МИНОБРНАУКИ РОССИИ

федеральное государственное бюджетное образовательное учреждение

высшего образования

«Нижегородский государственный технический университет

им. Р.Е. Алексеева» (НГТУ)

Кафедра: «Цифровая экономика»

Дисциплина: «Численные методы»

**Лабораторная работа №4**

**«Методы решений нелинейных уравнений»**

Выполнил:

студент 3-го курса группы 21-САИ

Краличев Игорь Евгеньевич

Проверил:

д.ф.м.н., проф. Катаева Лилия Юрьевна

14.01.2024  
Подпись преподавателя : \_\_\_\_\_\_\_\_\_\_\_\_\_\_

Нижний Новгород, 2023

Оглавление

[Постановка задач 3](#_Toc156274969)

[*Работа №1* 3](#_Toc156274970)

[*Работа №2* 3](#_Toc156274971)

[*Работа №3* 3](#_Toc156274972)

[*Работа №4* 3](#_Toc156274973)

[*Работа №5* 3](#_Toc156274974)

[*Работа №6* 4](#_Toc156274975)

[*Работа №7* 4](#_Toc156274976)

[Методы решений и инструменты 4](#_Toc156274977)

[*Инструменты* 4](#_Toc156274978)

[*Методы решений* 5](#_Toc156274979)

[Таблица идентификаторов 10](#_Toc156274980)

[*Работа 1* 10](#_Toc156274981)

[*Работа 2* 11](#_Toc156274982)

[*Работа 3* 11](#_Toc156274983)

[*Работа 4* 12](#_Toc156274984)

[*Работа 5* 12](#_Toc156274985)

[*Работа 6* 13](#_Toc156274986)

[*Работа 7* 13](#_Toc156274987)

[Ручной счёт 14](#_Toc156274988)

[*Работа 1* 14](#_Toc156274989)

[*Работа 2* 17](#_Toc156274990)

[*Работа 3* 18](#_Toc156274991)

[Работа 5 19](#_Toc156274992)

[*Работа 6* 21](#_Toc156274993)

[*Работа 7* 24](#_Toc156274994)

[Реализация в excel 26](#_Toc156274995)

[*Работа 1* 26](#_Toc156274996)

[*Работа 2* 27](#_Toc156274997)

[*Работа 3* 29](#_Toc156274998)

[*Работа 5* 31](#_Toc156274999)

[*Работа 6* 33](#_Toc156275000)

[Реализация в Mathcad 34](#_Toc156275001)

[*Работа 1* 34](#_Toc156275002)

[*Работа 2* 36](#_Toc156275003)

[*Работа 3* 37](#_Toc156275004)

[*Работа 4* 38](#_Toc156275005)

[*Работа 5* 39](#_Toc156275006)

[*Работа 6* 40](#_Toc156275007)

[*Работа 7* 41](#_Toc156275008)

[Реализация в С++ 42](#_Toc156275009)

[*Меню пользователя* 42](#_Toc156275010)

[*Класс для работы 1* 43](#_Toc156275011)

[*Класс для работы 2* 45](#_Toc156275012)

[*Класс для работы 3* 47](#_Toc156275013)

[*Класс для работы 4* 49](#_Toc156275014)

[*Класс для работы 5* 50](#_Toc156275015)

[*Класс для работы 6* 52](#_Toc156275016)

[*Класс для работы 7* 54](#_Toc156275017)

[Реализация в Java 56](#_Toc156275018)

[*Меню пользователя* 56](#_Toc156275019)

[*Класс для работы 1* 57](#_Toc156275020)

[*Класс для работы 2* 60](#_Toc156275021)

[*Класс для работы 3* 62](#_Toc156275022)

[*Класс для работы 4* 64](#_Toc156275023)

[*Класс для работы 5* 65](#_Toc156275024)

[*Класс для работы 6* 67](#_Toc156275025)

[*Класс для работы 7* 69](#_Toc156275026)

[Список литературы 71](#_Toc156275027)

# **Постановка задач**

## ***Работа №1***

Вариант 10

1) Отделить корни аналитически

2) Отделить корни аналитически и уточнить один из них методом проб с точностью до 0.001

3) Отделить корни графически

4) Отделить корни графически и уточнить один из них методом проб с точностью до 0.001

## ***Работа №2***

Вариант 10

1) Отделить корни графически и уточнить один из них методом хорд с точностью до 0.001

2) Отделить корни аналитически и уточнить один из них методом хорд с точностью до 0.001

## ***Работа №3***

Вариант 10

1) Отделить корни графически и уточнить один из них методом касательных с точностью до 0.001

2) Отделить корни аналитически и уточнить один из них методом касательных с точностью до 0.001

## ***Работа №4***

Вариант 10

Комбинированным методом хорд и касательных решить уравнение третьей степени, вычислив корни с точностью до 0.001

## ***Работа №5***

Вариант 10

1) Отделить корни графически и уточнить один из них методом итераций с точностью до 0.001

2) Отделить корни аналитически и уточнить один из них методом итераций с точностью до 0.001

## ***Работа №6***

1) Используя метод итераций, решить систему нелинейных уравнений с точностью до 0.001

2) Используя метод Ньютона, решить систему нелинейных уравнений с точностью до 0.001

## ***Работа №7***

Используя метод Горнера, найти один из корней уравнения с шестью значащими цифрами

**Методы решений и инструменты**

***Инструменты***

При решении лабораторной работы были использованы следующие инструменты:

1. Mathcad 15 (версия: M050 [], разрядность: x64);
2. Eclipse IDE (версия: 2022-09 (4.25.0), разрядность: x64);
3. Visual Studio IDE (версия: 17.7.3, разрядность: x64).

Mathcad 15 — это программное обеспечение для математического моделирования и анализа технических и научных данных. Оно предоставляет удобную среду для создания и решения математических выражений и уравнений, а также выполнения численных и символьных вычислений.

Eclipse IDE — это интегрированная среда разработки (IDE), используемая в компьютерном программировании. Оно содержит базовое рабочее пространство и расширяемую систему подключаемых модулей для настройки среды. Это вторая по популярности среда IDE для разработки на Java, и до 2016 года она была самой популярной. Eclipse написан в основном на Java и его основное применение заключается в разработке приложений Java, но он также может быть использован для разработки приложений на других языках программирования с помощью плагинов.

Visual Studio IDE- это интегрированная среда разработки (IDE) от компании Microsoft, предназначенная для разработки программного обеспечения. В Visual Studio предоставляются различные инструменты и функциональные возможности, упрощающие процесс разработки, отладки и тестирования приложений.

Visual Studio поддерживает множество языков программирования, включая C++, C#, Visual Basic, F#, JavaScript и другие. Для разработки на C++ в Visual Studio используется компилятор Microsoft C++, который обеспечивает мощные возможности компиляции и оптимизации кода.

Visual Studio обеспечивает разработчиков на C++ всеми необходимыми инструментами для создания высококачественного программного обеспечения. Отличительной чертой Visual Studio является его обширная функциональность и поддержка различных платформ и технологий, что делает его одним из популярных выборов для разработки на C++.

***Методы решений***

**Отделение корней аналитически**

Процесс отделения корней начинается с установления знаков функции f(x) в граничных точках a и b её существования. Затем определяются знаки функции f(x) в ряде промежуточных точек Если окажется, что , то в силу теоремы в интервале имеется корень уравнения .

Теорема. Если непрерывная функция f(x) принимает значения разных знаков на концах отрезка [а,b], т. е. , то внутри этого отрезка содержится по меньшей мере один корень уравнения , т. е. найдется хотя бы одно число , такое, что .

Корень будет единственным, если производная f ′(x) существует и сохраняет постоянный знак внутри интервала [а, b].

**Отделение корней графически**

Для отделения корней графически, необходимо в уравнении f1(x)= f2(x) построить на одной координатной плоскости функции f1(x) и f2(x). На полученных графиках отметить их точки пересечения друг с другом. Проведя перпендикуляр на ось x из точек пересечения, будут найдены корни уравнения.

**Уточнение корня методом проб**

Пусть дано уравнение и корень ξ отделён на отрезке [а, b], т.е. , причём .

На отрезке [а, b] берётся произвольным образом точка а1, которая разделит его на два отрезка [а, a1] и [a1, b]. Из этих двух отрезков следует выбрать тот, на концах которого функция принимает значения, противоположные знаку. Если , , то следует выбрать отрезок [a1, b], в противном случае [а, a1]. Повторяем этот процесс до тех пор, пока длина отрезка, на котором находится корень, не станет меньше . Корень ξ получается, как среднее арифметическое концов найденного отрезка, причём погрешность корня не превышает .

***Уточнение корня методом хорд***

Пусть дано уравнение , где f(x)-непрерывная функция, имеющая в интервале [а, b] производные первого и второго порядков. Корень считается отделённым на отрезке [а, b], т.е. .

Идея метода состоит в том, что на достаточно малом промежутке [а, b] дуга кривой заменяется стягивающей её хордой. В качестве приближенного значения корня принимается точка пересечения хорды с осью x.

Уравнение хорды AB имеет вид . Полагая и , получаем

.

Допустим, вторая производная f ′′(x) сохраняет постоянный знак, тогда можно выделить два случая:

, и ,. Случай сводится к рассматриваемому, если уравнение записать в форме . Первому случаю соответствует формула:

,

Второму случаю соответствует формула:

,

В первом случае остаётся неподвижным конец a, а во втором случае конец b.

***Уточнение корня методом касательных (Ньютона)***

Пусть дано уравнение , где f(x)-непрерывная функция, имеющая в интервале [а, b] производные первого и второго порядков. Корень считается отделённым на отрезке [а, b], т.е. .

Найдя какое-нибудь n приближенное значение корня , мы можем уточнить его по методу касательных (Ньютона) следующим образом. Положим

,

где считаем малой величиной. Отсюда, применяя формулу Тейлора, получим .Следовательно, .

Внеся эту поправку в формулу, находится следующее по порядку приближение корня

.

**Комбинированный метод хорд и касательных**

Пусть дано уравнение и корень ξ отделён на отрезке [а, b].

Если , то метод хорд даёт приближения корня с недостатком, а метод касательных- с избытком.

Если же , то метод хорд даёт приближения корня с избытком, а метод касательных- с недостатком.

