

МЕЖГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«БЕЛОРУССКО-РОССИЙСКИЙ УНИВЕРСИТЕТ»

Кафедра «Транспортные и технологические машины»

ИНФОРМАТИКА

*Методические рекомендации к курсовому проектированию
для студентов специальностей 1-36 11 01 «Инновационная
техника для строительного комплекса (по направлениям)»
и 1-37 01 02 «Автомобилестроение (по направлениям)»
очной и заочной форм обучения*



Могилев 2022

УДК 004.4
ББК 32.81
И74

Рекомендовано к изданию
учебно-методическим отделом
Белорусско-Российского университета

Одобрено кафедрой «Транспортные и технологические машины»
«01» февраля 2022 г., протокол № 7

Составитель ст. преподаватель В. И. Сёмчен

Рецензент канд. техн. наук, доц. А. Е. Науменко

Методические рекомендации разработаны на основании рабочих программ по дисциплине «Информатика» для студентов специальностей 1-36 11 01 «Инновационная техника для строительного комплекса (по направлениям)» и 1-37 01 02 «Автомобилестроение (по направлениям)» и предназначены для использования при выполнении курсовой работы.

Учебно-методическое издание

ИНФОРМАТИКА

Ответственный за выпуск	И. В. Лесковец
Корректор	Т. А. Рыжикова
Компьютерная верстка	Н. П. Полевничая

Подписано в печать . Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.
Печать трафаретная. Усл. печ. л. . Уч.-изд. л. . Тираж 36 экз. Заказ №

Издатель и полиграфическое исполнение:
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет».
Свидетельство о государственной регистрации издателя,
изготовителя, распространителя печатных изданий
№ 1/156 от 07.03.2019.
Пр-т Мира, 43, 212022, г. Могилев.

© Белорусско-Российский
университет, 2022

Содержание

Введение.....	4
1 Цель и задачи курсовой работы.....	5
2 Организация курсового проектирования.....	5
3 Примерная тематика курсовых работ.....	6
4 Содержание курсовой работы.....	6
5 Требования к содержанию разделов пояснительной записки.....	7
6 Оформление расчетно-пояснительной записки.....	13
7 Общие требования к программе.....	16
8 Защита курсовой работы.....	17
Список литературы.....	18
Приложение А. Структура титульного листа.....	19
Приложение Б. Пример выполнения задания.....	20

Введение

Информатика является дисциплиной, обеспечивающей подготовку по использованию информационных технологий в процессе обучения в вузе и последующей профессиональной деятельности. При изучении дисциплины обучающиеся получают представление о программном обеспечении персональных компьютеров, осваивают методы постановки и решения прикладных задач, закрепляют знания и умения в применении математического моделирования и выборе численных методов, приобретают навыки алгоритмирования, создания и отладки программ на языке высокого уровня, анализируют полученные результаты. Полученные знания в дальнейшем используются при изучении общетехнических, специальных дисциплин и дисциплин специализации.

Цель методических рекомендаций – оказание помощи обучающимся при выполнении курсовой работы по дисциплине «Информатика». В методических рекомендациях изложено содержание курсовой работы и даны указания по выполнению практической части и расчетно-пояснительной записки.

1 Цель и задачи курсовой работы

Одним из важных видов учебной деятельности обучающихся, предусмотренных учебным планом дисциплины «Информатика», является курсовая работа, выполняемая во втором семестре.

Цели курсовой работы:

- систематизация, закрепление и углубление теоретических знаний, полученных в процессе обучения;
- оценка степени усвоения основных положений дисциплины;
- выработка и развитие навыков решения задач на ЭВМ с использованием языка высокого уровня в программном комплексе Matlab.

Главным результатом выполнения курсовой работы являются отлаженные и протестированные программные модули, обеспечивающие решение задач, сформулированных в задании, и документация, поясняющая основные этапы создания программы.

2 Организация курсового проектирования

Основанием для выполнения курсовой работы является индивидуальное задание, утвержденное заведующим кафедрой. В задании указывается тема, исходные данные, содержание расчетно-пояснительной записки, календарный график выполнения курсовой работы на весь семестр с указанием сроков выполнения отдельных этапов.

Непосредственное руководство выполнением курсовой работы осуществляет преподаватель, который обязан:

- давать рекомендации по содержанию разделов курсовой работы;
- оценивать качество выполнения курсовой работы в соответствии с предъявляемыми к ней требованиями;
- осуществлять систематический контроль выполнения курсовой работы в соответствии с графиком;
- информировать заведующего кафедрой о случаях несоблюдения студентом графика выполнения курсовой работы.

Руководитель курсовой работы имеет право:

- выбрать удобную для него и обучаемого форму организации работы, согласовать график подготовки курсовой работы и установить периодичность личных встреч;
- по результатам каждой встречи требовать, чтобы обучаемый подготовил план и согласовал дальнейшие шаги по подготовке курсовой работы;
- требовать исполнения полученных рекомендаций;
- при выставлении оценки принять во внимание соблюдение студентом контрольных сроков графика подготовки курсовой работы.

Подведение итогов курсового проектирования включает следующие этапы:

- сдача курсовой работы на проверку руководителю;

- доработка курсовой работы с учетом замечаний руководителя;
- защита курсовой работы.

Законченная курсовая работа, подписанная обучаемым, представляется руководителю на проверку, и, если она удовлетворяет предъявляемым требованиям, руководитель делает пометку на титульном листе о допуске к защите. В случае отрицательного заключения работу необходимо доработать или переработать. Срок доработки проекта устанавливается руководителем с учетом сущности замечаний и объема необходимой доработки.

3 Примерная тематика курсовых работ

При выполнении курсовой работы студент решает задачи следующей тематики:

- решение уравнений и их систем с применением численных методов;
- операции над матрицами и векторами;
- использование численных методов для интегрирования и дифференцирования;
- моделирование статических состояний системы;
- моделирование плоского движения твердого тела или механизма.

4 Содержание курсовой работы

Выполнение курсовой работы включает проведение анализа исходных данных и постановку задачи, на основании которых выбирают и обосновывают метод решения и приводят его математическое описание. Далее составляют алгоритм и программу для ЭВМ. Производят тестирование программы и полученные в результате выполнения данные анализируют и при необходимости вносят изменения в математическую модель, алгоритм и текст программы.

Рекомендуемый объем пояснительной записки 15–25 страниц. Структура пояснительной записки приведена в таблице 4.1.

Таблица 4.1 – Структура пояснительной записки

Наименование структурных элементов пояснительной записки	Объем, с.
Титульный лист	1
Задание на курсовую работу	1
Содержание	1
Введение	1
1 Наименование раздела 1	

Окончание таблицы 4.1

Наименование структурных элементов пояснительной записки	Объем, с.
1.1 Постановка задачи	1
1.2 Математическая модель	1–2
1.3 Алгоритм программы	2–4
1.4 Интерфейс программы	1–3
1.5 Текст программы	2–4
1.6 Описание использованных в программе операторов и подпрограмм	1–2
1.7 Тестирование и анализ результатов решения задачи	1–3
Заключение	1
Список литературы	1

5 Требования к содержанию разделов пояснительной записки

Раздел «**Введение**» должен быть кратким и четким. Из введения должно быть понятно, чему посвящена работа, какие задачи и с помощью каких методов в ней решают, какие результаты должны быть получены в ходе выполнения курсовой работы.

Во введении отражают следующие основные моменты:

- общая формулировка решаемой проблемы;
- цель и задачи курсовой работы;
- описание структуры курсовой работы (названия разделов и их краткая характеристика).

Цель и задачи курсовой работы определяют направления, по которым автор раскрывает тему курсовой работы. Цель курсовой работы – это конечный результат, достигаемый в ходе выполнения курсовой работы. Обычно цель курсовой работы созвучна названию ей темы. После формулировки цели формируют задачи курсовой работы. Задачи курсовой работы определяют основные этапы для достижения поставленной цели.

В разделе «**Постановка задачи**» приводят исходные данные и формулируют цель решения задачи, поставленной в задании на курсовое проектирование. Анализируют характер и сущность всех величин, используемых в задаче, формулируют конечную цель решения задачи.

При формулировке цели решения задачи определяют форму представления результатов, перечисляют параметры и зависимости, которые нужно получить в результате решения задачи.

При наличии схемы механической системы приводят сведения об особенностях ее конструкции, действующих внешних факторах и геометрических параметрах.

В разделе «**Математическая модель**» производят анализ методов решения поставленной задачи и обосновывают выбор одного из них.

Математическая модель – это формальное описание решение задачи в виде математических зависимостей, связывающих исходные данные задачи с искомыми величинами.

Разработка модели решаемой задачи заключается в получении уравнений, описывающих выбранный численный метод для расчетных задач или определяющих положения механической системы в зависимости от изменения начальных условий при решении задач аналитической и исследовательской геометрии на плоскости.

При необходимости составляют расчетную схему, указывают принятые допущения.

Модель должна правильно описывать основные законы физического процесса. Построение или выбор математической модели требует понимания проблемы и знания соответствующих разделов математики.

В записке излагают последовательность вывода уравнений модели в общем виде с расшифровкой входящих в них величин, не упоминаемых ранее.

