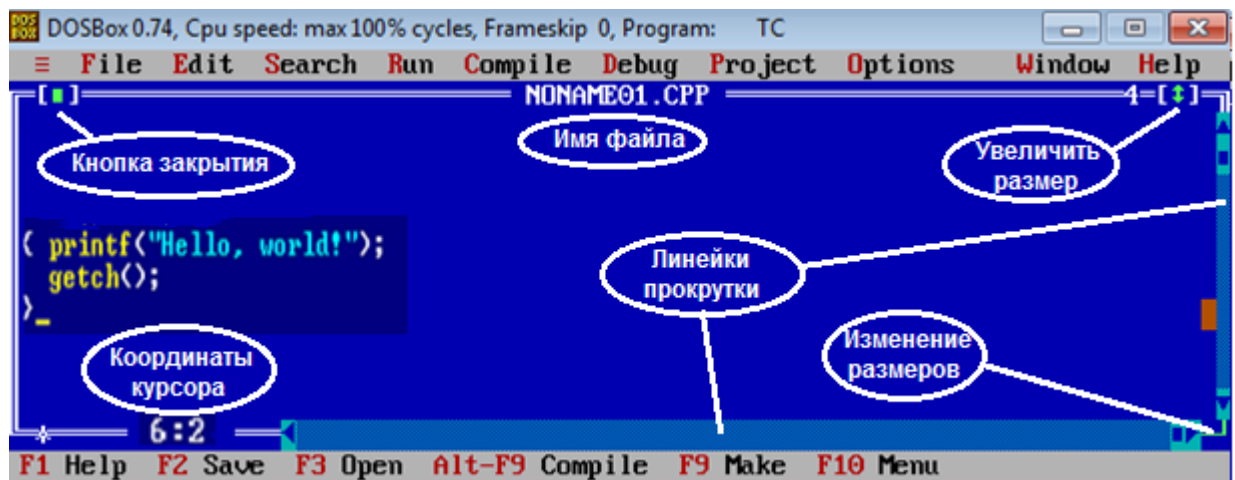


Рисунок 76. Типовое окно и его элементы управления в системе Turbo C++.

Переходим к настройке среды перед созданием программы. Для этого среда позволяет задавать настроечные параметры в специальных диалоговых окнах. Для этого выполняем последовательность команд **Options**⇒**Directories** (рис. 77).