Однако, во всех случаях истинный корень заключен между приближенными корнями, т.е. где -приближенное значение корня с недостатком, -приближенное значение корня с избытком.

Вычисления следует вести в таком порядке. Если , то со стороны конца a лежат приближенные значения корня, полученные по методу хорд, а со стороны конца b – значения, полученные по методу касательных, тогда

.

В общем виде записывается:

.

Если же , то со стороны конца a лежат приближенные значения корня, полученные по методу касательных, а со стороны конца b – значения, полученные по методу хорд, тогда

.

В общем виде записывается:

.

Процесс вычисления прекращается, как только станет выполняться неравенство .

За приближенное значение корня следует принять

.

***Уточнение корня методом итераций***

Пусть дано уравнение и корень ξ отделён на отрезке [а, b].

Заменим равносильным ему уравнением

. (1)

Каким-либо способом выбирается и подставим его в правую часть уравнения (1); тогда получается .

Повторяя этот процесс, получается последовательность чисел . Здесь могут встретиться два случая:

1). Последовательность x0, x1,…, xn,… сходится, т.е. имеет предел и тогда этот предел будет корнем уравнения .

2). Последовательность x0, x1,…, xn,… не сходится, т.е. не имеет предел.

Условие сходимости итерационного процесса: пусть на отрезке [a,b] имеется единственный корень уравнения и во всех точках этого отрезка производная удовлетворяет неравенству . Если при этом выполняется и условие , тол итерационный процесс сходится, а за нулевое приближение можно взять любое число из отрезка [a,b].

Пусть – точно значение корня уравнения , а число q определяется из соотношения . Тогда справедливо соотношение:

.

Если поставить условие, что истинное значение корня должно отличаться от приближённого значения на величину ε, т.е., то приближения x0,x1,…,xn надо вычислять до тех пор, пока не будет выполнено неравенство ,

или

.

**Метод Ньютона для системы нелинейных уравнений**

Пусть дана система

(2)

где f и φ – непрерывно дифференцируемые функции. Предположим, что известны n приближения неизвестных, тогда за более точные их значения можно принять

,

Тогда система (2) запишется в виде

Разложив функции f и φ в ряд Тейлора по степеням hn и kn получим

или

откуда

Таким образом, находим

**Метод итераций для системы нелинейных уравнений**

Пусть дана система

(3)

Для удобства система записывается в виде

(4)

Пусть x0 и y0 -начальные приближения корней, полученные графически или каким-либо другим способом. Подставив в правые части уравнений системы (4) вместо x и y начальные значения x0 и y0, получим

В общем случае

Если функции и непрерывны и последовательности x0, x1,…, xn,… и y0, y1,…, yn,… сходятся, то пределы их дают решение системы (4), а следовательно, и системы (3).

Пусть в некоторой замкнутой окрестности имеется одно и только одно решение и системы (4).

Тогда если:

1). функции и определены и непрерывно дифференцируемы в R;

2). начальные приближения x0, y0 и все последующие приближения xn, yn (n=1, 2, …) принадлежат R;

3). в R выполнены неравенства

или неравенства

то процесс последовательных приближений сходится к решению

Оценка погрешности n-го приближения определяется неравенством

где M- наибольшее из чисел q1 и q2.

**Метод Горнера**

Пусть дам многочлен вида .

Можно представить в таком виде ,

где

,

т.е.

.

Раскрыв скобки, получается

.

В левой и правой частях неравенства находятся многочлены от переменной x. Известно, что два многочлена считаются равными, если равны их коэффициенты при одинаковых степенях неизвестного x. Поэтому:

Как видно из последних равенств, каждый последующий коэффициент получается умножением предыдущего на и прибавлением соответствующего коэффициента ; так же находится и остаток r.

Вычисления удобно располагать по следующей схеме:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  | … |  |  |
|  | … |  | … |  |  |
|  |  |  | … |  |  |

В первой строке выписывают коэффициенты многочлена . В третью строку сразу же переносят . Затем каждый коэффициент умножают на и подписывают под следующим коэффициентом . Числа первой и второй строки складывают и результат записывают в третьей строке. Таким образов, в третьей строке записаны коэффициенты частного, полученного от деления многочлена на двучлен , и остаток, численно равный значению этого многочлена при , т.е. .

**Таблица идентификаторов**

***Работа 1***

Таблица 1

Таблица идентификаторов, переменных, используемых для решения задачи

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название переменной | Excel | Mathcad | C++ | Python | Комментарий |
| i |  | i | i | i | Итерации |
| a |  | a | a | a | Нижняя граница поиска корней |
| b |  | b | b | b | Верхняя граница поиска корней |
| y |  | y,y1 | y,y1 | y,y1 | уравнеине |
| n |  | n,n1,n2 | n,n1,n2 | n,n1,n2 | Количество шагов |
| Zadanie |  | Zadanie11,  Zadanie 12 | Zadanie 11,  Zadanie 12 | Zadanie 11,  Zadanie 12 | Функция решающая выбранное задание |
| h |  | h,h1,h2 | h,h1,h2 | h,h1,h2 | Шаг |
| SL |  | SL | SL | SL | Сумма для левых прямоугольников |
| SP |  | SP | SP | SP | Сумма для правых прямоугольников |
| I |  | I.I1I2 | I.I1.I2 | I.I1.I2 | Возвращение приближённого интеграла |

***Работа 2***

Таблица 2

Таблица идентификаторов, переменных, используемых для решения задачи

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название переменной | Excel | Mathcad | C++ | Python | Комментарий |
| i |  | i | i | i | Итерации |
| a |  | a | a | a | Нижняя граница интеграла |
| b |  | b | b | b | Верхняя граница интеграла |
| y |  | y | y | y | Подынтегральная функция |
| n |  | n | n | n | Количество шагов |
| Zadanie |  | Zadanie 21,  Zadanie 22 | Zadanie 21,  Zadanie 22 | Zadanie 21,  Zadanie 22 | Функция решающая выбранное задание |
| h |  | h,h1,h2 | h,h1,h2 | h,h1,h2 | Шаг |
| S |  | S1,S2,S3 | S1,S2,S3 | S1,S2,S3 | Суммирование значений функции для каждого массива |
| d |  | d0,d1,d2,  d3,d4 | d0,d1,d2,  d3,d4 | d0,d1,d2,  d3,d4 | Массивы с погрешностями |
| Δl |  | dl | dl | dl | Оценка погрешности |
| I |  | I | I | I | Возвращение приближённого интеграла |

***Работа 3***

Таблица 3

Таблица идентификаторов, переменных, используемых для решения задачи

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название переменной | Excel | Mathcad | C++ | Python | Комментарий |
| i |  | i | i | i | Итерации |
| a |  | a | a | a | Нижняя граница интеграла |
| b |  | b | b | b | Верхняя граница интеграла |
| y |  | y | y | y | Подынтегральная функция |
| n |  | n,n1,n2 | n,n1,n2 | n,n1,n2 | Количество шагов |
| Zadanie |  | Zadanie 3 | Zadanie 3 | Zadanie 3 | Функция решающая выбранное задание |
| h |  | h,h1,h2 | h,h1,h2 | h,h1,h2 | Шаг |
| S |  | S1,S2,S3 | S1,S2,S3 | S1,S2,S3 | Суммирование значений функции для каждого массива |
| SP |  | SP | SP | SP | Сумма для правых прямоугольников |
| I |  | I1,I2 | I1,I2 | I1,I2 | Возвращение приближённого интеграла |

***Работа 4***

Таблица 4

Таблица идентификаторов, переменных, используемых для решения задачи

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название переменной | Excel | Mathcad | C++ | Python | Комментарий |
| i |  | i | i | i | Итерации |
| a |  | a | a | a | Нижняя граница интеграла |
| b |  | b | b | b | Верхняя граница интеграла |
| k |  | k | k | k | Номер варианта |
| c |  | c | c | c | Просто переменная |
| y |  | y | y | y | Подынтегральная функция |
| n |  | n,n1,n2 | n,n1,n2 | n,n1,n2 | Количество шагов |
| Zadanie |  | Zadanie 41,  Zadanie 42 | Zadanie 41,  Zadanie 42 | Zadanie 41,  Zadanie 42 | Функция решающая выбранное задание |
| h |  | h,h1,h2 | h,h1,h2 | h,h1,h2 | Шаг |
| S |  | S1,S2,S3 | S1,S2,S3 | S1,S2,S3 | Суммирование значений функции для каждого массива |
| SP |  | SP | SP | SP | Сумма для правых прямоугольников |
| I |  | I1,I2,I3 | I1,I2,I3 | I1,I2,I3 | Возвращение приближённого интеграла |

***Работа 5***

Таблица 5

Таблица идентификаторов, переменных, используемых для решения задачи

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название переменной | Excel | Mathcad | C++ | Python | Комментарий |
| i |  | i | i | i | Итерации |
| a |  | a | a | a | Нижняя граница интеграла |
| b |  | b | b | b | Верхняя граница интеграла |
| k |  | k | k | k | Номер варианта |
| c |  | c | c | c | Просто переменная |
| y |  | y | y | y | Подынтегральная функция |
| t,C |  | t1,t2, C1,C2 | t1,t2, C1,C2 | t1,t2, C1,C2 | Квадратурные коэффициента Гаусса |
| n |  | n,n1,n2 | n,n1,n2 | n,n1,n2 | Количество шагов |
| Zadanie |  | Zadanie 5 | Zadanie 5 | Zadanie 5 | Функция решающая выбранное задание |
| h |  | h,h1,h2 | h,h1,h2 | h,h1,h2 | Шаг |
| S |  | S1,S2,S3 | S1,S2,S3 | S1,S2,S3 | Суммирование значений функции для каждого массива |
| SP |  | SP | SP | SP | Сумма для правых прямоугольников |
| I |  | I1,I2 | I1,I2 | I1,I2 | Возвращение приближённого интеграла |