Если на последующих этапах будет обнаружена непригодность выбранного метода, необходимо произвести корректировку выбранного метода решения задачи или принять другой численный метод.

В разделе «**Алгоритм программы**» производят описание алгоритмов решаемой задачи и осуществляют пошаговое описание логики их работы.

На практике наиболее распространены следующие формы представления алгоритмов:

- словесная (записи на естественном языке);
- графическая (изображения из графических символов).

Словесный способ не имеет широкого распространения по следующим причинам:

- такие описания строго не формализуемы;
- страдают многословностью записей;
- допускают неоднозначность толкования отдельных предписаний.

Перед началом разработки алгоритма необходимо четко уяснить задачу, что требуется получить в качестве результата, какие исходные данные необходимы и какие имеются в наличии, какие существуют ограничения на эти данные. Далее следует записать, какие действия нужно предпринять для получения из исходных данных требуемого результата и описать их в виде блок-схемы с помощью условных графических обозначений.

При разработке алгоритмов можно использовать следующие методы:

- метод частных целей, который заключается в сведении решения сложной задачи к рассмотрению последовательности более простых задач;
- метод подъема, суть метода в том, что вначале создается самый простой вариант построения алгоритма, который затем последовательно улучшается, пока не достигнет заданного качества.

Эта два метода следует использовать совместно: вначале сложную задачу разбивают на последовательность более простых подзадач, а затем для составления алгоритмов каждой из подзадач используют метод подъема или опять метод частных целей.

Особое внимание следует обратить на проверку выполнения условий и принятие решения по ее результатам, повторяемость (цикличность) вычислений.

Алгоритмы должны однозначно отображать логическую структуру процесса решения задачи.

Графическое изображение алгоритма в виде блок-схемы представляет собой последовательность блоков, соединенных линиями.

Внутри символов словами или с помощью формул указывают выполняемую операцию (ввод исходных данных; вычисление расчетных параметров; условие, изменяющее направление выполнения алгоритма).

Схемы алгоритмов следует выполнять в соответствии с требованиями [1]. Все блоки схемы должны иметь уникальные идентификаторы, необходимые для организации ссылок при описании работы алгоритма.

На практике в основном используют типы блоков, представленные далее.

Терминатор (рисунок 5.1). Используют для обозначения выхода во внешнюю среду и входа из внешней среды (начало или конец схемы). Если символ расположен в начале алгоритма, то внутри него записывают «Начало», а если в конце, то «Конец».

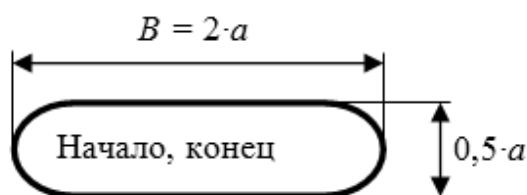


Рисунок 5.1 – Блок «Терминатор»

Данные (рисунок 5.2). Наиболее общий символ ввода-вывода данных, носитель которых не определен. Внутри символа записывают имена вводимых или выводимых данных.

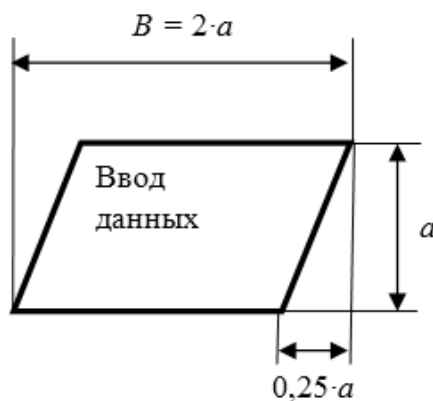


Рисунок 5.2 – Блок «Данные»

Процесс (рисунок 5.3). Указывает на обработку данных любого вида (выполнение определенной операции или группы операций, приводящее к

изменению значения, формы или размещения информации). Внутри символа записывают действия для выполнения операции или группы операций.



Рисунок 5.3 – Блок «Процесс»

Решение (выбор) (рисунок 5.4). Используют для выбора направления выполнения алгоритма в зависимости от некоторого условия, записываемого внутри символа: имеет один вход и ряд альтернативных выходов, один, и только один из которых может быть активизирован после вычисления условий, определенных внутри этого блока.

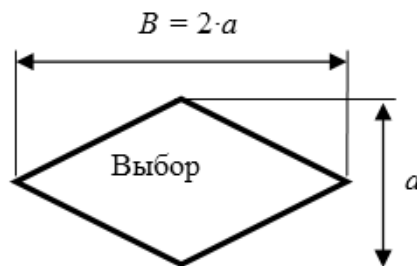


Рисунок 5.4 – Блок «Выбор»

Подготовка (рисунок 5.5). Указывает на использование цикла с параметром. Внутри записывается имя параметра цикла с указанием его начального и конечного значений, а также шаг изменения параметра цикла, если он не равен единице.

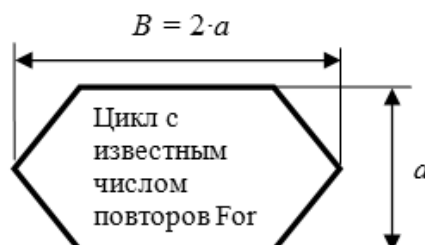


Рисунок 5.5 – Блок «Подготовка»

Предопределенный процесс (рисунок 5.6). Отображает одну или несколько операций, которые определены в подпрограмме, модуле. Внутри символа записывают имя подпрограммы, модуля.

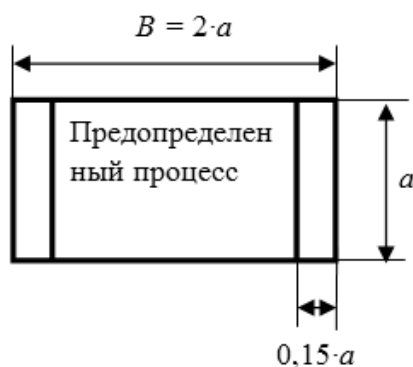


Рисунок 5.6 – Блок «Преопределенный процесс»

Линии потока указывают связи между символами. Для изображения линий потока существуют следующие правила:

- линии потока должны быть строго вертикальными или горизонтальными;
- направление линии потока сверху вниз и слева направо принимается за основное и стрелками не обозначается, в остальных случаях направление линии потока обозначается стрелками;
- изменение направления линии потока производится под углом 90° .

Соединитель (рисунок 5.7). Отображает выход в часть схемы и вход из другой части этой схемы и используется для обрыва линии потока и продолжения ее в другом месте. Соответствующие символы-соединители должны содержать одно и то же уникальное обозначение.

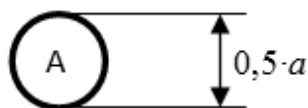


Рисунок 5.7 – Блок «Соединитель»

Комментарий (рисунок 5.8). Применяют для добавления описательных комментариев или пояснительных записей в целях объяснения операций, выполняемых отдельными символами или группой символов. Пунктирные линии в символе комментария связаны с соответствующим символом или могут обходить группу символов. Текст комментариев должен быть помещен около ограничивающей фигуры.

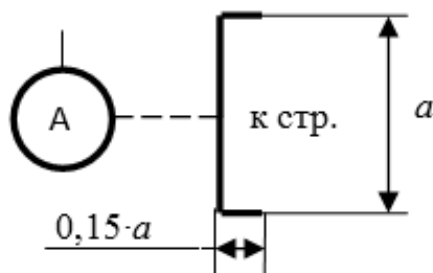


Рисунок 5.8 – Блок «Комментарий»

Граница цикла (рисунок 5.9). Используется для обозначения цикла с условием. Символ, состоящий из двух частей, отображает начало и конец цикла. Обе части символа имеют один и тот же идентификатор. Условия для инициализации, приращения, завершения и т. д. помещены внутри символа в начале или в конце в зависимости от расположения операции, проверяющей условие.

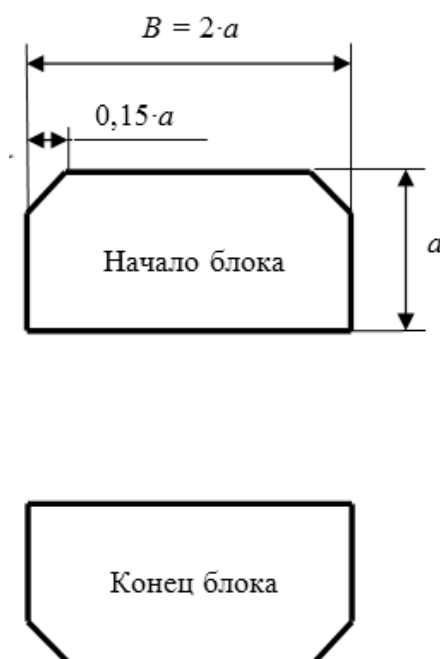


Рисунок 5.9 – Блок «Граница цикла»

Форма символов должна соответствовать [1], рекомендуется соблюдать пропорции, обозначенные на приведенных ранее изображениях, в качестве значения «а» задается целое число 2, 3 и т. д.

Каждому символу может присваиваться идентификатор (цифра, символ или их комбинация), который должен располагаться слева над символом. Идентификатор символа используют при описании работы алгоритма.