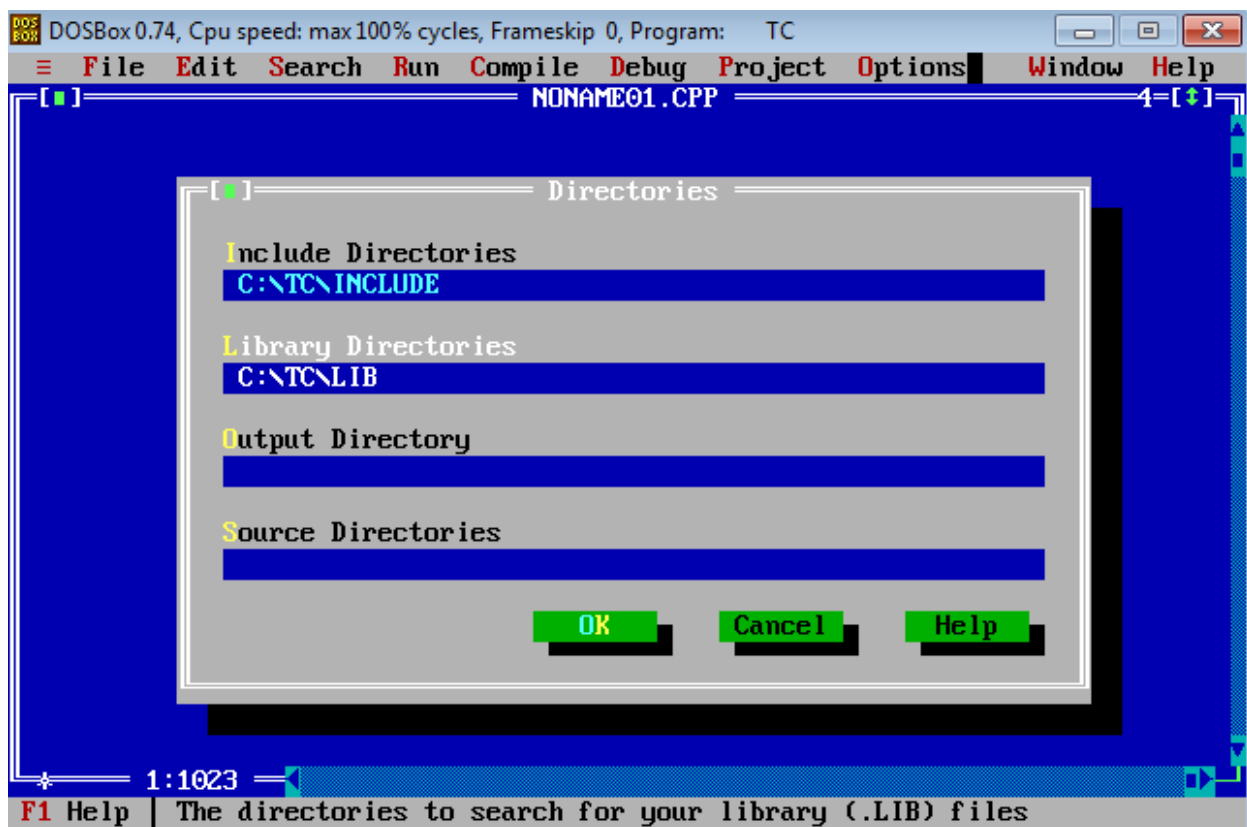


Рисунок 77. Настройка параметров в системе Turbo C++.

Обратите внимание!!! Нужно сделать две обязательные настройки.

В поле **Include Directories** следует ввести полное имя папки, в которой размещены заголовочные файлы стандартных библиотек. Эти папки просматривает препроцессор, в поисках файлов, вставляемых в обрабатываемый файл директивой препроцессора `#include`.

В поле **Library Directories** указывается полное имя папки, где размещены сами стандартные библиотеки. Ее просматривает редактор связей при поиске кода библиотечных функций. Если необходимо указать несколько папок, их имена разделяются точкой с запятой.

В поле **Output Directory** можно указать папку, в которую будут помещаться создаваемые `.obj` и `.exe` файлы. В поле **Source Directories** можно указать папки, где отладчик будет просматривать исходный код библиотек, которые не включены в проект. Эти настройки не являются обязательными.

Если будет необходимо использовать графический режим работы дисплея, то необходимо провести настройку согласно рисунку 78. Для этого требуется выполнить **Options**⇒**Linker**⇒**Libraries**, и в директории установить показанные позиции.

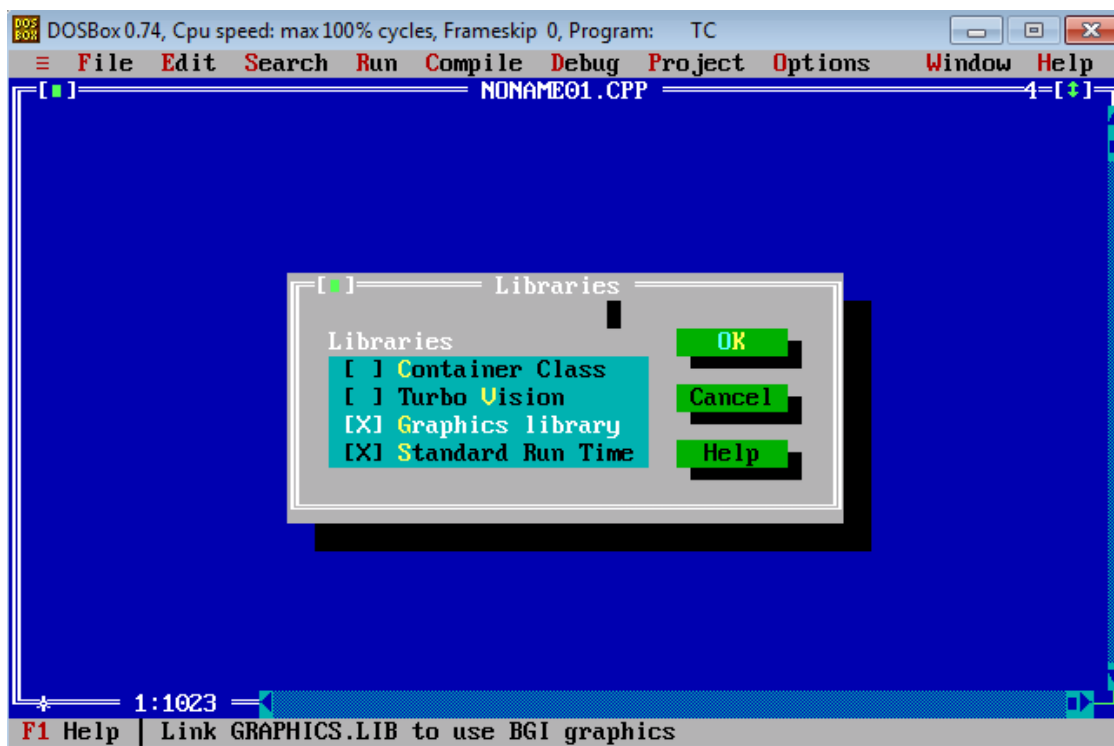


Рисунок 78. Активизация графического режима.