***Работа 6***

Таблица 6

Таблица идентификаторов, переменных, используемых для решения задачи

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название переменной | Excel | Mathcad | C++ | Python | Комментарий |
| i |  | i | i | i | Итерации |
| a |  | a | a | a | Нижняя граница интеграла |
| b |  | b | b | b | Верхняя граница интеграла |
| k |  | k | k | k | Номер варианта |
| c |  | c | c | c | Просто переменная |
| y |  | y | y | y | Подынтегральная функция |
| t,C |  | t1,t2, C1,C2 | t1,t2, C1,C2 | t1,t2, C1,C2 | Квадратурные коэффициента Гаусса |
| n |  | n,n1,n2 | n,n1,n2 | n,n1,n2 | Количество шагов |
| Zadanie |  | Zadanie 5 | Zadanie 5 | Zadanie 5 | Функция решающая выбранное задание |
| h |  | h,h1,h2 | h,h1,h2 | h,h1,h2 | Шаг |
| S |  | S1,S2,S3 | S1,S2,S3 | S1,S2,S3 | Суммирование значений функции для каждого массива |
| SP |  | SP | SP | SP | Сумма для правых прямоугольников |
| I |  | I1,I2 | I1,I2 | I1,I2 | Возвращение приближённого интеграла |

***Работа 7***

Таблица 7

Таблица идентификаторов, переменных, используемых для решения задачи

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Название переменной | Excel | Mathcad | C++ | Python | Комментарий |
| i |  | i | i | i | Итерации |
| a |  | a | a | a | Нижняя граница интеграла |
| b |  | b | b | b | Верхняя граница интеграла |
| k |  | k | k | k | Номер варианта |
| c |  | c | c | c | Просто переменная |
| y |  | y | y | y | Подынтегральная функция |
| t,C |  | t1,t2, C1,C2 | t1,t2, C1,C2 | t1,t2, C1,C2 | Квадратурные коэффициента Гаусса |
| n |  | n,n1,n2 | n,n1,n2 | n,n1,n2 | Количество шагов |
| Zadanie |  | Zadanie 5 | Zadanie 5 | Zadanie 5 | Функция решающая выбранное задание |
| h |  | h,h1,h2 | h,h1,h2 | h,h1,h2 | Шаг |
| S |  | S1,S2,S3 | S1,S2,S3 | S1,S2,S3 | Суммирование значений функции для каждого массива |
| SP |  | SP | SP | SP | Сумма для правых прямоугольников |
| I |  | I1,I2 | I1,I2 | I1,I2 | Возвращение приближённого интеграла |

**Ручной счёт**

## ***Работа 1***

1. Исходные данные:

Обозначим . Находим производную

Вычислим корень производной:

x≠i;

Составим таблицу знаков функции , полагая равным: а) критическим значениям функции (корням производной) или близким к ним, б) граничным значениям (исходя из области допустимых значений неизвестного):

Таблица 8

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| x |  | - | i |  | 3 |  |
|  | + | + |  | + | + | - |

Так как происходит одна перемена знака функции, то уравнение имеет один действительный корень. Чтобы завершить операцию отделения корней, следует уменьшить промежутки, содержащие корни, так чтобы их длина была не больше 1. Для этого составим новую таблицу знаков функции :

Таблица 9

|  |  |  |
| --- | --- | --- |
| x | 3 | 4 |
|  | + | - |

Отсюда видно, что корень заключен в следующих промежутках:

2)Исходные данные:

Полагая , имеем .

Найдем корни производной:

Составим таблицу знаков функций

Таблица 10

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| x |  | -1 | 0 | 3 |  |
|  | + | - | + | - | + |

Из таблицы видно, что уравнение имеет четыре действительных корня: ,

Уменьшим промежутки, в которых находятся корни:

Таблица 11

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| x | -2 | -1 | 0 | 1 | 3 | 5 |
|  | + | - | + | - | - | + |

Следовательно, ,

Уточним один из корней, например , методом проб до сотых долей. Все вычисления удобно производить, используя следующую таблицу:

Таблица 12

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n |  |  |  |  |
| 0 | 3 | 5 | 4 | -30 |
| 1 | 4 | 5 | 4.5 | 138.6875 |
| 2 | 4 | 4.5 | 4.25 | 41.51172 |
| 3 | 4 | 4.25 | 4.125 | 2.797607 |
| 4 | 4 | 4.125 | 4.0625 | -14.3105 |
| 5 | 4.0625 | 4.125 | 4.09375 | -5.93751 |
| 6 | 4.09375 | 4.125 | 4.109375 | -1.61569 |
| 7 | 4.109375 | 4.125 | 4.117188 | 0.579465 |
| 8 | 4.109375 | 4.117188 | 4.113281 | -0.52098 |
| 9 | 4.113281 | 4.117188 | 4.115234 | 0.028526 |
| 10 | 4.113281 | 4.115234 | 4.114258 | -0.24641 |
| 11 | 4.114258 | 4.115234 | 4.114746 | -0.10898 |
| 12 | 4.114746 | 4.115234 | 4.11499 | -0.04024 |
| 13 | 4.11499 | 4.115234 | 4.115112 | -0.00586 |

Ответ:

3)Исходные данные: .

Обозначим:

Графики этих функций представлены на рисунке 1.

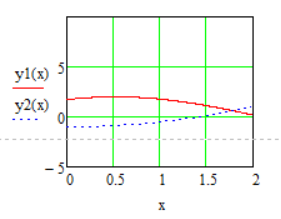


Рисунок 1.-Графический способ

Из графика видно, что уравнение имеет корень в промежутке .

4)Исходные данные: . Обозначим , , построим графики этих функций.

Графики представлены на рисунке 2.

Из графика видно, что уравнение имеет два корня

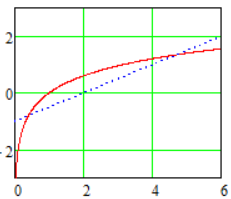


Рисунок 2.- Графический метод

Для уточнения этого корня методом проб выберем промежуток, концах которого функция имеет разные знаки   
Составим таблицу:

Таблица 13

|  |  |  |
| --- | --- | --- |
| x | 4 | 5 |
|  | + | - |

Дальнейшие вычисления производим в таблицe 14:

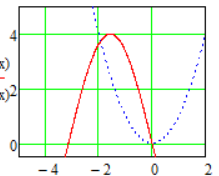
Таблица 14

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n |  |  |  | ) |
| 0 | 4 | 5 | 4.5 | 0.056425 |
| 1 | 4.5 | 5 | 4.75 | -0.02161 |
| 2 | 4.5 | 4.75 | 4.625 | 0.017723 |
| 3 | 4.625 | 4.75 | 4.6875 | -0.00187 |

Ответ:

## ***Работа 2***

1)Исходные данные: . Отделим корень графически. Построим графики функций и :



Из графика видно, что уравнение имеет 2 корня и

Чтобы уточнить корень методом хорд, определим знаки функции на концах промежутка и знак её второй производной в этом промежутке:

;

;

при .

Для вычислений применяем формулу

, где

Вычисления удобно располагать в таблице:

Таблица 16

|  |  |  |  |
| --- | --- | --- | --- |
| n |  |  |  |
| 0 | -1 | -1 | 4.365 |
| 1 | -2.09 | 0.091 | 7.842 |
| 2 | -1.996 | -0.004 | 7.626 |

Таблица 17

|  |  |  |  |
| --- | --- | --- | --- |
| n |  |  |  |
| 0 | 1 | -4.00307 | 1.091 |
| 1 | 4.370746 | -7.47954 | -0.095 |
| 2 | 3.982434 | -7.2641 | 0.005 |

Ответ:

2)Исходные данные: .

Отделим корни аналитически. Находим

Составим таблицу знаков функции :

Таблица 18

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x |  | 0 | 1 |  |
|  | - | - | + | + |

Уравнение имеет один действительный корень, лежащий в промежутке

Чтобы уточнить корень, находим вторую производную в промежутке выполняется неравенство .

Для вычислений применяем формулу

*,* где ; .

Вычисления располагаются в таблице:

Таблица 19

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| n |  |  |  |  |  |
| 0 | 1 | 1 | 1 | 1 | -0,9 |
| 1 | 0,9 | 0,729 | 0,099 | 0,9 | -0,8902 |
| 2 | 0,8902 | 0,7054 | 0,0105 | 0,8902 | -0,8891 |
| 3 | 0,8891 | 0,7029 | 0,0011 | 0,8891 | -0,8891 |
| 4 | 0,8890 | 0,7027 | 0,0001 | 0,8891 | -0,8890 |

Ответ: .

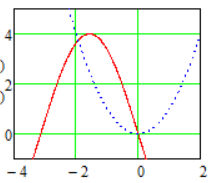
## ***Работа 3***

1)Исходные данные:. Отделим корень графически. Построим графики функций и , составив таблицы значений этих функций:

Таблица 15

Таблица идентификаторов, переменных, используемых для решения задачи

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| x | -2 | -1,8 | -1,6 | -1,4 | -1,2 | -1 |
|  | 4 | 3,24 | 2,56 | 1,96 | 1,44 | 1 |
|  | 3,64 | 3,90 | 4,00 | 3,94 | 3,73 | 3,37 |



Таким образом, положительный корень уравнения заключен в промежутке .

;

;

при .

Для вычислений применяем формулу

,где

Вычисления удобно располагать в таблице:

Таблица 16

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| n | xn | -4sin(xn) | xn^2 | F(xn) | F(xn)/F'(x0) |
| 1 | -2 | -3,637 | 4 | 0,3628 | -0,064 |
| 2 | -1,936 | -3,736 | 3,7479 | 0,012 | -0,002 |
| 3 | -1,9339 | -3,7392 | 3,73996 | 0,00076 | -0,0001 |

Ответ:

2)Исходные данные: .

Отделим корни аналитически. Находим

Составим таблицу знаков функции :

Таблица 18

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x |  | 0 | 1 |  |
|  | - | - | + | + |

Уравнение имеет один действительный корень, лежащий в промежутке

Чтобы уточнить корень, находим вторую производную в промежутке выполняется неравенство .

Для вычислений применяем формулу

*,* где ; .

Вычисления располагаем в таблице:

Таблица 19

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| n | xn | xn^3 | F(xn) | F'(xn) | h |
| 1 | 1 | 1 | 1 | 9 | 0,111111111 |
| 2 | 0,888 | 0,702332 | -0,001 | 9,037037 | -0,000151791 |
| 3 | 0,889 | 0,702692 | 0 | 9,036936 | -8,49479E-10 |

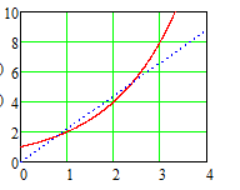
Ответ: .