При наличии в блок-схеме ветвления над линиями связи пишут слово «да», если линия управления изображает поток при значении логического выражения «True», и слово «нет», если линия управления изображает поток при значении логического выражения «False».

Пример использования графических элементов для построения блок-схем показан в приложении Б.

В разделе «Текст программы» необходимо привести текст программы, включая программные модули, процедуры и функции, созданные в ходе разработки программы.

Раздел «Описание использованных в программе операторов и подпрограмм» содержит перечень операторов, их синтаксис и правила работы, описание стандартных и пользовательских подпрограмм (процедур и функций), применяемых в тексте программы [2–4]. Если в тексте программы исполь-

зованы классы среды или пользователя, то необходимо представить описание их методов и свойств.

При описании подпрограмм и методов приводят их назначение, заголовок, включая имя, список входных и выходных параметров с указанием типа и назначения параметров. Для свойств описывают имя свойства и тип принимаемых значений.

В разделе «Тестирование и анализ результатов решения задачи» приводят сведения о проверке правильности работы программы. Применяют наборы входных данных, для которых результат работы программы заранее известен. Входные данные подбирают так, чтобы охватить все возможные варианты работы алгоритма программы.

Оценка тестирования заключается в сравнении результатов, полученных при выполнении теста, с заранее известными контрольными значениями. В программах решения задач аналитической и исследовательской геометрии оценивают влияние наборов входных данных на характер движения и возможность существования кинематических связей между элементами схемы.

Раздел «Заключение» содержит итоги выполненной работы по созданию программы, ее назначение, степень соответствия требованиям задания, перечень выполненных работ, рекомендации по возможному улучшению эффективности работы созданной программы.

В разделе «Список литературы» приводят перечень использованных источников (не менее 5–6 ссылок), в том числе ГОСТов, справочников по языку программирования и т. п. В тексте пояснительной записки на все источники должны быть приведены ссылки.

Список литературы составляется по мере появления ссылок в тексте пояснительной записки и оформляется в соответствии с требованиями [3]. Пример оформления приведен в приложении Б.

6 Оформление расчетно-пояснительной записки

Курсовую работу оформляют в виде пояснительной записки. В пояснительной записке, используя общепринятую научно-техническую терминологию, сжато, логично и аргументированно излагают порядок выполнения всех этапов курсовой работы.

Пояснительную записку следует оформлять в соответствии с требованиями [5].

Пояснительная записка курсовой работы выполняется на листах бумаги формата А4 на одной стороне. Основную надпись выполняют по ГОСТ 2.104–68.

Титульный лист является первой страницей пояснительной записки, перенос слов на нем не допускается, точки в конце заголовков не ставятся (приложение А).

Страницы записки нужно нумеровать арабскими цифрами, соблюдая сквозную нумерацию по всему ее тексту. Номер страницы проставляют в правой нижней части листа без точки.

Нумерацию листов производят начиная с титульного, при этом на нем самом номер не ставится.

На листе содержания используют рамку высотой 40 мм, на всех остальных листах – 15 мм (рисунки 6.1 и 6.2).

					<Шифр>			
Изм.	Лист	№ докум.	Подпись	Дата				
Разраб.	Фамилия И. О.				<Название работы> Пояснительная записка	Лит.	Лист	Листов
Провер.	Фамилия И. О.						3	25
Реценз.						Бел.-Рос. ун-т ПДМ-191		
Н. Контр.								
Утверд.								

Рисунок 6.1 – Основная надпись высотой 40 мм

					<Шифр>			Лист
Изм.	Лист	№ докум.	Подпись	Дата				4

Рисунок 6.2 – Основная надпись высотой 15 мм

<Шифр> имеет следующую структуру:

ПДМ-191.00.<№ зачетки> ПЗ

где ПДМ-191 – код специальности и номер группы;

<№ зачетки> – последние три цифры номера зачетной книжки.

<Название работы> – тема, указанная в задании на курсовую работу.

Текст должен быть набран в редакторе Word и отпечатан на принтере. Текстовую часть документа выполняют шрифтом Times New Roman высотой 14 pt. Размеры полей: правое – 10 мм, верхнее – 15 мм, левое – 25, нижнее – 30 мм, межстрочный интервал – одинарный. Абзацный отступ в тексте записки устанавливают равным 10 мм, производят выравнивание текста по ширине.

Текст документа при необходимости разбивают на разделы и подразделы.

Разделы должны иметь порядковые номера в пределах всего документа, обозначенные арабскими цифрами без точки и записанные с абзацного отступа. Подразделы должны иметь нумерацию в пределах каждого раздела. Номер подраздела состоит из номера раздела и номера подраздела. В конце номера подраздела точку не ставят.

Разделы, как и подразделы, должны состоять из одного или нескольких пунктов. Каждый новый раздел необходимо начинать с нового листа, подразделы следуют друг за другом через двойной интервал. Текст после заголовка раздела и подраздела печатается через один интервал. Если между разделом и подразделом нет текста, то расстояние между ними три интервала.

Заголовки разделов и подразделов выделяют полужирным шрифтом и выравнивают по ширине. Названия разделов «Содержание», «Введение», «Заключение» и «Список литературы» располагают по центру.

Внутри текста записки могут быть приведены перечисления. Перед каждым перечислением следует ставить дефис или при необходимости ссылки в тексте документа на одно из перечислений строчную букву, после которой ставят скобку.

Иллюстрации (рисунки, графики, диаграммы, блок-схемы) и таблицы располагают в записке непосредственно после текста, в котором они упоминаются впервые, или на следующей странице, если в указанном месте они не помещаются.

Иллюстрации нумеруют арабскими цифрами. Применяют сквозную нумерацию в пределах всего документа, состоящую из одной цифры – «Рисунок 1». Допускается нумеровать иллюстрации в пределах раздела или подраздела. В этом случае номер иллюстрации должен состоять из номера раздела – «Рисунок 1.1». Наименование иллюстрации выполняют шрифтом 12 pt и размещают под иллюстрацией после пояснительных данных, выполняемых шрифтом 10 pt. Точка в конце подписи к иллюстрации не ставится.

Таблицы нумеруют арабскими цифрами. Применяют сквозную нумерацию в пределах всего документа, состоящую из одной цифры «Таблица 1». Допускается нумеровать таблицы в пределах раздела. В этом случае номер таблицы состоит из номера раздела и порядкового номера таблицы «Таблица 1.1». Таблицы при необходимости могут иметь название, которое располагают через дефис за номером таблицы «Таблица 1.1 – Исходные данные». Наименование таблицы выполняют шрифтом 12 pt и помещают перед таблицей. Точка после наименования таблицы не ставится.

На все иллюстрации и таблицы в тексте должны быть даны ссылки, при этом слова «Рисунок» и «Таблица» пишут полностью: «из рисунка 2.1 следует...», «в таблице 1.2 приведены...».

Формулы следует записывать в общем виде с новой строки, выравнивание формулы производят симметрично тексту. Формулы необходимо набирать в редакторе формул. Если формула одна и требуется пояснение символов, входящих в формулу, то за формулой ставят запятую, в противном случае ставят точку. Формулы, следующие одна за другой и не разделенные текстом, разделяют точкой с запятой. При этом за последней формулой ставят либо запятую, если необходима расшифровка символов формул, либо точку, если символы формул не надо расшифровывать.

Все формулы должны нумероваться сквозной нумерацией арабскими цифрами, которые записывают на уровне формулы справа в круглых скобках. Применяют либо сквозную нумерацию в пределах всего документа, состоящую из одной цифры (1), либо в пределах раздела или подраздела, тогда номер должен состоять из номера раздела или подраздела и порядкового номера формулы, разделенных точкой (3.1).

Пояснения символов, входящих в формулы, если они не пояснены ранее в тексте, должны быть приведены непосредственно под формулой. Пояснения каждого символа следует давать с новой строки в той последовательности, в которой символы приведены в формуле. Первая строка пояснения должна начинаться словом «где», после которого ставят пробел, затем приводят обозначение символа и через дефис дают описание физического смысла символа с указанием его размерности.

При использовании формул, научно-технических положений, стандартов и других данных необходимо делать ссылку на использованный источник, указывая его номер из списка литературы в квадратных скобках. Источники располагают в порядке появления ссылок на них в тексте. Запись источника в списке литературы производят в следующей последовательности: фамилия и инициалы автора (авторов), название, место издания (город), издательство, год издания и объем в страницах [6].

Приложения представляют собой материал, дополняющий текст пояснительной записки. Приложениями могут быть, например, графический материал, таблицы большого формата, расчеты, описания схем алгоритмов, тесты программ и т. д.

Приложения оформляют как продолжение курсовой работы. Каждое приложение должно начинаться с новой страницы с указанием вверху посередине страницы слова «приложение» и его обозначения, а под ними в круглых скобках строчными буквами указывают вид приложения (обязательное, рекомендуемое или справочное). Далее с новой строки симметрично относительно текста записывают с прописной буквы заголовки приложения. Приложения обозначают буквами русского алфавита, начиная с буквы А. При ссылках на приложения в тексте следует писать «в соответствии с приложением А ...».

Пример оформления пояснительной записки представлен в приложении Б.