Работа в текстовом редакторе по созданию текста программ производится обычными способами. Выделим некоторые полезные команды.

Завершения ввода каждой строки производится нажатием клавиши **Enter**.

Отмена неправильно введенного материала производится командой меню **Edit, Undo** или комбинацией клавиш **Alt+ BackSpace**.

Весьма ускоряет процесс ввода текста умение работать с блоками.

Для выделения блока текста нужно держать нажатой клавишу **Shift** и нажимать клавиши со стрелками. Блок текста копируется в буфер обмена командой меню **Edit, Copy** или клавишами **Ctrl+Insert**.

Содержимое буфера обмена вставляется в точку расположения курсора командой **Edit, Paste** или клавишами **Shift+Insert**. При этом содержимое буфера обмена не исчезает и его можно снова вставить в другое место.

Команда меню **Edit, Cut** или клавишная комбинация **Shift+Delete** вырезает выделенный фрагмент из текста и вставляет его в буфер обмена.

Полезно запомнить клавишную команду **Ctrl+Y**, которая удаляет из текста текущую строку.

Решение любой задачи на ЭВМ происходит в несколько этапов: постановка задачи; конструирование алгоритма решения задачи; составление программы по разработанному алгоритму; ввод в ЭВМ программы и исходных данных; отладка и тестирование программы; получение решения и анализ результатов.

Алгоритм – конечная последовательность точно определенных действий, приводящих к однозначному решению поставленной задачи. Главная особенность любого алгоритма – формальное исполнение, позволяющее выполнять заданные действия (команды) не только человеку, но и различным техническим устройствам (исполнителям). Процесс составления алгоритма называется алгоритмизацией.

Алгоритмы могут быть заданы: словесно, таблично, графически (с помощью блок-схем). Словесное задание описывает алгоритм с помощью слов и предложений. Табличное задание служит для представления алгоритма в форме таблиц и расчетных формул. Графическое задание, или блок-схема, – способ представления алгоритма с помощью геометрических фигур, называемых блоками. Последовательность блоков и соединительных линий образуют блок-схему. Описание алгоритмов с помощью блок-схем является наиболее наглядным и распространенным способом задания алгоритмов. Блок-схемы располагаются сверху вниз. Линии соединения отдельных блоков показывают направление процесса обработки в схеме. Каждое такое направление называется ветвью. Алгоритм независимо от его структуры всегда имеет по одному блоку «Начало» и «Конец». Его ветви должны в конце сойтись, и по какой бы ветви не было бы начато движение, оно всегда должно привести к блоку «Конец». При задании алгоритма с помощью блок-схемы используются строго определенные блоки. Основные типы блоков приведены в таблице 15. Следует отметить, что все блоки нумеруются. В этом случае номера проставляются вверху слева от блока (блоки «Начало», «Конец» и соединительные блоки не нумеруются). Стрелки на соединяющих линиях обычно не ставят при направлении сверху вниз и слева направо; если направление противоположное, то его показывают стрелкой на линии.

Пример: Составить алгоритм вычисления $z = \phi(y)$, $y = f(x)$, где ϕ , f – известные функции, при заданном значении переменной x (рис. 79).

Знак «←» означает: внести значение переменной в ячейку памяти с определенным именем (операция присваивания).

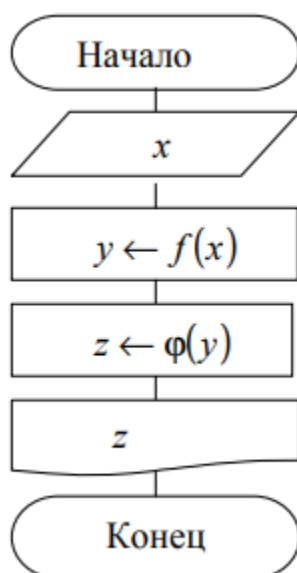
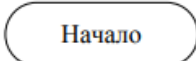


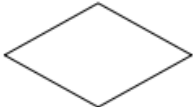







Рисунок 79. Алгоритм решения.