# Работа 5

Исходные данные:

Построим графики функций по таблице.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| x | 0 | 0,5 | 1 | 1,5 | 2 | 2,5 | 3 |
| y1 | 1 | 1,414214 | 2 | 2,828427 | 4 | 5,656854 | 8 |
| y2 | 0 | 1,1 | 2,2 | 3,3 | 4,4 | 5,5 | 6,6 |



Из графика видно, что уравнение имеет два действительных корня, заключенных в промежутке

на промежутке

Возьмем k=2

За начальное приближение возьмем , остальные приближения определим с помощью формулы (13). Результаты занесем в таблицу.

|  |  |  |  |
| --- | --- | --- | --- |
| n | xn | F(xn) | a(xn) |
| 0 | 0 | -1 | -0,5 |
| 1 | 0,5 | -0,31421 | -0,15711 |
| 2 | 0,657107 | -0,13128 | -0,06564 |
| 3 | 0,722748 | -0,06028 | -0,03014 |
| 4 | 0,752886 | -0,02881 | -0,01441 |
| 5 | 0,767292 | -0,01403 | -0,00701 |
| 6 | 0,774307 | -0,00689 | -0,00345 |
| 7 | 0,777753 | -0,0034 | -0,0017 |
| 8 | 0,779454 | -0,00168 | -0,00084 |
| 9 | 0,780295 | -0,00083 | -0,00042 |

2) Исходные данные:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| x | -*ꭃ* | 0 | 1 | +*ꭃ* |
| signf(x) | - | - | + | + |

Уравнение имеет один действительный корень в промежутке

на промежутке

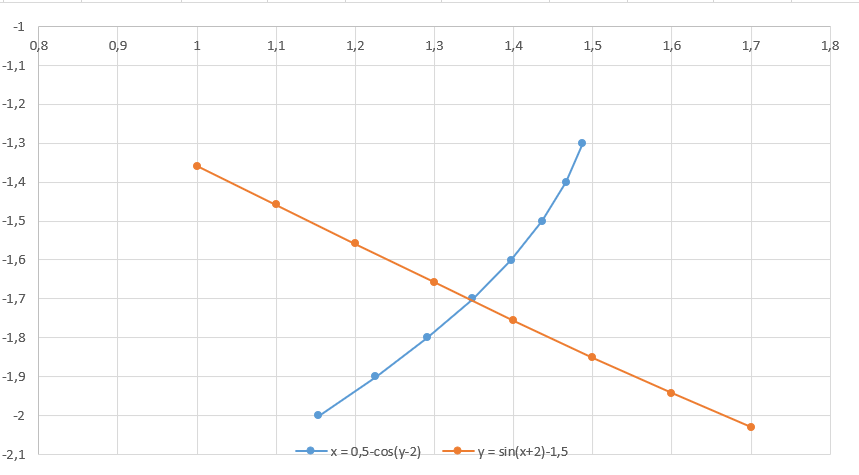
возьмем k=4

Составим таблицу и по формуле (14) найдем значения

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| n | xn | xn^3 | -0,1\*xn^2 | F(xn) | a(xn) |
| 0 | 0 | 0 | 0 | -1 | 0,25 |
| 1 | 0,25 | 0,015625 | -0,0125 | -0,87188 | 0,467969 |
| 2 | 0,467969 | 0,102483 | -0,0438 | -0,70733 | 0,644802 |
| 3 | 0,644802 | 0,268089 | -0,08315 | -0,49266 | 0,767968 |
| 4 | 0,767968 | 0,452928 | -0,11795 | -0,28104 | 0,838229 |
| 5 | 0,838229 | 0,588962 | -0,14053 | -0,13245 | 0,871341 |
| 6 | 0,871341 | 0,661552 | -0,15185 | -0,05462 | 0,884997 |
| 7 | 0,884997 | 0,693147 | -0,15664 | -0,021 | 0,890247 |
| 8 | 0,890247 | 0,705555 | -0,15851 | -0,00783 | 0,892204 |
| 9 | 0,892204 | 0,710219 | -0,15921 | -0,00288 | 0,892925 |
| 10 | 0,892925 | 0,711943 | -0,15946 | -0,00106 | 0,89319 |
| 11 | 0,89319 | 0,712575 | -0,15956 | -0,00039 | 0,893286 |

## ***Работа 6***

Перепишем систему в виде и построим графики.



Из графика видно, что система имеет одно решение, заключенное в области D

D: ;

Убедимся в том, что метод итераций применим для решения системы, перепишем ее в виде.

Так как , , то в области D имеем

Условия сходимости выполняются. Вычисления производим по формулам:

За начальное приближение принимаем

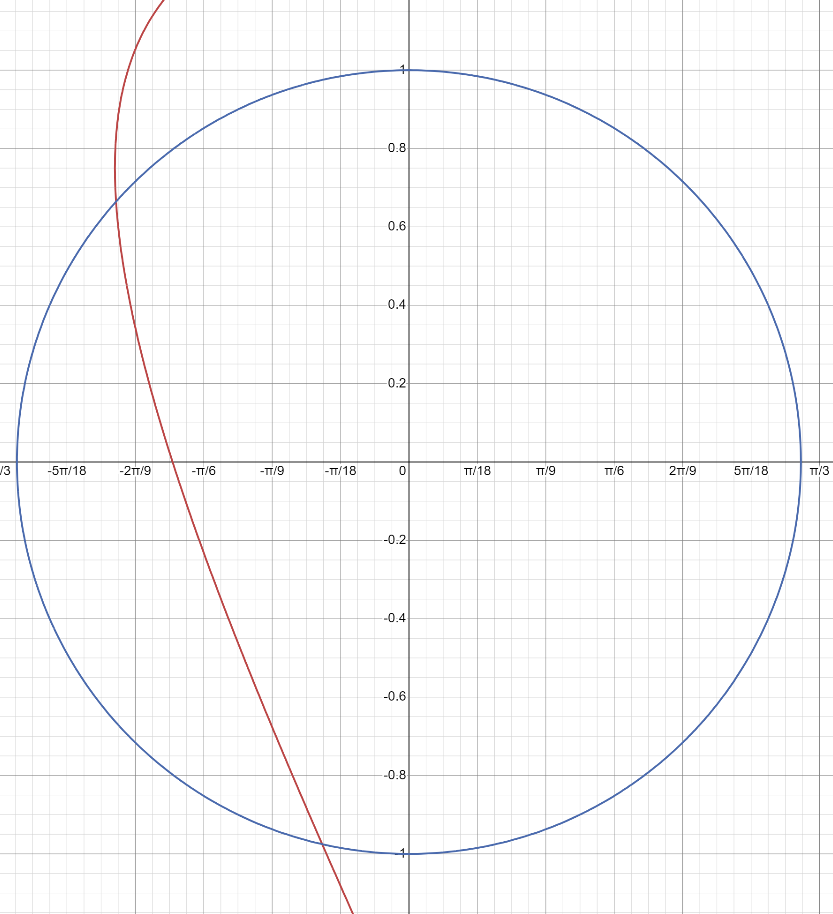
Вычисления удобно располагать в таблице.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| n | xn | yn | sin(x+2) | cos(y-2) |
| 1 | 1,2 | -1,6 | -0,05837 | -0,896 |
| 2 | 1,396 | -1,558374 | -0,25241 | -0,914 |
| 3 | 1,414 | -1,752406 | -0,26943 | -0,819 |
| 4 | 1,319 | -1,769433 | -0,17666 | -0,809 |
| 5 | 1,309 | -1,676657 | -0,16692 | -0,8602 |
| 6 | 1,360 | -1,66692 | -0,21691 | -0,865 |
| 7 | 1,365 | -1,716905 | -0,22171 | -0,839 |
| 8 | 1,339 | -1,721709 | -0,19615 | -0,8363 |
| 9 | 1,336 | -1,69615 | -0,19358 | -0,850 |
| 10 | 1,350 | -1,693576 | -0,20703 | -0,854 |
| 11 | 1,351 | -1,707033 | -0,20836 | -0,844 |
| 12 | 1,344 | -1,708356 | -0,20137 | -0,8434 |
| 13 | 1,3436 | -1,701374 | -0,20068 | -0,847 |
| 14 | 1,347 | -1,700679 | -0,20433 | -0,8477 |
| 15 | 1,347 | -1,70433 | -0,20469 | -0,8457 |
| 16 | 1,345 | -1,704691 | -0,20279 | -0,8456 |
| 17 | 1,345 | -1,702789 | -0,2026 | -0,8466 |
| 18 | 1,347 | -1,702601 | -0,20359 | -0,8467 |
| 19 | 1,3467 | -1,703593 | -0,20369 | -0,8461 |
| 20 | 1,3461 | -1,703691 | -0,20317 | -0,8461 |
| 21 | 1,3461 | -1,703174 |  |  |

,

2)

Отделение корней произведем графически.