7 Общие требования к программе

Программа, разрабатываемая в курсовой работе, должна удовлетворять следующим требованиям:

- использование программного комплекса Matlab;
- организация диалога с пользователем посредством GUI (графического интерфейса пользователя);
- вывод результатов осуществляют на экран в виде числовых значений и графических зависимостей;
- другим требованиям, указанным в задании на выполнение работы.

8 Защита курсовой работы

Защите в установленные сроки подлежит полностью выполненная и оформленная курсовая работа после ее проверки и соответствующего разрешения руководителя. Обучающийся представляет материалы по своей работе и делает доклад продолжительностью 5–7 мин, после чего отвечает на вопросы, поставленные членами комиссии. По результатам защиты выставляется оценка по десятибалльной системе на основе анализа пояснительной записки, доклада, демонстрации программных средств и ответов на вопросы.

Для оценки курсовой работы используются следующие критерии:

- соответствие разработанной программы заданию, включая правильность составления математической модели, построения алгоритма, полноту анализа результатов и тестирования;
- универсальность программы и ее модулей;
- качество оформления документации;
- полнота ответов на вопросы членов комиссии.

Примерный перечень этапов выполнения курсовой работы и количество баллов за каждый из них представлен в таблице 8.1.

Таблица 8.1 – Начисление баллов при выполнении курсовой работы

Этап выполнения	Минимум	Максимум
1 Теоретические исследования проблемы, анализ исходных данных, постановка задачи	6	10
2 Выбор и разработка математической модели	9	15
3 Разработка алгоритма	9	15
4 Разработка программного кода и его отладка в	9	15
5 Анализ результатов и оформление пояснительной записки	3	5
Итого за выполнение курсовой работы	36	60
Защита курсовой работы	15	40
Итого	51	100

Максимальные баллы начисляются за выполненные точно в срок разделы. Минимальные баллы начисляются при несвоевременном выполнении разделов (без соблюдения календарных сроков выполнения курсовой работы). Итоговая оценка вычисляется как сумма баллов и переводится в 10-балльную шкалу по таблице 8.2.

Таблица 8.2 – Итоговая оценка по курсовой работе

Оценка	10	9	8	7	6	5	4	3	2	1	0
Балл	100..94	93..87	86..80	79..73	72..66	65..59	58..51	50..34	33..17	16..1	0

Студент, не представивший в установленный срок готовую курсовую работу или не защитивший ее, считается имеющим академическую задолженность.

Список литературы

1 **ГОСТ 19.701–90**. Единая система программной документации. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения. – Москва: Стандартинформ, 2010. – 23 с.: ил.

2 **Дьяконов, В.** Matlab: учебный курс / В. Дьяконов. – Санкт-Петербург: Питер, 2001. – 560 с.

3 **Потемкин, В. Г.** Matlab 6: среда проектирования инженерных приложений / В. Г. Потемкин. – Москва: ДИАЛОГ-МИФИ, 2003. – 448 с.

4 **Кетков, Ю. Л.** MATLAB 7. Программирование, численные методы / Ю. Л. Кетков. – Санкт-Петербург : БХВ-Петербург, 2005. – 734 с.

5 **ГОСТ 2.105–95**. Единая система конструкторской документации. Общие требования к текстовым документам. – Минск: Межгосударственный совет по стандартизации, метрологии и сертификации. 1995. – 28 с.: ил.

6 **ГОСТ 7.1–2003**. Библиографическая запись. Библиографическое описание. Общие требования и правила оформления. – Москва: Изд-во стандартов, 2004. – 141 с.: ил.

Приложение А (справочное)

Структура титульного листа

Министерство образования Республики Беларусь
Межгосударственное образовательное учреждение высшего образования
«Белорусско-Российский университет»

Кафедра «Транспортные и технологические машины»

КУРСОВАЯ РАБОТА
по дисциплине «Информатика»

Название курсовой работы

Пояснительная записка

Специальность – номер группы.00.последние три цифры зачетки ПЗ

Выполнил студент гр.
_____ ФИО студента

Руководитель:
_____ ФИО руководителя

Могилев 2022 г.

Приложение Б (справочное)

Пример выполнения задания

Б.1 Вычисление корней нелинейного уравнения

Б.1.1 Постановка задачи.

В соответствии с заданием на курсовую работу необходимо найти с точностью 0,001 все корни уравнения на интервале от $-2,2$ до $2,2$, используя численные методы и расположить их в порядке убывания.

Представим уравнение в виде функции аргумента x :

$$f(x) = x^3 - 2,92 \cdot x^2 + 1,4355 \cdot x + 0,791136. \quad (\text{Б.1})$$

Вычисления организуем с использованием одного из численных методов приближенного вычисления, проверку результата произведем с помощью стандартных функций Matlab для решения нелинейных уравнений.

Для наглядного представления перед вычислениями выведем график функции.

Исходными данными для работы программы являются границы интервала поиска корней и точность расчета.

Ввод исходных данных организуем в режиме диалога с пользователем или с использованием графического интерфейса.

Организуем вывод графика исследуемой функции и значения найденных корней в числовом виде на экран в окне Matlab или в компонентах созданного графического интерфейса.

Б.1.2 Математическая модель.

Корнем уравнения называют такое значение аргумента, при котором функция обращается в ноль. Решить нелинейное уравнение – значит найти все его корни или доказать, что корней нет.

Проведем исследование существования корней уравнения (Б.1), определим их количество и расположение в исследуемом интервале для этого вычислим значения функции (таблица Б.1).

По результатам расчета построим график функции (рисунок Б.1).

Анализируя результаты, приходим к выводу, что на исследуемом интервале имеется два корня, т. к. значение функции меняет знак 2 раза и график функции дважды пересекает ось абсцисс.

Определим интервалы изоляции корня (отрезки локализации) так, чтобы на интервале этот корень был единственным. Таким образом, получаем два интервала локализации $[-0,5;0]$ и $[1;1,5]$.

Так как в задании не определено, каким методом производить вычисления, выберем один из известных численных итерационных методов:

- метод дихотомии (половинного деления);
- метод Ньютона (касательных);
- метод последовательного приближения к значению корня на отрезках длиной равной точности вычисления.

Таблица Б.1 – Значения аргумента и функции

x	$f(x)$	x	$f(x)$
-2,2	-27,1478	0,2	0,969436
-1,8	-17,0856	0,6	0,817236
-1,4	-9,68576	1	0,306636
-1	-4,56436	1,4	-0,17836
-0,6	-1,33736	1,8	-0,25376
-0,2	0,379236	2,2	0,464436

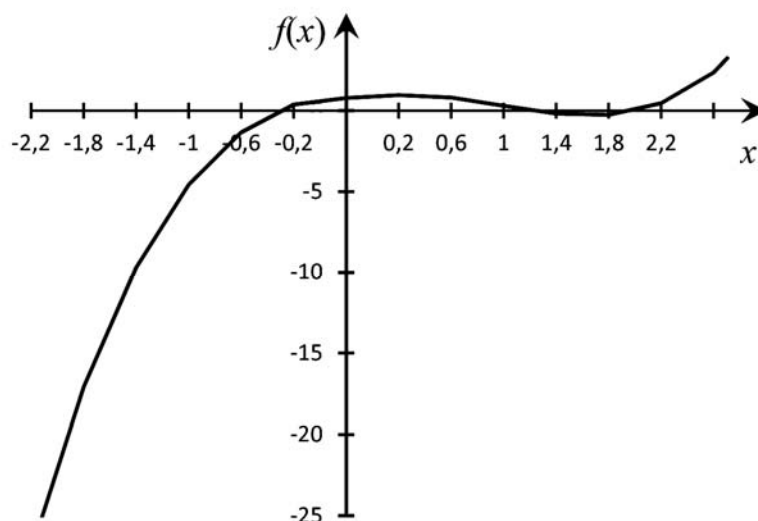


Рисунок Б.1 – График функции

Выбираем последний метод: как более простой и застрахованный от неудачи, он всегда приводит к решению, если на отрезке есть хотя бы один корень, позволяет найти несколько корней на одном отрезке без их предварительной локализации.

Итерационный процесс решения уравнения $f(x) = 0$ состоит в последовательном уточнении некоторого начального приближения x_0 . В результате итераций находят последовательность приближенных значений корня x_1, x_2, \dots, x_n к точному решению.

Для реализации метода необходимо последовательно рассматривать отрезки длиной равной точности, при этом, если знаки функции на концах отрезка различны, корень есть, иначе – корень отсутствует.

Началом первого отрезка принимаем начало отрезка локализации корня a .

Определим конец первого рассматриваемого отрезка:

$$x = a + e, \quad (\text{Б.2})$$

где e – заданная точность вычисления, $e = 0,001$.

Выполним расчет значений функции на концах отрезка $f(a)$ и $f(x)$.

$$\begin{aligned} f(a) &= a^3 - 2,92 \cdot a^2 + 1,4355 \cdot a + 0,791136; \\ f(x) &= x^3 - 2,92 \cdot x^2 + 1,4355 \cdot x + 0,791136. \end{aligned} \quad (\text{Б.3})$$

Проверим знаки функции на концах отрезка, вычислив значение логического выражения

$$f(a) \cdot f(x) < 0. \quad (\text{Б.4})$$

Если условие в выражении (Б.4) выполняется, то знаки значений функции на концах отрезка различны и один из корней будет лежать в рассматриваемой области значений аргумента.