Таблица 15. Основные типы блоков

 Начало  Конец	<i>Начало и конец алгоритма (для функций «Вход», «Выход»)</i>
	<i>Блок обработки. Внутри блока записываются формулы, обозначения и функции</i>
	<i>Блок условия. Внутри блока записываются условия выбора направления действия алгоритма</i>
	<i>Блок предопределенного процесса (функция/подпрограмма)</i>
	<i>Блок ввода информации</i>
	<i>Блок цикла с известным количеством повторений</i>
	<i>Блок вывода информации на печатающее устройство</i>
	<i>Соединительный блок</i>

Пример: Составить блок-схему нахождения корней квадратного уравнения. Коэффициенты квадратного уравнения ввести с клавиатуры (рис. 80).

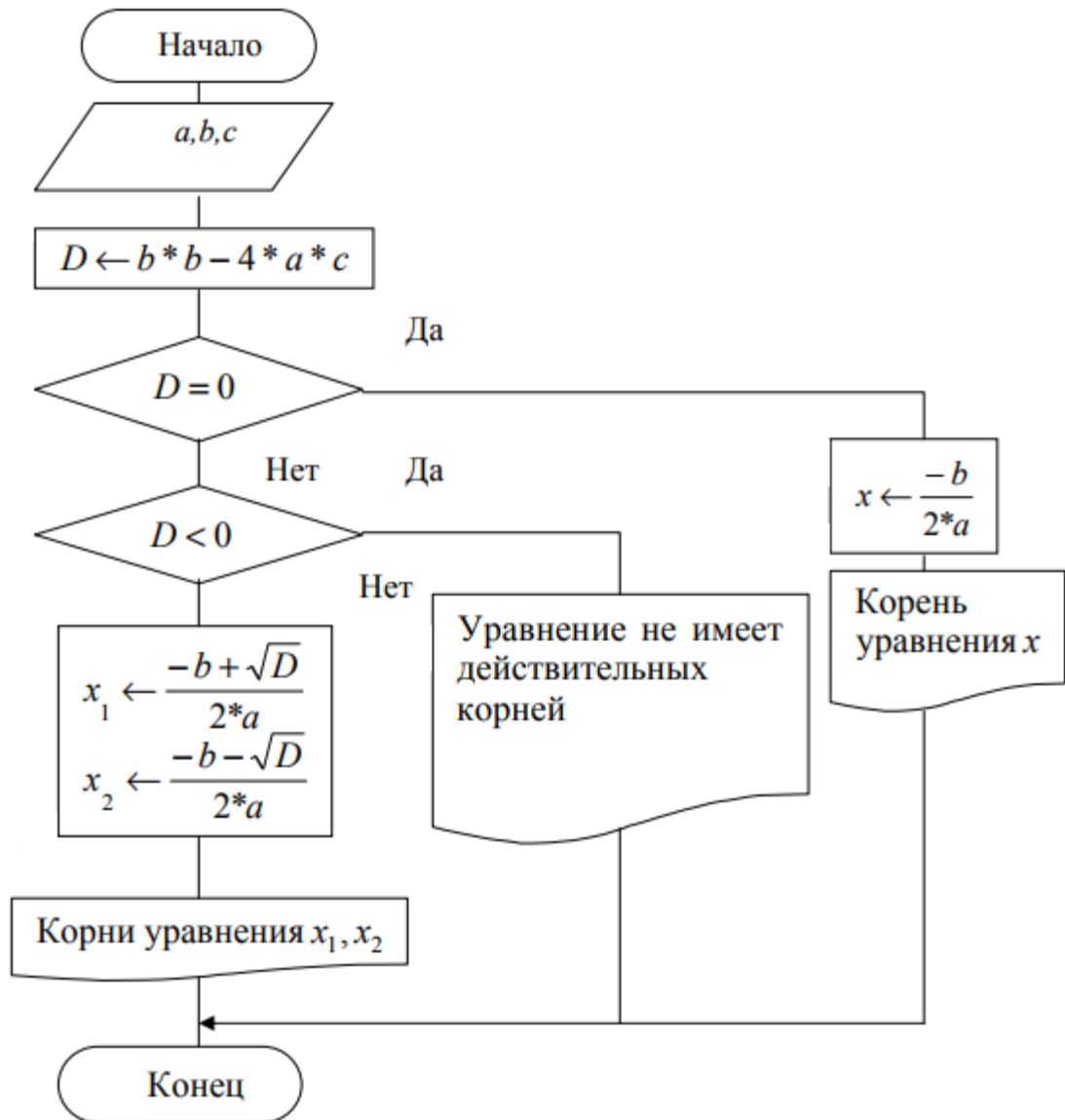


Рисунок 80. Алгоритм решения.

Задание: Построить алгоритмы согласно варианту (табл. 16).

Таблица 16. Варианты заданий.