Значения x берем исходя из следующих условий: -1<=1,2x<=1 и -1<x<1

Получаем неравенство:

Значения y берем исходя из следующих условий:

Получаем неравенство:

Система имеет два решения. Уточним одно из них, принадлежащее области:0.4, За начальное приближение примем

Имеем:

C помощью формулы (15) находим приближение по методу Ньютона

Вычисления удобно располагать в таблице.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| n | xn | xn^2 | xn+yn | sin(xn+yn) | F(xn,yn) | F'x(xn,yn) | F'y(xn,yn) |  |  |  |
| yn | yn^2 | cos(xn+yn) | G(xn,yn) | G'x(xn,yn) | G'y(xn,yn) |  |  |
| 0 | -1 | 1 | 1,40000 | -0,9854497 | 0,11455027 | -1,03003 | 0,1699671 | -1,72399 | 0,118835 | 0,102095 |
| -0,4 | 0,16 | 0,16996714 | 0,16 | -2 | -0,8 | -0,064295 | -0,05524 |
| 1 | -0,8979047 | 0,8062328 | 1,41010 | -0,9764069 | 0,001078703 | -0,98406 | 0,2159389 | -1,72668 | 0,0038919 | 0,003032 |
| -0,4552384 | 0,207242 | 0,21593889 | 0,013474737 | -1,79581 | -0,910477 | 0,0113228 | 0,00882 |
| 2 | -0,894873 | 0,8007977 | 1,41004 | -0,9737791 | 0 | -0,9725 | 0,2274956 | -1,72665 | 0 | 0 |
| -0,4464183 | 0,1992893 | 0,22749558 | 0 | -1,78975 | -0,892837 | 0 | 0 |

## ***Работа 7***

Для отделения корней воспользуемся аналитическим способом:

X1= x2=

Составим таблицу знаков функции F(x):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| X | -ꭃ | -32 | -31 | +ꭃ |
| Sign x | - | - | + | + |

Из таблицы видно, что уравнение 1корень на промежутке [-32;-31]

Для уточнения корня по методу Горнера следует предварительно преобразовать уравнение с помощью подстановки x=-y. В результате получим уравнение: φ(x)=

Искомый корень этого уравнения [-32;-31]отсюда следует, что

Составим уравнение, с помощью которого определяется следующая цифра корня, для чего выполним деление по схеме Горнера:

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | -31 | 0 | -26 |
| 1 | 0 | 0 | -26 |
| 1 | 31 | 961 |  |
| 1 | 62 |  |  |

В результате получим уравнение:

С1=0

|  |  |  |  |
| --- | --- | --- | --- |
| 1 | 620 | 96100 | -26000 |
| 1 | 620 | 96100 |  |
| 1 | 620 |  |  |

Таким образом,

Разделив модуль свободного члена на коэффициент при y, получим следующие три цифры искомого числа:

В результате находим корень уравнения:

# **Реализация в excel**

## ***Работа 1***



Рисунок 18.-Работа 1 задание 2 и 4



Рисунок 19.-Работа 1 задание 2 и 4

## ***Работа 2***



Рисунок 20.-Работа 2 задание 1 и 2



Рисунок 21.-Работа 2 задание 1 и 2

## ***Работа 3***



Рисунок 22.-Работа 3 задание 1 и 2



Рисунок 23.-Работа 3 задание 1 и 2

## ***Работа 5***



Рисунок 24.-Работа 5 задание 1 и 2



Рисунок 25.-Работа 5 задание 1 и 2

## ***Работа 6***



Рисунок 26.-Работа 6 задание 1 и 2



Рисунок 27.-Работа 6 задание 1 и 2

# **Реализация в Mathcad**

## ***Работа 1***

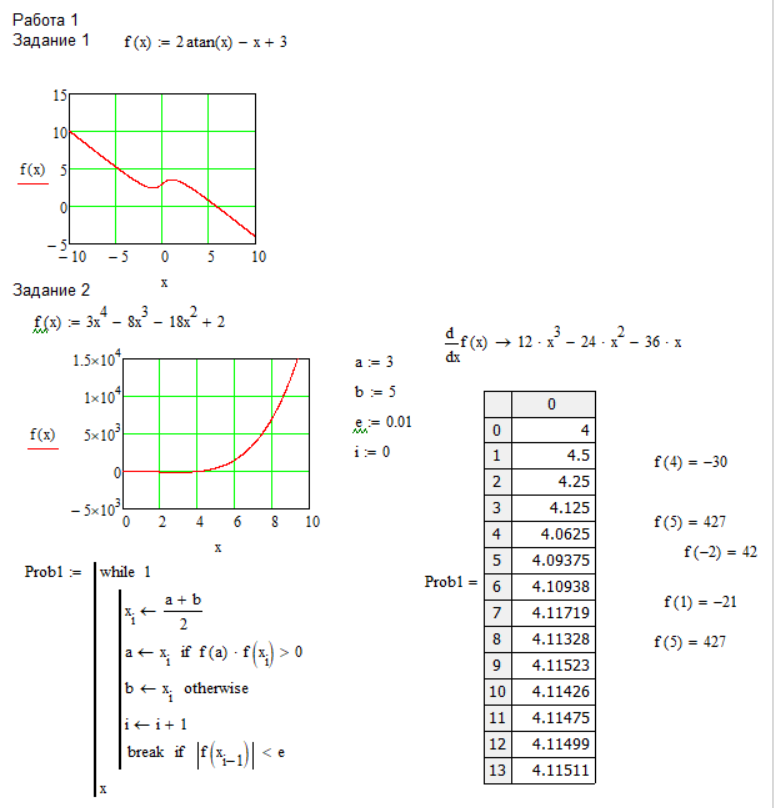


Рисунок 27.-Работа 1 задание 1 и 2

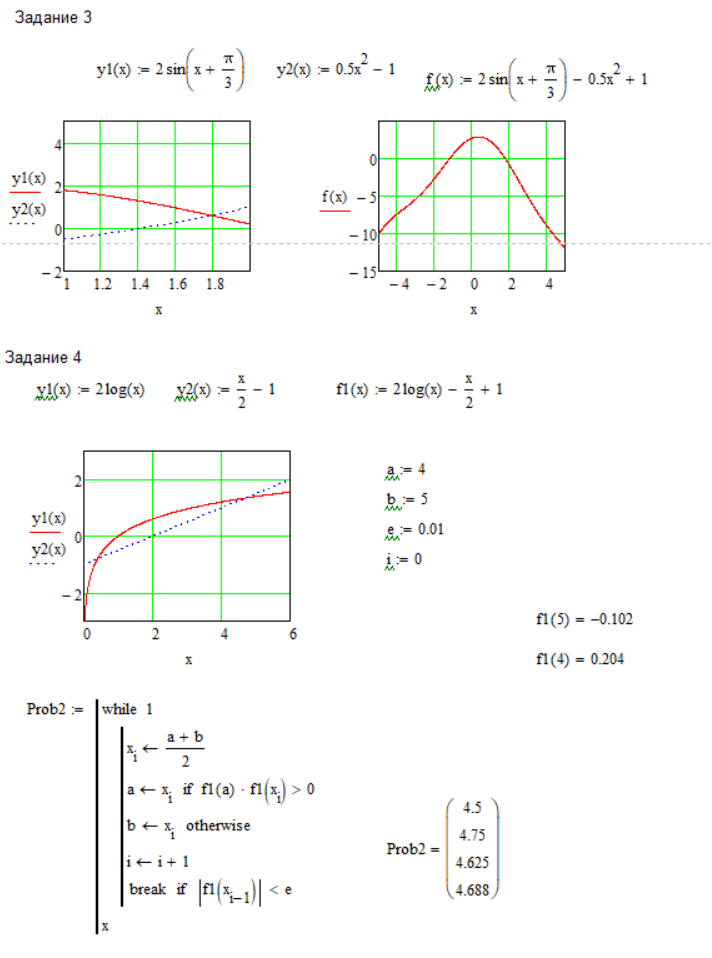


Рисунок 28.-Работа 1 задание 3 и 4

## ***Работа 2***

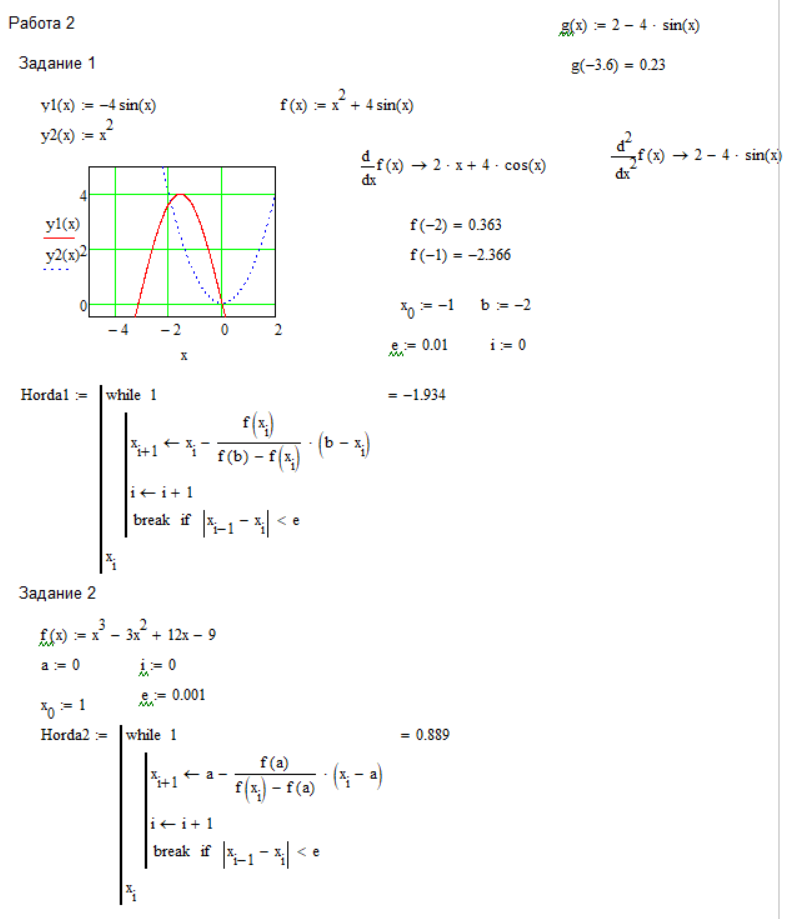


Рисунок 29.-Работа 2 задание 1 и 2

## ***Работа 3***

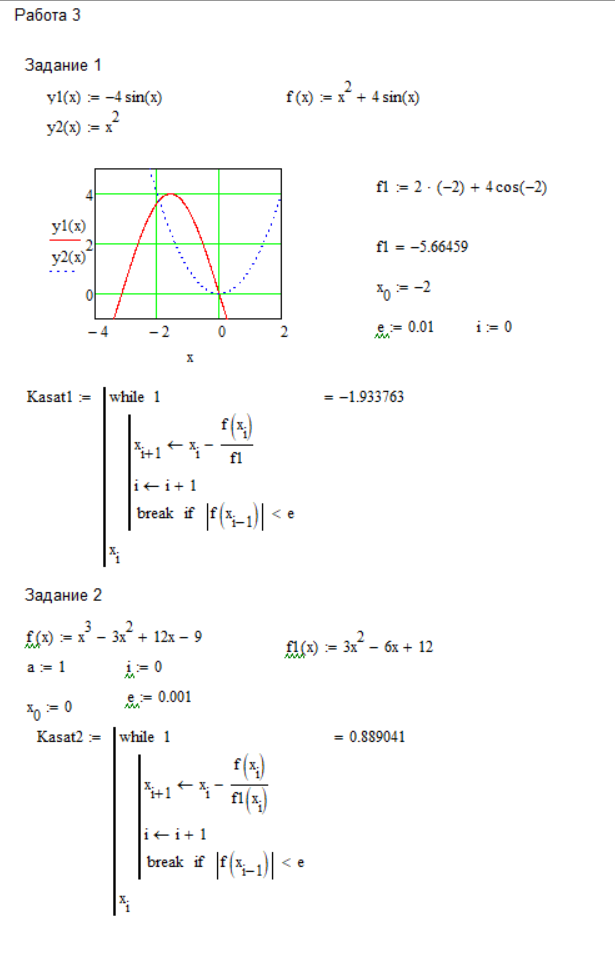


Рисунок 30.-Работа 3 задание 1 и 2

## ***Работа 4***

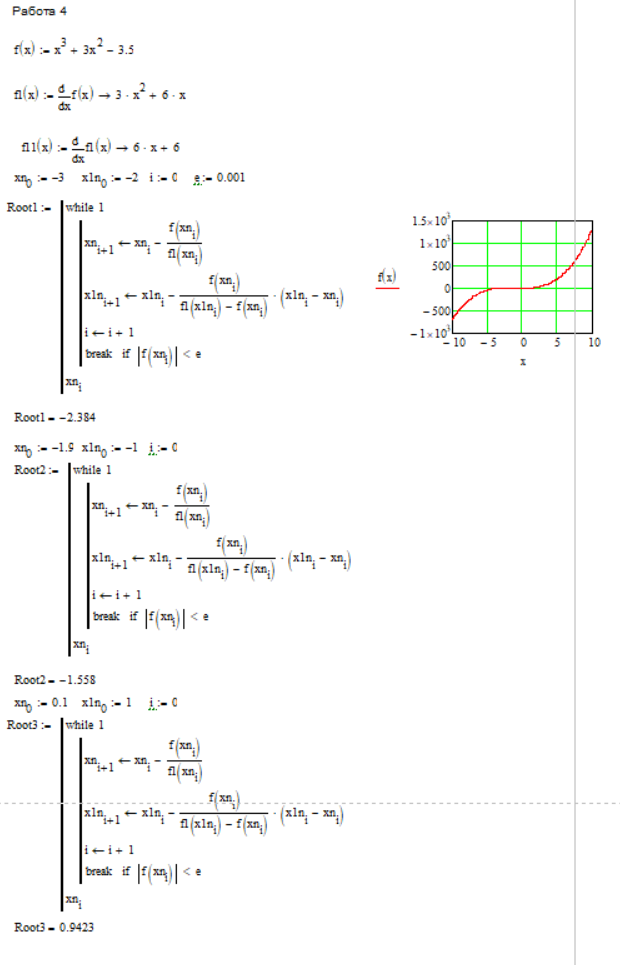


Рисунок 31.-Работа 4

## ***Работа 5***

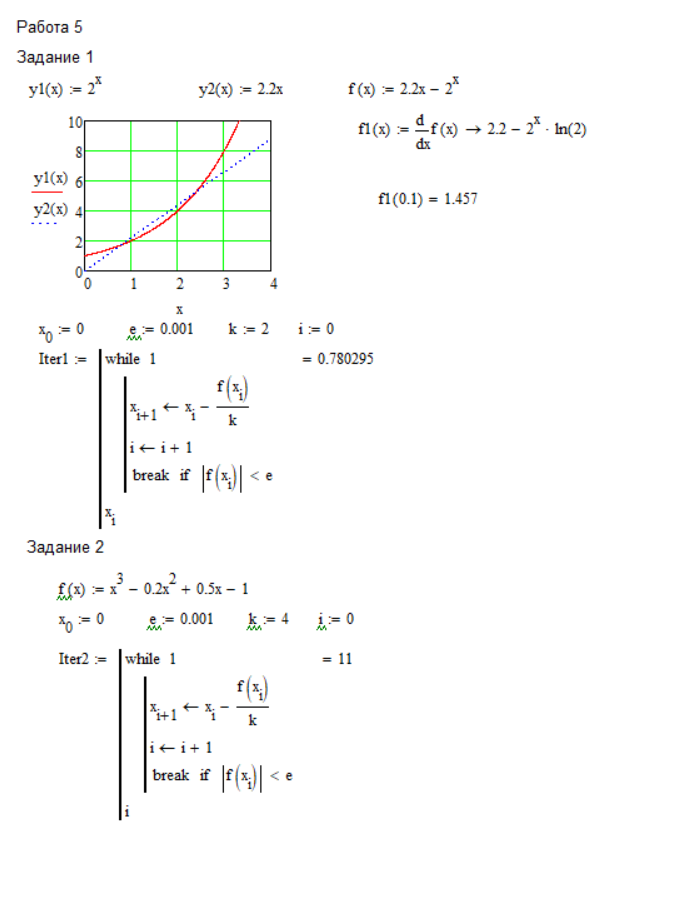


Рисунок 32.-Работа 5 задание 1 и 2

## ***Работа 6***

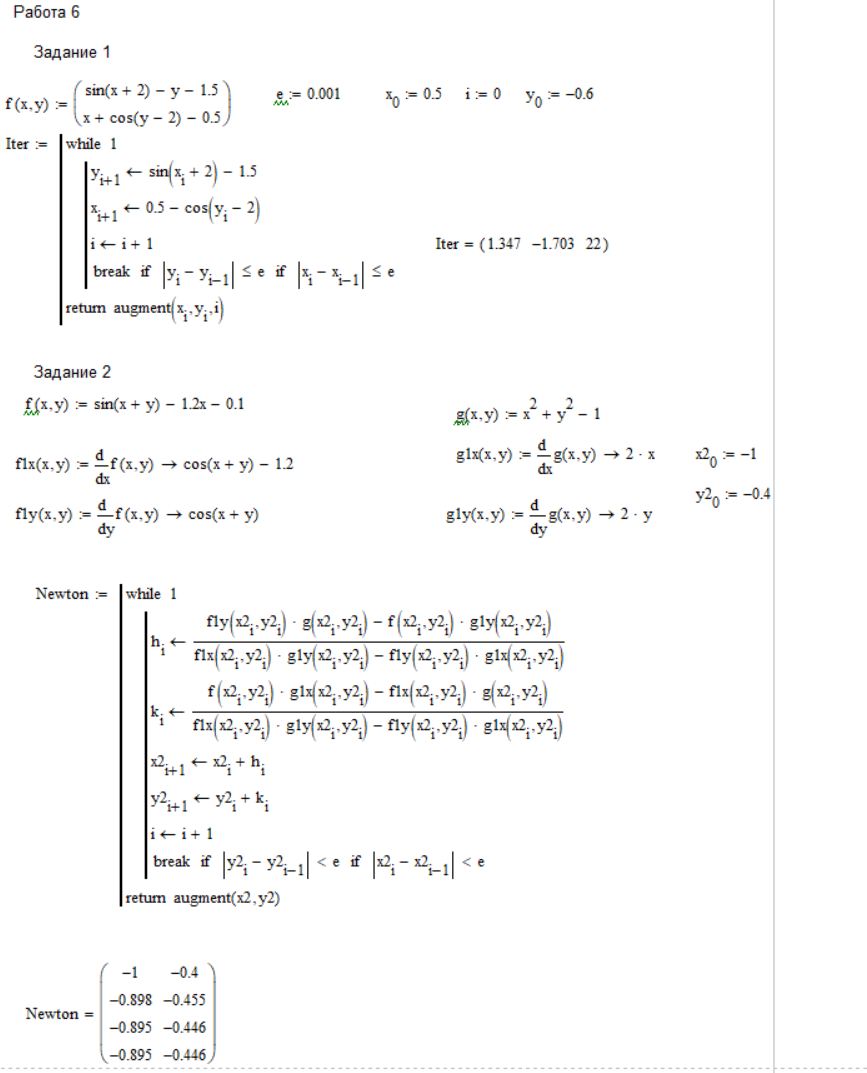
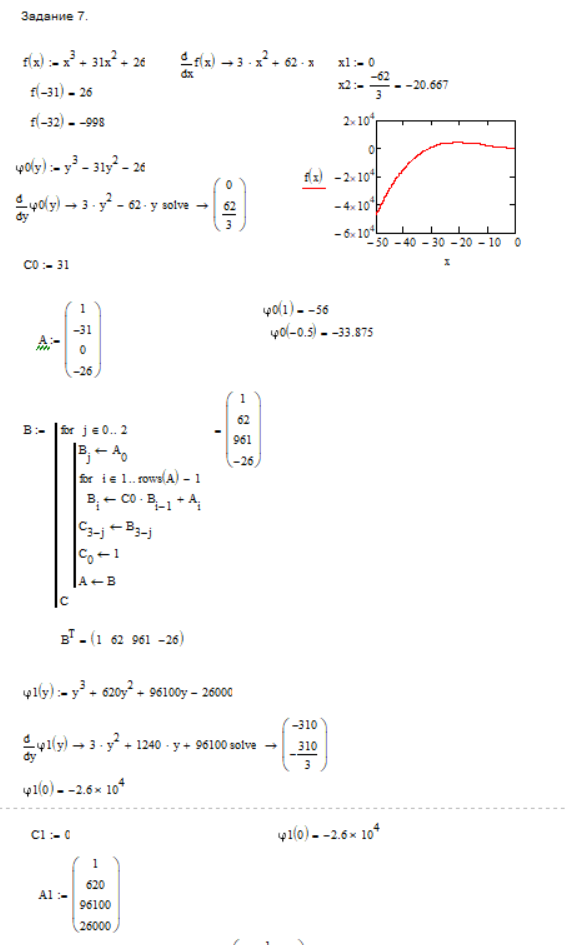


Рисунок 33.-Работа 6 задание 1 и 2

## ***Работа 7***



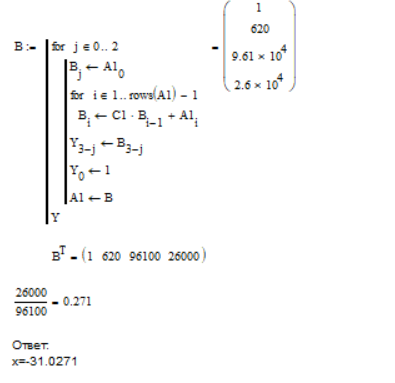


Рисунок 33.-Работа 7

# **Реализация в С++**

## ***Меню пользователя***

#include <iostream>

#include "Zadanie1.h"