Рассмотрим следующий отрезок, его начало a поместим в точку x – конец предыдущего отрезка – и далее выполним вычисления по формулам (Б.2) и (Б.3).

Данный процесс позволяет проверить все отрезки длиной равной точности, на интервале локализации корня итерации должны продолжаться, пока не достигнут конец интервала локализации корня и выполняется условие $a < b$.

Б.1.3 Алгоритм решения задачи.

Для реализации поставленной задачи составим алгоритм решения задачи в графической форме (рисунки Б.2–Б.5).



Рисунок Б.2 – Блок-схема алгоритма программы для вычисления корней стандартными средствами Matlab

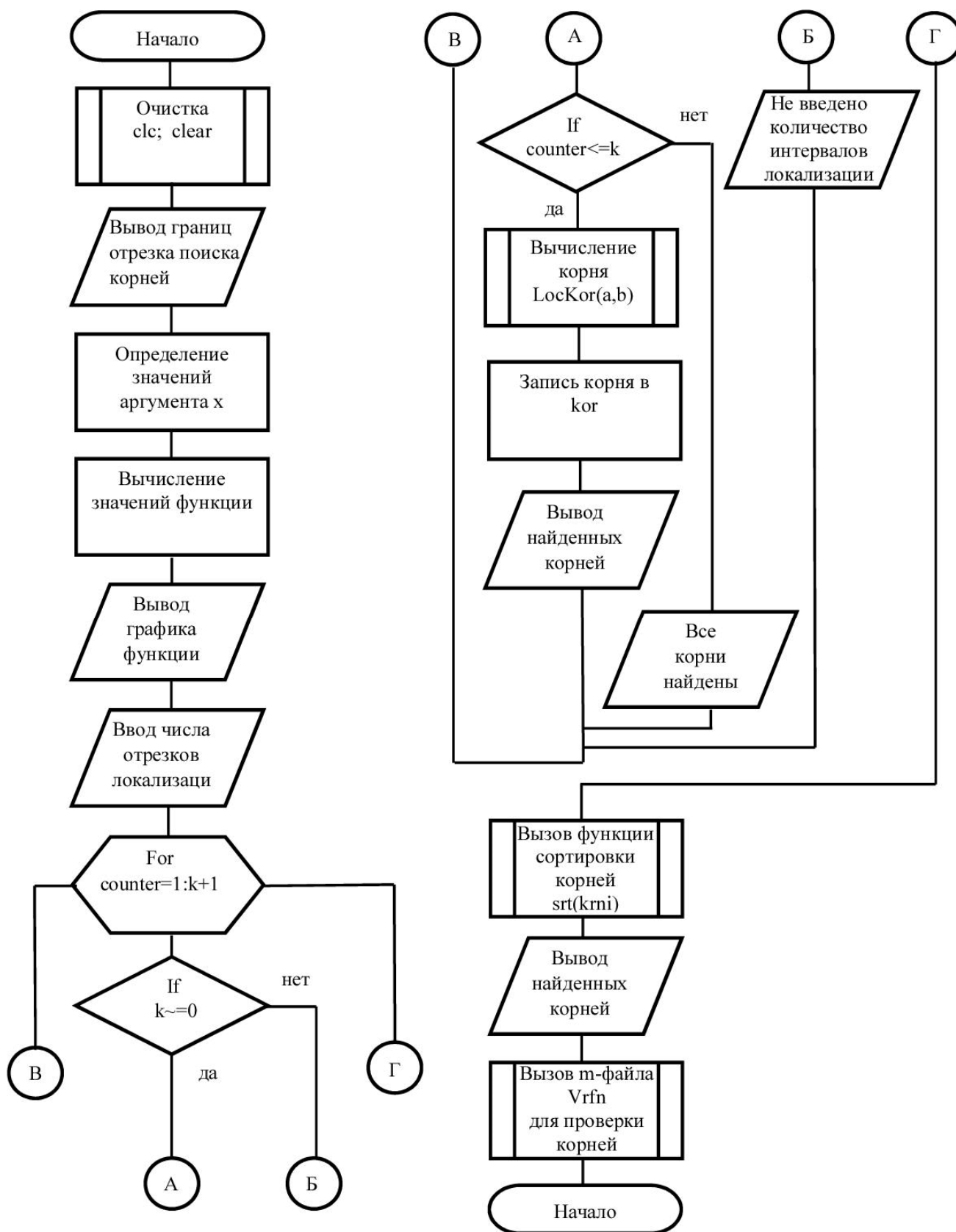


Рисунок Б.3 – Блок-схема алгоритма основной программы для вычисления корней

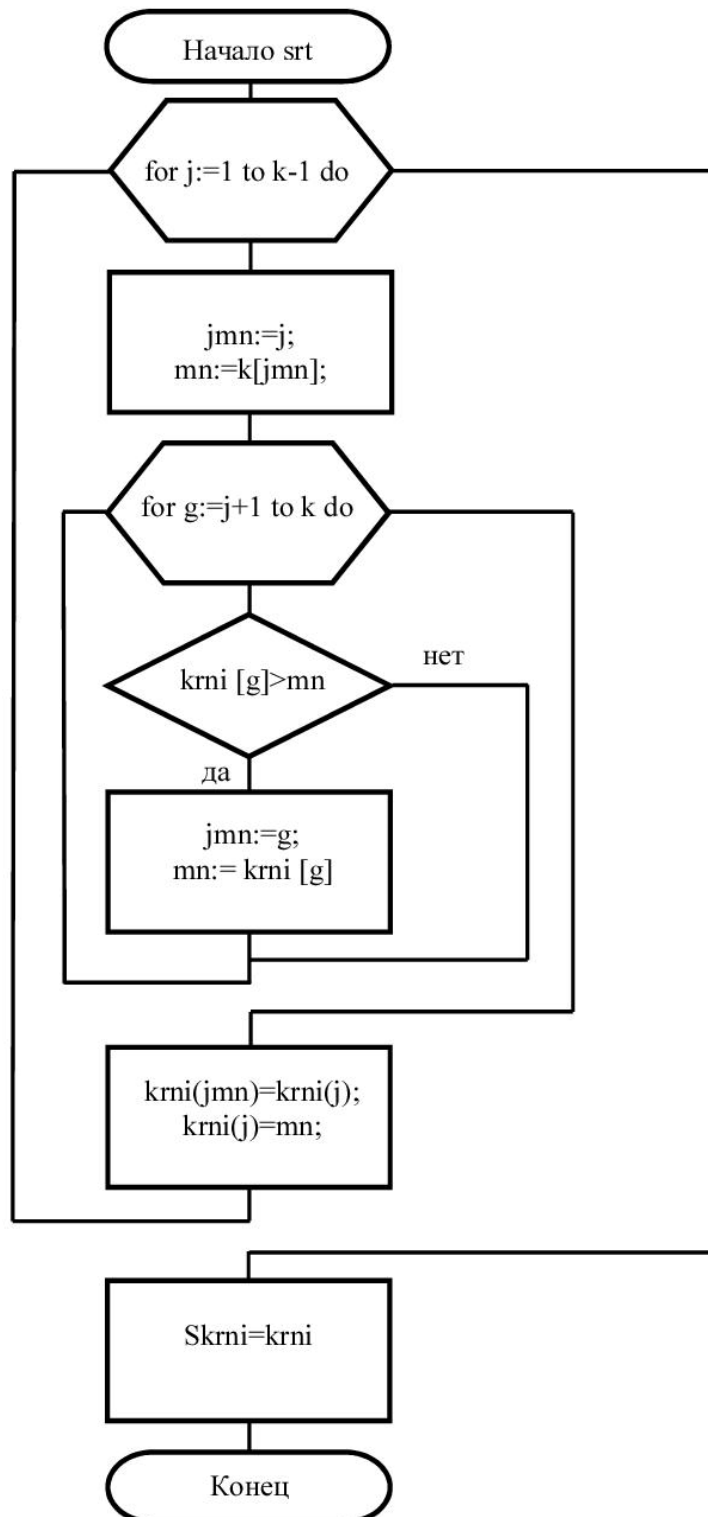


Рисунок Б.4 – Блок-схема функции сортировки корней srt

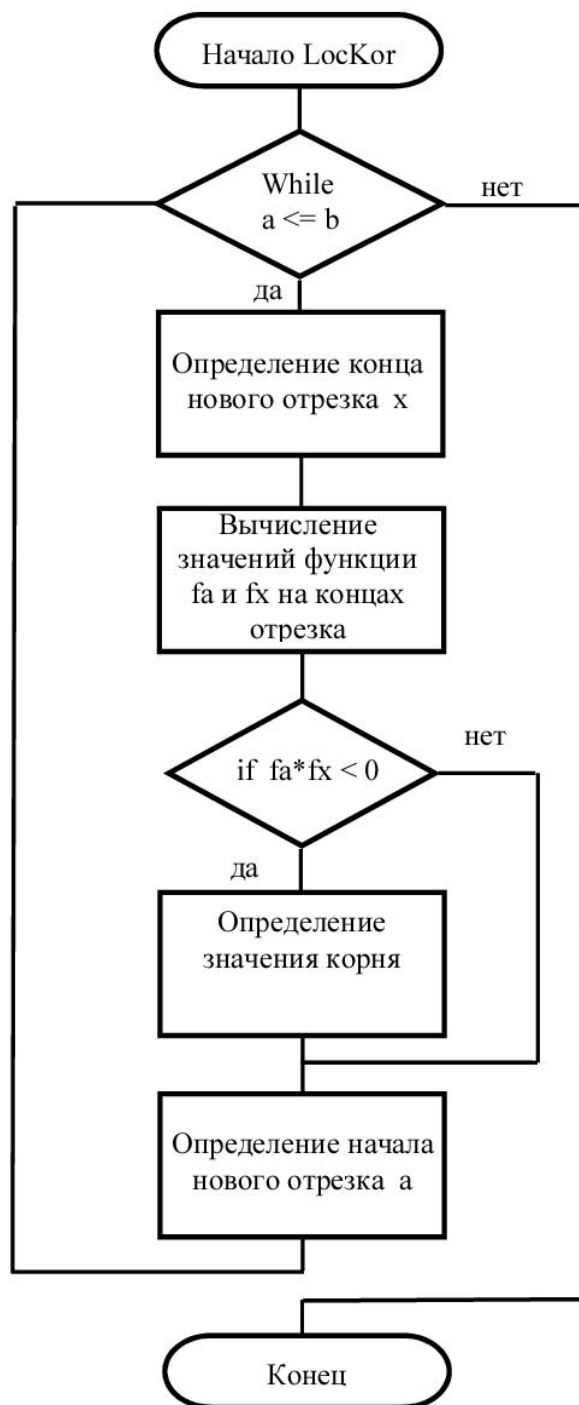


Рисунок Б.5 – Блок-схема функции нахождения корня LocKor

Б.1.4 Текст программы (без использования графического интерфейса).

Составим программу на языке Matlab для решения поставленной задачи. Реализуем в программе диалог с пользователем и обработку возможных ошибок при вводе исходных данных. Программа содержит функции пользователя.

Для поиска корней была использована функция пользователя LocKor.

```
function [kor]=LocKor(a,b)
while a <= b
    x= a +0.01;
    fa= a * a * a -2.92* a * a+1.4355* a +0.791136;
    fx=x*x*x-2.92*x*x+1.4355*x+0.791136;
    if fa*fx<0
        kor=x;
    end
    a=x;
end
end
```

Сортировка полученных корней производится посредством функции пользователя srt.

```
function [skrni]=srt(krni,k)
for j=1:k-1
    jmn=j;
    mn=krni(jmn);
    for g=j+1:k
        if krni(g)>mn
            jmn=g;
            mn=krni(g);
        end
    end
    krni(jmn)=krni(j);
    krni(j)=mn;
end
skrni=krni;
```

Ввод исходных данных и вызов подпрограмм пользователя для обработки данных и расчета осуществляется в основной программе main.

```
clc;
clear;
global a b

% График функции на отрезке [a,b]
a=input('Ввести начало исследуемого отрезка');
b=input('Ввести конец исследуемого отрезка');
x=a:0.1:b;
fx=x.^3-2.92.*(x.^2)+1.435.*x+0.791136;
figure(1);
plot(x,fx,'-r');
```

```

grid on;
ylabel('fx -----> ');
xlabel('x -----> ');
title('График исследуемой функции');
k=input('Ввести количество отрезков локализации');
counter=0;
for counter=1:k+1
    if k~=0
        if counter<=k
            a=input('Ввести начало исследуемого отрезка локализации');
            b=input('Ввести конец исследуемого отрезка локализации');
            [kor]=LocKor(a,b);
            krni(counter)=kor;
            disp(num2str(krni));
            else warning ('Все корни найдены');
        end
        else error ('Не введено количество интервалов локализации');
    end
end
[skrni]=srt(krni,k);
disp(['Результат сортировки корней,' ' ', num2str(skrni)]);
vrfn;
disp(['Корни вычисленные стандартной функцией'," ,num2str(transpose(r))]);

```

Для проверки правильности реализации численного метода поиска корней и визуализации графика функции составлена программа vrfn.

```

%Проверка численного метода
% Исходное уравнение  $z=x^3-2,92*x^2+1,435x+0.791136$ 
p=[1 -2.92 1.435 0.791136];
r=roots(p);

```