Вариант	Задание
1	Даны x, y . Получить $\frac{ x - y }{1 + xy }$
2	Даны два числа. Найти среднее арифметическое кубов этих чисел и среднее геометрическое модулей этих чисел.
3	Вычислить расстояние между двумя точками с данными координатами.

4	Найти наименьшее из трех данных чисел.
5	Даны три числа, являющиеся длинами сторон треугольника. Определить тип треугольника (равносторонний, равнобедренный, разносторонний).
6	Дано x . Вычислить y , если: $y = \begin{cases} x^2, & \text{при } -2 \leq x \leq 2, \\ 4, & \text{при } x < -2 \text{ и } x > 2 \end{cases}$
7	Дано x . Вычислить y , если: $y = \begin{cases} 0, & \text{при } x \leq 0 \\ x, & \text{при } 0 < x \leq 1, \\ x^4, & \text{при } x > 1 \end{cases}$
8	Дано x . Вычислить y , если: $y = \begin{cases} x^2 + 4x + 5, & \text{при } x \leq 2, \\ \frac{1}{x^2 + 4x + 5}, & \text{при } x > 2 \end{cases}$
9	Дано x . Вычислить y , если: $y = \begin{cases} 0, & \text{при } x \leq 0 \\ x^2 - x, & \text{при } 0 < x \leq 1, \\ x^2 - \sin \pi x^2 - 1, & \text{при } x > 1 \end{cases}$
10	Даны x, y, z . Найти: $\max\{x + y + z, xyz\} + 3$

Содержание отчета

1. Цель работы.
2. Подробное описание методики построения алгоритма программы согласно варианту.
3. Выводы.

Лабораторная работа 12. Программирование циклических алгоритмов в C++

Задание: выполнить программирование на языке C++.

Выделяют три типа операторов цикла: цикл с предусловием, цикл с постусловием и цикл с параметром. Ниже приведены форматы приведенных типов операторов (табл. 17).

Таблица 17. Форматы типов операторов цикла.

Тип оператора	Формат
Цикл с предусловием	<code>while (выражение) оператор;</code>
Цикл с постусловием	<code>do оператор while (выражение);</code>
Цикл с параметром	<code>for (выражение_1; выражение_2; выражение_3) оператор;</code>

Цикл с предусловием повторяет свое выполнение, пока значение выражения отлично от нуля, т. е. заключенное в нем условие цикла истинно.

Пример программы с оператором такого типа приведен ниже.

```
// Программа вычисления факториала
#include <iostream.h>
void main() {
    long int F; int i,N;
    cout<<"N="; cin>>N;
    F=i=1;
    while(i<=N) F*=i++;
    cout<<"\n"<<N<<"!="<<F;
}
```

Цикл с постусловием выполняется до тех пор, пока выражение отлично от нуля, т.е. заключенное в нем условие цикла истинно. Выход из цикла происходит после того, как значение выражения станет ложным, иными словами равным нулю.

Пример программы с оператором такого типа приведен ниже.

```

#include <iostream.h>
void main() {
long int F; int i,N;
cout<<"N="; cin>>N;
F=i=1;
do F*=i++; while(i<=N);
cout<<"\n"<<N<<"!="<<F;
}

```

Цикл с предусловием имеет тело цикла, которое может быть простым или составным. В последнем случае используют фигурные скобки.

Согласно формату этого оператора (табл. 1) в составе имеются 3 выражения. *Выражение 1* выполняется только один раз в начале цикла. Обычно оно определяет начальное значение параметра цикла (инициализирует параметр цикла). Выражение 2 является условием выполнения цикла. Выражение 3 обычно определяет изменение параметра цикла,

Алгоритм выполнения цикла *for* представлен на блок-схеме на рисунке 81.



Рисунок 81. Алгоритм выполнения цикла *for*.

Согласно рисунку 81, после вычисления выражения 3 происходит возврат к вычислению выражения 2 (проверке условия повторения цикла).

С помощью цикла *for* нахождение $N!$ можно организовать так:

```

F=1;
for(i=1;i<=N;i++) F*=i;

```

Некоторых элементов в операторе *for* может не быть, однако разделяющие их точки с запятой обязательно должны присутствовать.

В следующем примере инициализирующая часть вынесена из оператора **for**.

```
F=1;
i=1;
for(;i<=N;i++) F=F*i;
```

Ниже показан еще один вариант вычисления $N!$. В нем на месте тела цикла находится пустой оператор, а вычислительная часть внесена в выражение 3:

```
for(F=1, i=1; i<=N; F=F*i, i++) ;
```

или

```
for(F=1, i=1; i<=N; F*=i++) ;
```

В языке Си оператор **for** является достаточно универсальным средством для организации циклов. С его помощью можно программировать даже итерационные циклы.