#include "Zadanie2.h"

#include "Zadanie3.h"

#include "Zadanie4.h"

#include "Zadanie5.h"

#include "Zadanie6.h"

#include "Zadanie7.h"

using namespace std;

int main() {

setlocale(LC\_ALL, "rus");

Zadanie1 Zadanie1;

Zadanie2 Zadanie2;

Zadanie3 Zadanie3;

Zadanie4 Zadanie4;

Zadanie5 Zadanie5;

Zadanie6 Zadanie6;

Zadanie7 Zadanie7;

//task1 ts;

//ts.run();

cout << "Введите номер работы (с 1 по 7)\n";

cout << "8- завершение работы\n";

int choice;

int exit = 0;

do {

cin >> choice;

switch (choice) {

case 1:

Zadanie1.run();

break;

case 2:

Zadanie2.run();

break;

case 3:

Zadanie3.run();

break;

case 4:

Zadanie4.run();

break;

case 5:

Zadanie5.run();

break;

case 6:

Zadanie6.run();

break;

case 7:

Zadanie7.run();

break;

case 8:

cout << "Завершение работы.\n\n";

cout << "Обратная связь:\n";

cout << "Студент гр. 21-САИ, Краличев Игорь Евгеньевич, ikralichev@list.ru\n";

exit = 1;

break;

}

} while (exit == 0);

}

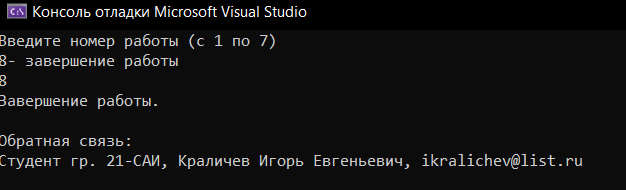


Рисунок 34.-Меню выбора задания

## ***Класс для работы 1***

#include <iostream>

#include <math.h>

using namespace std;

class Zadanie1 {

public:

double f(double x, int tasknum) {

if (tasknum == 1) return 2 \* atan(x) -x+3;

if (tasknum == 2) return 3 \* pow(x, 4) -8 \* pow(x, 3) - 18 \* pow(x, 2) +2;

if (tasknum == 3) return 2\*sin(x+3.14/3.0)-0.5\*x\*x+1;

if (tasknum == 4) 2\*log10(x)-x/2+1;

//return 2 \* exp(x) + 3 \* x + 1;

//return 3 \* pow(x, 4) + 4 \* pow(x, 3) - 12 \* pow(x, 2) - 5;

//return x \* (log(x + 1) / log(3)) - 2;

//return cos(x + 0.3) - x \* x;

}

void precise(double x, double e, int tasknum) {

double h = 0.001;

double a = x - h;

double b = x + h;

double c;

while (fabs(b - a) > e) {

c = (a + b) / 2;

if (f(a, tasknum) \* f(c, tasknum) > 0) a = c;

else b = c;

}

cout << "Уточненный корень x= " << (a + b) / 2 << "\n";

}

void analitic(double\* y, double\* x, double a, double b, double h, double e, bool key, int tasknum) {

int j = 0;

for (double i = a; i < b + h; i = i + h) {

//cout << "i = " << i << ", f(i) = " << f(i) << "\n";

if (fabs(f(i, tasknum)) < e) {

if (j > 0) {

if (fabs(x[j - 1] - i) < 1) continue;

}

x[j] = i;

y[j] = f(i, tasknum);

cout << "x[" << j << "] находится на промежутке: [" << x[j] - h << "; " << x[j] + h << "]\n";

j++;

}

}

if (key == 1) precise(x[0], e, tasknum);

//cout << "проверка = " << f(x[0]) << "\n";

}

void graphic(double\* y, double\* x, double a, double b, double h, double e, bool key, int tasknum) {

int j = 0;

for (double i = a; i < b + h; i = i + h) {

//cout << "i = " << i << ", f(i) = " << f(i) << "\n";

if (fabs(f(i, tasknum)) < e) {

if (j > 0) {

if (fabs(x[j - 1] - i) < 1) continue;

}

x[j] = i;

y[j] = f(i, tasknum);

cout << "x[" << j << "] находится на промежутке: [" << x[j] - h << "; " << x[j] + h << "]\n";

j++;

}

}

if (key == 1) precise(x[0], e, tasknum);

//cout << "проверка = " << f(x[0]) << "\n";

}

void run() {

// setlocale(LC\_ALL, "rus");

// объявление и инициализация поинтера на функцию

double a = -5, b = 5, h = 0.001;

const int n = 10;

const double e = 0.01;

double\* y = new double[n];

double\* x = new double[n];

//graphic(y, x, a, b, h, e, 1, 3);

cout << "Введите номер задания(с 1 по 4): \n";

int input;

cin >> input;

while (input == 1 || input == 2 || input == 3 || input == 4) {

if (input == 1) {

analitic(y, x, a, b, h, e, 0, 1);

}

if (input == 2) {

analitic(y, x, a, b, h, e, 1, 2);

}

if (input == 3) {

graphic(y, x, a, b, h, e, 0, 3);

}

if (input == 4) {

graphic(y, x, a, b, h, e, 1, 4);

}

cout << "введите не номер задания для выхода \n";

cin >> input;

}

cout << "выход\n";

}

};

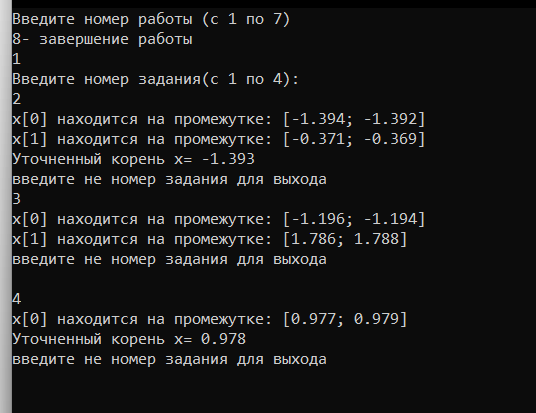


Рисунок 35.-Работа 1

## ***Класс для работы 2***

#pragma once

#include <iostream>

#include <math.h>

using namespace std;

class Zadanie2 {

public:

double f(double x, int tasknum) {

if (tasknum == 1) return x \* x + 4\*sin(x);

if (tasknum == 2) return pow(x, 3) - 3 \* pow(x, 2) + 12 \* x - 9;

}

double dif1(double x, int tasknum) {

if (tasknum == 1) return 2 \* x +4 \* cos(10 \* x);

if (tasknum == 2) return 3 \* pow(x, 2) + 6 \* x + 6;

}

double dif2(double x, int tasknum) {

if (tasknum == 1) return 2- 4 \* sin(x);

if (tasknum == 2) return 6 \* x + 6;

}

void precise(double x, double e, int tasknum) {

double h = 0.001;

double a = x - h;

double b = x + h;

double c;

while (fabs(b) > e \* 1000) {

if (f(a, tasknum) \* dif2(a, tasknum) > 0) {

b = a - (f(a, tasknum) / (f(b, tasknum) - f(a, tasknum))) \* (b - a);

}

if (f(a, tasknum) \* dif2(a, tasknum) < 0) {

a = a - (f(a, tasknum) / (f(b, tasknum) - f(a, tasknum))) \* (b - a);

}

}

cout << "уточнённый корень x[" << 0 << "] равен " << x << "\n";

}

void analitic(double\* y, double\* x, double a, double b, double h, double e, bool key, int tasknum) {

int j = 0;

for (double i = a; i < b + h; i = i + h) {

//cout << "i = " << i << ", f(i) = " << f(i) << "\n";

if (fabs(f(i, tasknum)) < e) {

if (j > 0) {

if (fabs(x[j - 1] - i) < 1) continue;

}

x[j] = i;

y[j] = f(i, tasknum);

cout << "x[" << j << "] находится на промежутке: [" << x[j] - h << "; " << x[j] + h << "]\n";

j++;

}

}

if (key == 1) precise(x[0], 0.001, tasknum);

//cout << "проверка = " << f(x[0], tasknum) << "\n";

}

void graphic(double\* y, double\* x, double a, double b, double h, double e, bool key, int tasknum) {

int j = 0;

for (double i = a; i < b + h; i = i + h) {

//cout << "i = " << i << ", f(i) = " << f(i) << "\n";

if (fabs(f(i, tasknum)) < e) {

if (j > 0) {

if (fabs(x[j - 1] - i) < 1) continue;

}

x[j] = i;

y[j] = f(i, tasknum);

cout << "x[" << j << "] находится на промежутке: [" << x[j] - h << "; " << x[j] + h << "]\n";

j++;

}

}

if (key == 1) precise(x[0], 0.001, tasknum);

//cout << "проверка = " << f(x[0], tasknum) << "\n";

}

void run() {

// setlocale(LC\_ALL, "rus");

// объявление и инициализация поинтера на функцию

double a = -10, b = 10, h = 0.001;

const int n = 10;

const double e = 0.01;

double\* y = new double[n];

double\* x = new double[n];

//graphic(y, x, a, b, h, e, 1, 3);

cout << "Введите номер задания(1 или 2): \n";

int input;

cin >> input;

while (input == 1 || input == 2) {

if (input == 1) {

analitic(y, x, a, b, h, e, 1, 1);

}

if (input == 2) {

graphic(y, x, a, b, h, e, 1, 2);

}

cout << "Введите не номер задания для выхода \n";

cin >> input;

}

cout << "выход\n";

}

};

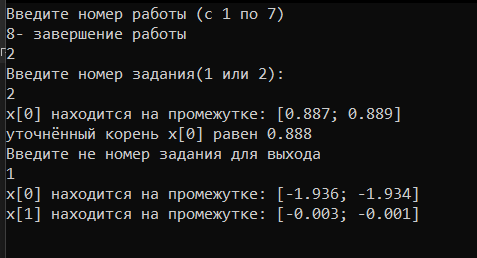


Рисунок 36.-Работа 2

## ***Класс для работы 3***

#pragma once

#include <iostream>

#include <math.h>

using namespace std;

class Zadanie3 {

public:

double f(double x, int tasknum) {

if (tasknum == 1) return x \* x + 4 \* sin(x);

if (tasknum == 2) return pow(x, 3) - 3 \* pow(x, 2) + 12 \* x - 9;