Б.1.5 Текст программы (с использованием графического интерфейса).

Составим программу на языке Matlab для решения поставленной задачи с использованием графического интерфейса пользователя для ввода и вывода исходных данных и управления работой программы.

Особенностью основной части программы является отсутствие проверки на ошибки, т. к. используемый метод позволяет найти все корни на рассматриваемом интервале без их предварительной локализации.

Для поиска корней и их сортировки использованы пользовательские функции LocKor и srt, а для поиска корней стандартным методом вызывается m-файл vrfn.

При нажатии кнопки «Построить графики» выполняется следующая функция обратного вызова:

```
function pushbutton1_Callback(hObject, eventdata, handles)
%Построение графиков
a=str2double(get(handles.edit1,'string'));
b=str2double(get(handles.edit2,'string'));
x=(a:0.25:b);fx=x.^3-2.92.*(x.^2)+1.435.*x+0.791136;
set(handles.text2,'String', fx);
plot(handles.axes2, x,fx,'-r. ');
ylabel('fx -----> ');
xlabel('x -----> ');
title('График исследуемой функции');
grid on;
```

При нажатии кнопки «Вычислить корни» выполняется следующая функция обратного вызова:

```
function pushbutton3_Callback(hObject, eventdata, handles)
a=str2double(get(handles.edit1,'string'));
b=str2double(get(handles.edit2,'string'));
[kor]=LocKor(a,b);
set(handles.text5,'String',kor);
[skrni]=srt(kor,3);
vrfn;
set(handles.text6,'String',r);
set(handles.text9,'String',skrni);
```

Б.1.6 Описание графического интерфейса программы (если предусмотрен).

Для организации диалогового режима работы с пользователем и управления работой программы разработаем графический интерфейс, используя возможности GUI программного комплекса Matlab.

При проектировании графического интерфейса, представленного на рисунке Б.6, использованы компоненты pushbutton1, pushbutton3 типа Push Button. Для изменения и добавления надписей на этих компонентах используют свойство String. Связь нажатия на кнопку с текстом программы реализована с помощью функций обратного вызова Callback этих компонентов.

При нажатии кнопки «Построить график» выполняется фрагмент программы, отвечающий за чтение введенных пользователем исходных данных, вычисления значений исследуемой функции и вывод графика и значений функции в компоненты графического окна.

Кнопка «Вычислить корни» выполняет запуск фрагмента программы для запуска вычисления и сортировки полученных корней и вывода результатов в компоненты графического окна удаления результатов вычислений, хранящихся в памяти и отображенных на экране.

Графический интерфейс также содержит компоненты edit1, edit2 типа Edit Text – редактируемые поля ввода для отображения или редактирования текстовых строк. Применяется в программе для указания начала и конца

отрезка, на котором отыскиваются корни. Для организации ввода и вывода данных используется свойство String компонента Edit Text.

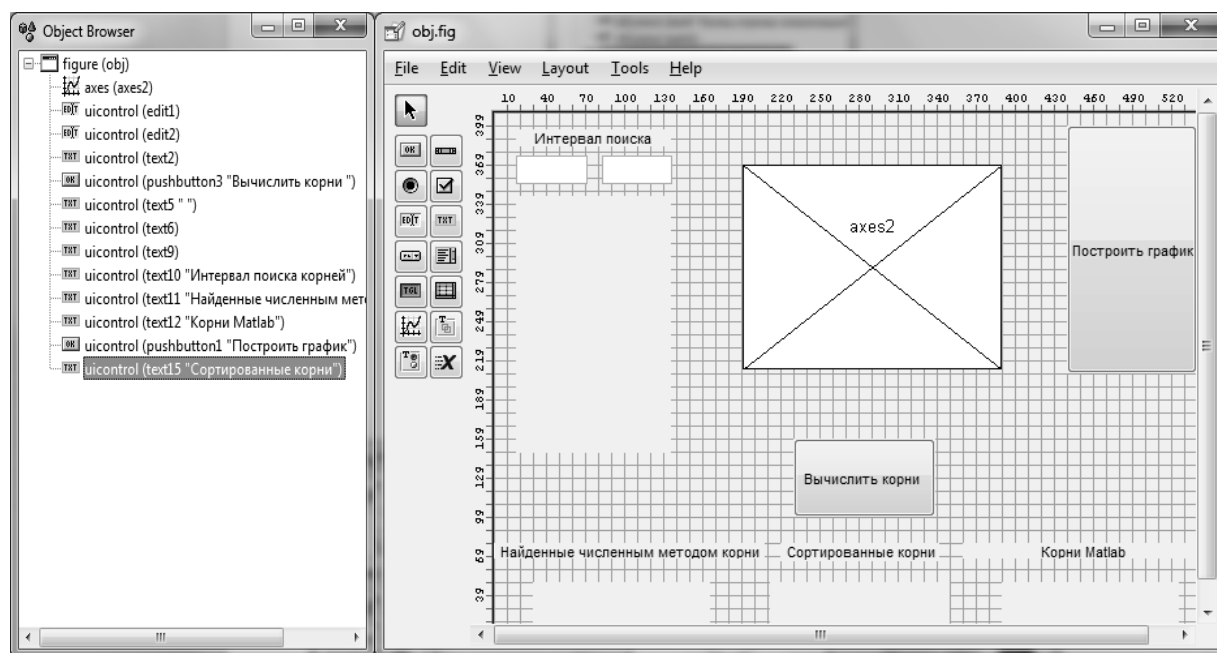


Рисунок Б.6 – Структура графического интерфейса программы

Компоненты text10, text11, text11, text15 типа Static Text – текстовые метки – используются для подписи других компонентов. Надписи на компонентах Static Text выполнены путем изменения свойства String.