Ниже показан пример вычисления суммы элементов гармонического ряда, превышающих заданную величину ε :

```
for(n=1, S=0; 1.0/n>eps && n<INT_MAX; n++) S+=1.0/n;
```

Та же сама задача с пустым телом цикла:

```
for(n=1, S=0; 1.0/n>eps && n<INT_MAX; S+=1.0/n++);
```

Следующий фрагмент программы на Си++ содержит два вложенных цикла **for**. для получения таблицы умножения.

```
for(x=2; x<=9; x++)
    for(y=2; y<=9; y++)
        cout<<"\n"<<x<<"*"<<y<<"="<<x*y;
```

Результатом выполнения этой программы будет:

```
2*2=4
2*3=6
. . .
9*8=72
9*9=81
```

Если выполнение очередного шага цикла требуется завершить до того, как будет достигнут конец тела цикла, используем оператор **continue**.

Следующий фрагмент программы обеспечивает вывод на экран всех четных чисел в диапазоне от 1 до 100:

```
for(i=1;i<=100;i++){if(i%2) continue; cout<<"\t"<<i;}
```

Для нечетных значений переменной *i* остаток от деления на 2 будет равен единице, этот результат воспринимается как значение «истина» в условии ветвления, и выполняется оператор **continue**. Он завершит очередной шаг цикла, выполнение цикла перейдет к следующему шагу.

Оператор **break** используется для досрочного прерывания циклов и выполняет переход к оператору, следующему за телом цикла.

Компиляция. Построчное выполнение программы осуществляется путем нажатия клавиши **F7** (команда меню **Run, Trace into**). Каждое нажатие **F7** приводит к выполнению одной строки кода, которая при этом выделяется подсветкой. При нажатии **F7** происходит «заход» в код функций, если в выполняемой строке есть обращение к функции.

Команда **Run, Step over** или **F8** также осуществляет построчное выполнение программы, но без захода в функции.

По команде **Run, Go to cursor** или **F4** программа выполняется до точки расположения курсора и останавливается. Дальнейшее выполнение программы можно делать построчно, нажимая **F7** или **F8**. Ускорить отладку можно с использованием **Точек останова (BreakPoint)**, установить или снять которые можно используя клавиши **Ctrl+F8**.

Выход из режима отладки производится командой **Run, Program reset** или нажатием комбинации клавиш **Ctrl+F2**.

Можно прервать отладку и продолжить дальнейшее выполнение программы, выполнив команду **Run, Run или Ctrl+F9**.

После компиляции запустить программу можно клавишами **Ctrl+F5**, или путем запуска файла с расширением ***.exe** в папке **Debug** расположения проекта.

Используемые при программировании команды

- **Build Solution ()** – собрать проект. При этом перекомпилируются все файлы проекта.
- **Rebuild Solution ()** – пересобрать проект.

- **Clean Solution** – очистить проект. При этом удаляются все лишние файлы, необходимые на момент разработки и отладки, но не нужные в конечном продукте.
- **Compile ()** – скомпилировать проект. При этом перекомпилируются только измененные файлы проекта.
- **Start Debugging ()** – начать отладку. Запускает программу под отладчиком.
- **Start without Debugging ()** – запустить без отладчика. Просто осуществляется запуск откомпилированной программы.
- **Step Into ()** – Пошаговое выполнение с заходом в процедуру.
- **Step Over ()** – Пошаговое выполнение без захода в процедуру.
- **Toggle Breakpoint ()** – Установить/снять точку останова.
- **Breakpoints ()** – показать текущие точки останова.

Задания для работы

Задание 1.

1. Создать простейшее приложение (рис. 82), выводящее на экран фразу «Пора работать», получить результат на экране.
2. Произвести процедуру компиляции.
3. Проанализировать программу на наличие ошибок программирования. В случае обнаружения ошибок, устранить их.

```
#include "stdafx.h"
#include ;
using namespace std;
void main()
{
// подключаем русский язык для отображения в консоли
setlocale(LC_STYPE, "Russian");
// выводим сообщение на экран консоли
cout <<"Пора работать" << endl;
system ("pause");
}
```

Рисунок 82. Задание 1.

- Задание 2.** Создать простейшее приложение (рис. 83), позволяющее вводить произвольное целое число, увеличивать его на единицу, получить результат на экране.

```
#include "stdafx.h"
#include ;
using namespace std;
void main()
{
// подключаем русский язык для отображения в консоли
setlocale(LC_STYPE, "Russian");
int x; // определяем целый тип данных для переменной x
// выводим приглашение для ввода значения переменной
cout << x; "Введите целое значение x: "; cin >> x; // вводим значение x с консоли
x++; // увеличиваем значение x на 1
// выводим результат вычисления на экран консоли
cout << "Результат: " << x << endl;
system ("pause");
}
```

Рисунок 83. Задание 2.

Содержание отчета

1. Цель работы.
2. Подробное описание методики создания приложения согласно варианту.
3. Листинг действующей программы.
4. Выводы.

Список использованных источников

1. Макаров, Е. Г. Mathcad: учебный курс / Е. Г. Макаров.— СПб: Питер, 2009.— 384 с.
2. Солодов, А. П. Mathcad. Дифференциальные модели / А. П. Солодов, В. Ф. Очков.— М.: Издательство МЭИ, 2002.— 239 с.
3. Глушаков, С. В. Математическое моделирование. Mathcad 2000. Matlab 5.3 / С. В. Глушаков, И. А. Жакин, Т.С. Хачиров.— Харьков; М.: Фолио, 2001. — 524 с.
4. Воскобойников, Ю. Е. Программирование и решение задач в пакете Mathcad. Руководство программиста / Ю. Е. Воскобойников, В. Ф. Очков.— Новосибирск: НГАСУ, 2002.— 138 с.
5. Охорзин, В. А. Прикладная математика в системе MATHCAD / В. А. Охорзин.— СПб, Лань, 2008. — 352 с.
6. Очков, В. Ф. Mathcad 14 для студентов, инженеров и конструкторов / В.Ф. Очков.— СПб, БХВ - Петербург, 2007.— 368 с.
7. Попов, В. Б. Основы компьютерных технологий / В. Б. Попов.— М.: Финансы и статистика, 2002. — 704 с.
8. Поршнева, С. В. Численные методы на базе MATHCAD / С. В. Поршнева, И. В. Беленкова. — СПб: БХВ-Петербург, 2005. — 464 с.
9. Плисс, А. И. MathCad 2000. Математический практикум для экономистов и инженеров / А. И. Плисс, Н. А. Сливина.— М.: Финансы и статистика, 2003. — 655 с.
10. Гурский, Д. А. Вычисления в MathCad / Д. А. Гурский. — Минск: Новое знание, 2003. — 814 с.
11. Черных И. В. Пакет ELCUT: моделирование устройств индукционного нагрева, Математика в приложениях. Среда разработки, EXPONENTA PRO, 2003, №2(2), с. 4-8.
12. ELCUT. Моделирование двумерных полей методом конечных элементов. Версия 5.5. Руководство пользователя // СПб: Производственный кооператив ТОР, 2007.— 298 с.
13. Татур, Т. А. Основы теории электромагнитного поля: Справочное пособие / Т. А. Татур. — М.: Высшая школа, 1989. — 271 с.
14. Кокцинская Е.М. Автоматизация расчёта кабеля в программе ELCUT // Videonauka: сетевой журнал, 2016, №1(1). URL: <https://videonauka.ru/stati/13-tekhnicheskie-nauki/39-avtomatizatsiya-raschjota-kabelya-v-programme-elcut> (дата обращения 15.11.2017).
15. Wilson E.L., Nickell R.E. Application of the Finite Element Method to Heat Conduction Analysis // Nuclear Engineering and Design, 1966, № 4, pp. 276–286.
16. Chari M.V., Silvester P. Analysis of turboalternator magnetic field by finite element //IEEE Trans. PAS. 1971. Vol. 90, №2. P. 970–976.
17. Лыткин Л.К., Филатов Г.А., Цепилов Г.В. Расчет автоэмиссионных структур с помощью ELCUT // Современные

проблемы науки и образования, 2014, № 5. URL: <https://science-education.ru/ru/article/view?id=15041> (дата обращения: 15.11.2017).

18. Гурова Е.Г., Панченко Ю.В., Макаров С.В. Моделирование характеристик компенсатора жесткости на основе супермагнитов с использованием программы ELCUT // *Фундаментальные исследования*. – 2015. – № 11-7. – С. 1313-1317. URL: <https://www.fundamental-research.ru/ru/article/view?id=39830> (дата обращения: 15.11.2017).

19. Черных, И. В. Моделирование электротехнических устройств в MATLAB, SimPowerSystems и Simulink / И. В. Черных.— М.: ДМК Пресс, 2007.— 288 с.

20. Дьяконов, В. П. MATLAB 6. Учебный курс / В. П. Дьяконов.— СПб: Питер, 2001.— 592 с.

21. Тарасик, В. П. Математическое моделирование технических систем / В.П. Тарасик.— Минск: Дизайн ПРО, 2004.— 640 с.