На форме расположены текстовые метки:

- «Интервал поиска»;
- «Найденные численным методом корни»;
- «Сортированные корни»;
- «Корни Matlab».

Часть компонентов Static Text – text5, text6, text9 – предназначены для вывода результатов, имеют пустые значения свойства String и не отображаются до записи в них значений.

Компонент типа Axes – axes2 предназначен для вывода графической информации на графический интерфейс пользователя.

Для корректной работы GUI необходимо до нажатия кнопок ввести значения начала и конца интервала, на котором отыскиваются корни.

Б.1.7 Описание использованных в программе операторов и подпрограмм.

В процессе создания программы использовались следующие стандартные функции:

- input (Символьная константа) – отвечает за организацию диалога с пользователем при вводе исходных данных, используется в правой части оператора присваивания, после вызова выводит сообщение, содержащееся в Символьной константе, и блокирует работу программы, пока пользователь не осуществил ввод;

– `disp` (Выражение) – отвечает за вывод данных получаемых после вычисления выражения любого типа, вызывается непосредственно в тексте программы;

– `transpose (A)` – функция транспонирования матрицы или вектора использовалась в программе представления корней уравнения, полученных с использованием стандартной функции Matlab в виде вектора столбца;

– `roots (p)` – функция вычисления корней уравнения, коэффициенты при неизвестных которого и свободный член записаны в вектор `p`;

– `str2double (аргумент)` – функция для преобразования аргумента символьного типа в вещественное числовое значение, как правило, при чтении данных;

– `num2str (аргумент)` – функция для преобразования аргумента числового типа в символьное значение, как правило, при выводе данных на экран;

– `plot (x, y, s1)` – функция, обеспечивающая построение графика функции, `x` – вектор значений, `y` – вектор значений функции, `s` – параметр символьного типа определяющий внешний вид графика.

– `ylabel (символьная константа)`, `xlabel (символьная константа)` – функция для установки надписей возле осей `x`, `y`, содержание которой определяется символьной константой;

– `title (символьная константа)` – функция для установки в окне фигуры над графиком титульной надписи, содержание которой определяется символьной константой.

В процессе создания программы использовались следующие методы для работы со свойствами графических объектов:

– `get (handles.имя_объекта, свойство объекта)` – метод для чтения значений свойств, которые содержит графический объект при вводе исходных данных, используется в правой части оператора присваивания;

– `set (handles.имя_объекта, свойство объекта, имя_переменной)` – метод для вывода значений переменных на графический объект (записи значения свойства объекта), вызывается непосредственно в тексте программы.

Для решения задачи созданы две внешние функции пользователя `LocKor` и `srt`.

Функция `function [kor] = LocKor(a, b)` предназначена для нахождения корня на заданном интервале. Входными параметрами `a`, `b` являются границы отрезка локализации корней, в выходном параметре `kor` хранится значение найденного корня (или корней) для передачи в основную программу.

Функция `function [skrni] = srt(krni, k)` предназначена для сортировки найденных корней в порядке убывания. Входными параметрами `krni`, `k` являются вектор `krni`, содержащий найденные корни и целое число `k`, указывающее количество корней, в выходном параметре `skrni` хранятся отсортированные корни.

В тексте программы использовался оператор присваивания, имеющий синтаксис:

имя_переменной = Выражение;

где имя_переменной – имя простой переменной, структурированной переменной (вектора, матрицы), имя функции;

Выражение – арифметическое, логическое, строковое выражение или значение.

Тип переменной определяется системой автоматически по типу выражения, при работе оператора вычисляется выражения справа от символа «=» и затем результат присваивается переменной слева от символа «=».

Оператор повтора for использован в программе для организации циклических алгоритмов при известном количестве повторений цикла и обеспечивает выполнение тела цикла до тех пор, пока оно не будет выполнено со всеми значениями переменной цикла.

```
for Имя ПЦ= НПЦ : ШПЦ : КПЦ
    Инструкции
End
```

где Имя ПЦ – идентификатор (имя) переменной цикла;

НПЦ – начальное значение переменной цикла;

КПЦ – конечное значение переменной цикла,

ШПЦ – шаг, на который изменяется значение переменной цикла при повторении цикла;

Инструкции – оператор или группа операторов, составляющих рабочую часть (тело) цикла.

Для работы использованного оператора for должно выполняться условие $\text{НПЦ} \leq \text{КПЦ}$ при $\text{ШПЦ} > 0$.

Для реализации разветвляющихся алгоритмов при проверке данных в программе использована следующая форма условного оператора:

```
If Условие
    Инструкции_1
else Инструкции_2
end
```

где Инструкции 1 и 2 – операторы или группы операторов, выполняющихся в зависимости от значения логического выражения, вычисляемого при проверке условия.

Б.1.8 Тестирование и анализ результатов решения задачи (без использования GUI).

Для запуска программы необходимо открыть в Matlab файл main.m и выполнить его запуск. После запуска появятся строки диалогового режима для ввода границ исследуемого отрезка, представленные на рисунке Б.7.

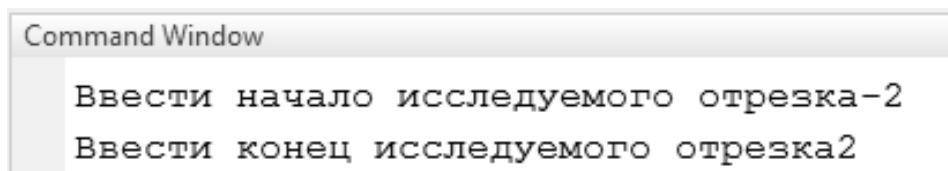


Рисунок Б.7 – Запрос на ввод данных в диалоговом режиме

После ввода исходных данных, предусмотренных в тексте программы, отобразится график функции, представленный на рисунке Б.8.

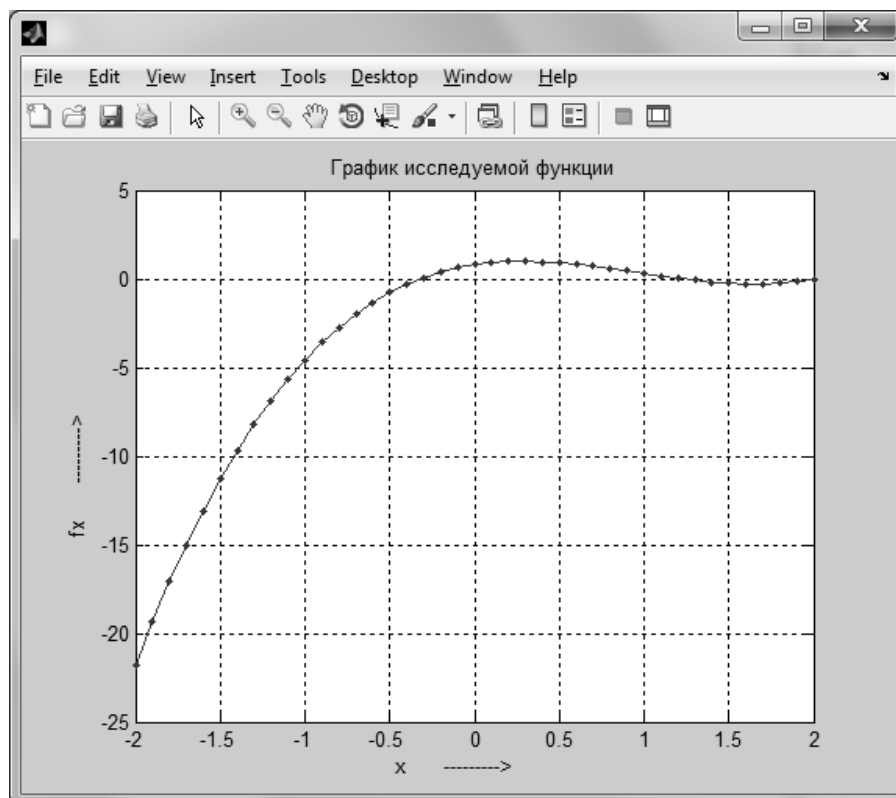


Рисунок Б.8 – Графическое представление исследуемой функции

Далее программа в диалоговом режиме запрашивает данные о количестве отрезков локализации корней и их границах. Для тестирования используем значения, которые определены при составлении математической модели и ее анализе, в дальнейшем при эксплуатации программы эти данные устанавливаются по выведенному ранее графику.

По мере ввода данных в диалоговом режиме программа вычисляет корни, формирует из них вектор и сортирует его убыванию.