22. Лазарев, Ю. Моделирование процессов и систем в MATLAB. Учебный курс / Ю. Лазарев.— СПб: Питер; Киев: Изд. группа ВУВ, 2005.— 512 с.

23. Алексеев, Е. Р. MATLAB 7. Самоучитель / Е. Р. Алексеев, О. В. Чеснокова.— М.: ИТ Пресс, 2006.— 464 с.

24. Мартынов, Н. Н. MATLAB 5.x. Вычисления, визуализация, программирование / Н. Н. Мартынов, А. П. Иванов.— М.: КУДИЦ-ОБРАЗ, 2000.— 336 с.

25. Плохотников, К. Вычислительные методы. Теория и практика в среде MATLAB / К. Плохотников.— М.: Горячая Линия–Телеком, 2013.— 496 с.

26. Цифровая обработка сигналов и MATLAB / А. И. Солонина [и др.].— СПб: БХВ-Петербург, 2013.— 512 с.

27. Гультияев, А. Р. Визуальное моделирование в среде MATLAB Учебный курс / А. Р. Гультияев.— СПб : Питер, 2000.— 432 с.

28. Данилов, А. И. Компьютерный практикум по курсу “Теория управления”. Simulink-моделирование в среде MATLAB: Учебное пособие / А. И. Данилов; под ред. А. Э. Софиева.— М.: МГУИЭ, 2002.— 128 с.

29. Дьяконов, В. MATLAB 6/6.1/6.5 + Simulink 4/5 в математике и моделировании / В. Дьяконов.— М.: СОЛОН-Пресс, 2003.— 576 с.

30. Хоровиц, П. Искусство схемотехники / П. Хоровиц, У. Хилл.— М.: БИНОМ, 2015.— 704 с.

31. Бесекерский, В. А. Теория систем автоматического управления / В. А. Бесекерский, Е. П. Попов.— СПб: Профессия, 2003.— 752 с.

32. Малеев, Р. А. Компьютерные технологии: учеб. пособие / Р. А. Малеев, В. В. Регада, О. Н. Регада.— Пенза: Изд-во ПГУ, 2014.— 230 с.

33. Сергиенко, А. Б. Цифровая обработка сигналов / А. Б. Сергиенко.— СПб: Питер, 2002. — 608 с.
34. Виртуальная лаборатория математического моделирования. URL: <http://www.mathmod.narod.ru/>
35. Образовательный математический сайт Exponenta.ru. URL: <http://old.exponenta.ru/soft/matlab/matlab.asp>
36. Моделирование, оптимизация и информационные технологии. Электронный журнал. URL: https://moit.vivt.ru/?page_id=287&lang=ru
37. Новый подход к моделированию полей. URL: <https://elcut.ru/>
38. Макаров, Е. Г. Инженерные расчеты в Mathcad. Учебный курс. URL: <http://emakarov.nsknet.ru/>
39. Аллен И. Голуб. Правила программирования на Си и Си++: пер. с англ. – М.: Вильямс, 2001. – 241 с.
40. Павловская, Т.А. С/С ++. Программирование на языке высокого уровня: учебник для вузов / Т.А. Павловская.– СПб: Питер, 2010. – 461 с.
41. Прата, С. Язык программирования С++. Лекции и упражнения / С. Прата.– СПб: Питер, 2003. – 645 с.
42. Шилдт, Г. С++: базовый курс / Г. Шилдт. – М.: Вильямс, 2010. – 624 с.
43. Подбельский, В. В. Язык СИ++ / В. В. Подбельский. – М.: Финансы и статистика, 2008. – 560 с.
44. Дейтел, Х. М. Как программировать на С++ / Х. М. Дейтел, П. Дж. Дейтел. – М.: Бином, 2008. – 1456 с.
45. Культин, Н. Б. С/С ++ в задачах и примерах / Н. Б. Культин. – СПб: БХВ-Петербург, 2008. – 281 с.

Учебное издание

ГАЛЬЦЕВА Ольга Валерьевна

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ПРИБОРОСТРОЕНИИ

Учебное пособие

Научный редактор
кандидат технических наук, доцент
Б.Б. Мойзес

*Компьютерная верстка
О.В. Гальцева*

Для дизайна обложки использованы материалы из открытых интернет-источников

**Отпечатано в Издательстве ООО «Вайар» в полном соответствии с качеством
предоставленного оригинал-макета**

Подписано к печати 20.11.2020. Формат 60x84/16. Бумага «Снегурочка».
Печать XEROX. Усл.печ.л. 5,58. Уч.-изд.л. 2,16.
Заказ 406. Тираж 100 экз.

Издательство ООО «Вайар»
г. Томск, ул. Советская, 69
т. 8(3822) 52-98-11, yr.tomsk.ru