После вычисления выводит корни, определенные стандартной функцией. Результаты работы программы приведены на рисунке Б.9. Различие в количестве корней при тестировании программы объясняется тем, что примененная стандартная функция отыскивает корни во всей области определения функции, а реализованный численный метод ограничен интервалом $[-2, 2]$.

Сравнивая значения, полученные при анализе математической модели, результаты работы программы при реализации численного метода и вычис-

лении стандартной функцией Matlab, можно сделать вывод о хорошей сходимости полученных значений и точности расчета.

```

Command Window
Ввести начало исследуемого отрезка-2
Ввести конец исследуемого отрезка2
Ввести количество отрезков локализации2
Ввести начало исследуемого отрезка локализации-0.5
Ввести конец исследуемого отрезка локализации0
-0.3199
Ввести начало исследуемого отрезка локализации0
Ввести конец исследуемого отрезка локализации1.5
-0.3199      1.2301
Warning: Все корни найдены
> In main at 27
Результат сортировки корней  1.2301      -0.3199
Корни вычисленные стандартной функцией  2.0106      1.2295      -0.32004

```

Рисунок Б.9 – Результаты вычисления корней

Очевидно, результаты тестирования подтверждают правильность программной реализации выбранного численного метода, соответствие найденных корней результату начального анализа.

Б.1.9 Тестирование и анализ результатов решения задачи (с использованием GUI).

Для запуска программы необходимо открыть в Matlab файл obj.fig. Далее в полях «Интервал поиска» указываем границы отрезка, на котором необходимо исследовать наличие корней (в данной реализации отсутствует необходимость в локализации корней, отыскиваются все существующие на интервале корне). При нажатии кнопки «Построить графики» считываются введенные данные, рассчитываются значения функции и выводятся в числовом виде, а также строится график функции. Результат работы представлен на рисунке Б.10. По полученным числовым значениям и графику определяют наличие и количество существующих на интервале корней.

Следующий шаг в работе программы – непосредственно вычисление корней и вывод результата при нажатии на кнопку «Вычислить корни», результат работы программы приведен на рисунке Б.11.

Сравнивая значения, полученные при анализе математической модели, результаты работы программы при реализации численного метода и вычислении стандартной функцией Matlab, можно сделать вывод о хорошей сходимости полученных значений и точности расчета.

Результаты тестирования подтверждают правильность программной реализации выбранного численного метода, соответствие найденных корней результату начального анализа.

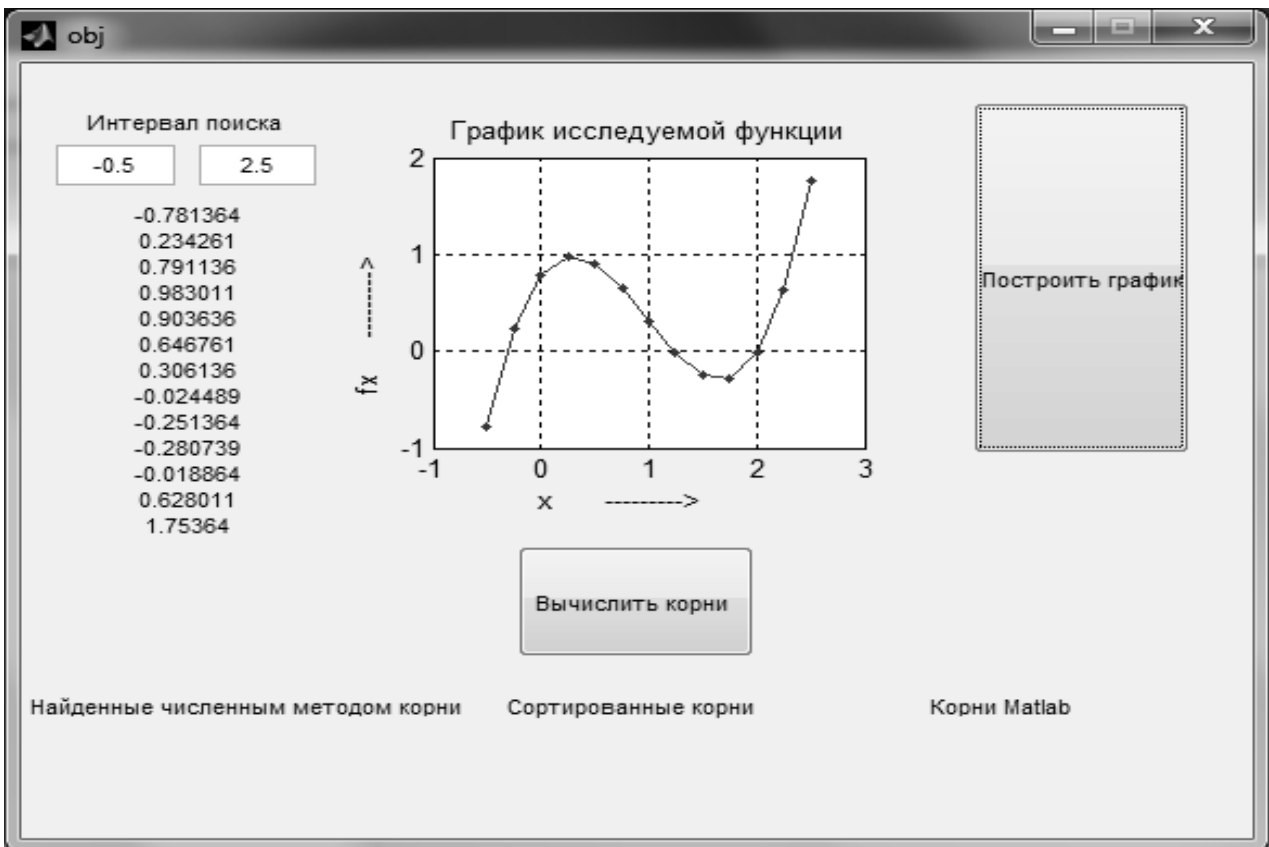


Рисунок Б.10 – Визуализация наличия корней на исследуемом интервале поиска

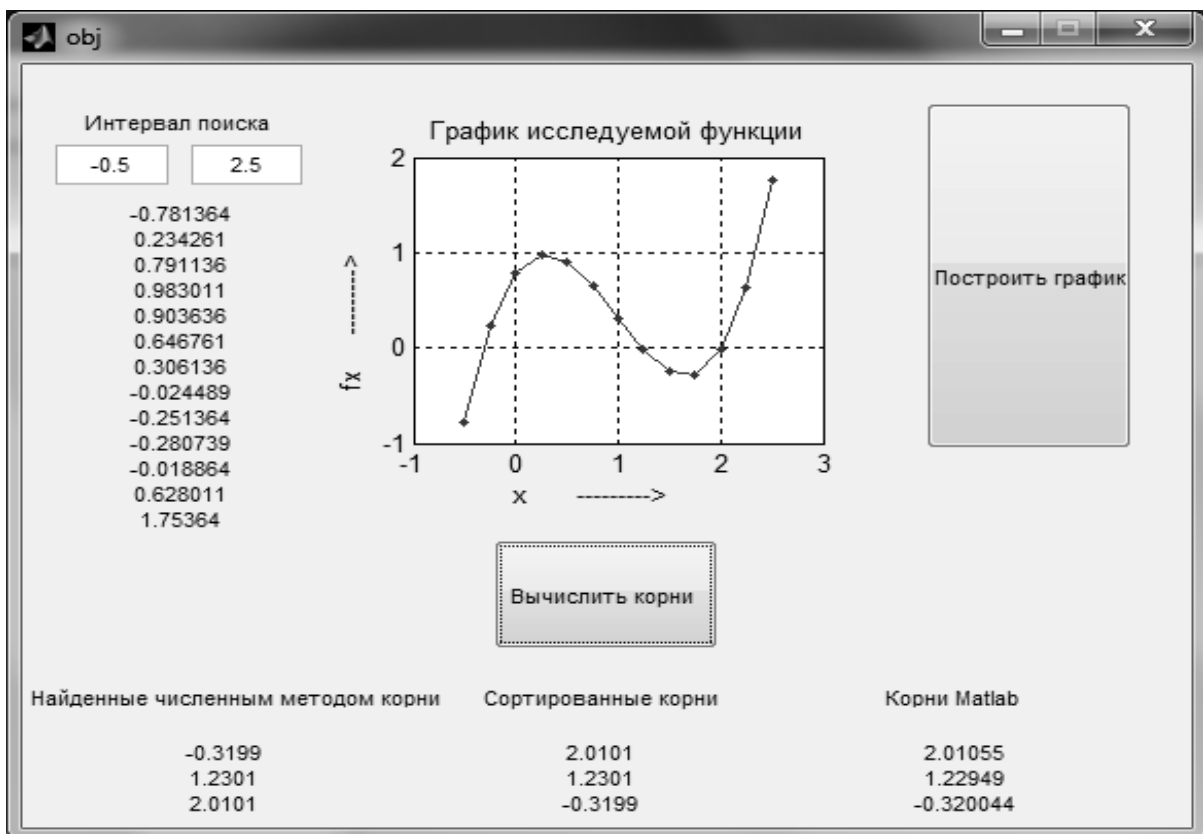


Рисунок Б.11– Результаты вычисления корней с использованием GUI