}

double dif1(double x, int tasknum) {

if (tasknum == 1) return 2 \* x + 4 \* cos(10 \* x);

if (tasknum == 2) return 3 \* pow(x, 2) + 6 \* x + 6;

}

double dif2(double x, int tasknum) {

if (tasknum == 1) return 2 - 4 \* sin(x);

if (tasknum == 2) return 6 \* x + 6;

}

void precise(double x, double e, int tasknum) {

double h = 0.0001;

double a = x - h;

double b = x + h;

double c1 = 11, c0, c;

if (f(a, tasknum) \* dif2(a, tasknum) > 0) c0 = a;

else c0 = b;

c = c0;

while (fabs(c1) < e) {

c1 = c0 - (f(c0, tasknum) / dif1(c, tasknum));

c0 = c1;

}

cout << "уточнённый корень x[" << 0 << "] равен " << x + 3 \* h << "\n";

}

void analitic(double\* y, double\* x, double a, double b, double h, double e, bool key, int tasknum) {

int j = 0;

for (double i = a; i < b + h; i = i + h) {

//cout << "i = " << i << ", f(i) = " << f(i) << "\n";

if (fabs(f(i, tasknum)) < e) {

if (j > 0) {

if (fabs(x[j - 1] - i) < 1) continue;

}

x[j] = i;

y[j] = f(i, tasknum);

cout << "x[" << j << "] находится на промежутке: [" << x[j] - h << "; " << x[j] + h << "]\n";

j++;

}

}

if (key == 1) precise(x[0], 0.001, tasknum);

//cout << "проверка = " << f(x[0], tasknum) << "\n";

}

void graphic(double\* y, double\* x, double a, double b, double h, double e, bool key, int tasknum) {

int j = 0;

for (double i = a; i < b + h; i = i + h) {

//cout << "i = " << i << ", f(i) = " << f(i) << "\n";

if (fabs(f(i, tasknum)) < e) {

if (j > 0) {

if (fabs(x[j - 1] - i) < 1) continue;

}

x[j] = i;

y[j] = f(i, tasknum);

cout << "x[" << j << "] находится на промежутке: [" << x[j] - h << "; " << x[j] + h << "]\n";

j++;

}

}

if (key == 1) precise(x[0], 0.001, tasknum);

//cout << "проверка = " << f(x[0], tasknum) << "\n";

}

void run() {

// setlocale(LC\_ALL, "rus");

// объявление и инициализация поинтера на функцию

double a = -10, b = 10, h = 0.001;

const int n = 10;

const double e = 0.01;

double\* y = new double[n];

double\* x = new double[n];

//graphic(y, x, a, b, h, e, 1, 3);

cout << "введите номер задания(1/2): \n";

int input;

cin >> input;

while (input == 1 || input == 2) {

if (input == 1) {

analitic(y, x, a, b, h, e, 1, 1);

}

if (input == 2) {

graphic(y, x, a, b, h, e, 1, 2);

}

cout << "введите не номер задания для выхода \n";

cin >> input;

}

cout << "выход\n";

}

};

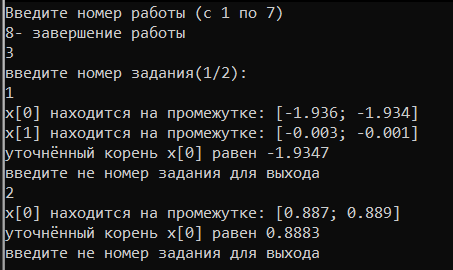


Рисунок 37.-Работа 3

## ***Класс для работы 4***

#pragma once

#include <iostream>

#include <math.h>

using namespace std;

class Zadanie4 {

public:

double f(double x, int tasknum) {

if (tasknum == 1) return pow(x, 3) +3 \* pow(x, 2) - 3.5;

}

void precise(double x, double e, int tasknum) {

double h = 0.001;

double a = x - h;

double b = x + h;

double c;

while (fabs(b - a) > e) {

c = (a + b) / 2;

if (f(a, tasknum) \* f(c, tasknum) > 0) a = c;

else b = c;

}

cout << "уточнённый корень x[" << 0 << "] равен " << (a + b) / 2 << "\n";

}

void analitic(double\* y, double\* x, double a, double b, double h, double e, bool key, int tasknum) {

int j = 0;

for (double i = a; i < b + h; i = i + h) {

//cout << "i = " << i << ", f(i) = " << f(i) << "\n";

if (fabs(f(i, tasknum)) < e) {

if (j > 0) {

if (fabs(x[j - 1] - i) < 1) continue;

}

x[j] = i;

y[j] = f(i, tasknum);

cout << "x[" << j << "] находится на промежутке: [" << x[j] - h << "; " << x[j] + h << "]\n";

j++;

}

}

if (key == 1) precise(x[0], 0.001, tasknum);

//cout << "проверка = " << f(x[0]) << "\n";

}

void run() {

// setlocale(LC\_ALL, "rus");

// объявление и инициализация поинтера на функцию

double a = -5, b = 5, h = 0.001;

const int n = 10;

const double e = 0.01;

double\* y = new double[n];

double\* x = new double[n];

analitic(y, x, a, b, h, e, 1, 1);

}

};

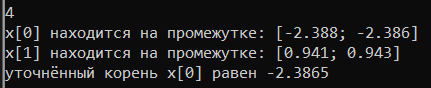


Рисунок 38.-Работа 4

## ***Класс для работы 5***

#include <iostream>

#include <math.h>

using namespace std;

class Zadanie5 {

public:

double f(double x, int tasknum) {

if (tasknum == 1) return 2.2\*x -pow(2,x);

if (tasknum == 2) return pow(x, 3) -0.2 \* x \* x + 0.5\*x -1;

}

void precise(double x, double e, int tasknum) {

double h = 0.001;

double a = x - h;

double b = x + h;

double c;

while (fabs(b - a) > e) {

c = (a + b) / 2;

if (f(a, tasknum) \* f(c, tasknum) > 0) a = c;

else b = c;

}

cout << "уточнённый корень x[" << 0 << "] равен " << (a + b) / 2 << "\n";

}

void analitic(double\* y, double\* x, double a, double b, double h, double e, bool key, int tasknum) {

int j = 0;

for (double i = a; i < b + h; i = i + h) {

//cout << "i = " << i << ", f(i) = " << f(i) << "\n";

if (fabs(f(i, tasknum)) < e) {

if (j > 0) {

if (fabs(x[j - 1] - i) < 1) continue;

}

x[j] = i;

y[j] = f(i, tasknum);

cout << "x[" << j << "] находится на промежутке: [" << x[j] - h << "; " << x[j] + h << "]\n";

j++;

}

}

if (key == 1) precise(x[0], e, tasknum);

//cout << "проверка = " << f(x[0]) << "\n";

}

void graphic(double\* y, double\* x, double a, double b, double h, double e, bool key, int tasknum) {

int j = 0;

for (double i = a; i < b + h; i = i + h) {

//cout << "i = " << i << ", f(i) = " << f(i) << "\n";

if (fabs(f(i, tasknum)) < e) {

if (j > 0) {

if (fabs(x[j - 1] - i) < 1) continue;

}

x[j] = i;

y[j] = f(i, tasknum);

cout << "x[" << j << "] находится на промежутке: [" << x[j] - h << "; " << x[j] + h << "]\n";

j++;

}

}

if (key == 1) precise(x[0], e, tasknum);

//cout << "проверка = " << f(x[0]) << "\n";

}

void run() {

// setlocale(LC\_ALL, "rus");

// объявление и инициализация поинтера на функцию

double a = -5, b = 5, h = 0.001;

const int n = 10;

const double e = 0.01;

double\* y = new double[n];

double\* x = new double[n];

//graphic(y, x, a, b, h, e, 1, 3);

cout << "Введите номер задания(1 или 2): \n";

int input;

cin >> input;

while (input == 1 || input == 2) {

if (input == 1) {

graphic(y, x, a, b, h, e, 1, 1);

}

if (input == 2) {

analitic(y, x, a, b, h, e, 1, 2);

}

cout << "Введите не номер задания для выхода \n";

cin >> input;

}

cout << "выход\n";

}

};

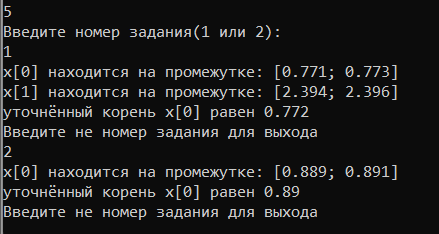


Рисунок 39.-Работа 5

## ***Класс для работы 6***

#pragma once

#include <iostream>

#include <math.h>

#include <algorithm>

using namespace std;

class Zadanie6 {

public:

//функция общая

double func(double x, int num, int tasknum) {

double f[2];

if (tasknum == 1) {

f[0] = 0.5-cos(x-2); // x =

f[1] = sin(x+2)-1.5; // y =

}

else {

f[0] = sqrt((1 - 1.5 \* x \* x) / 0.8); // x =

f[1] = 2.0 \* x + asin(1.2 \* x + 0.4) - 3.14; // y =

}

/\*

f[0] = sqrt((1 - 1.5 \* x \* x) / 0.8); // x =

f[1] = 2.0 \* x + asin(1.2 \* x + 0.4) - 3.14;

//f[1] = 2.0 \* x - asin(1.2 \* x + 0.4); // y =

f[0] = (1.0 / 3.0) \* cos(x) + 0.3; // x =

f[1] = sin(x - 0.6) - 1.6; // y =

\*/

return f[num];

}

// борды для первого задания

void borders(double\* bonds, double a, double b, double h, int tasknum) {

double minix = b, maxix = a, miniy = b, maxiy = a;

for (double i = a; i < b; i = i + h) {

//cout << "func0 = " << func(i, 0, tasknum) << ", func1 = " << func(i, 1, tasknum) << ", i = " << i << "\n";

minix = min(func(i, 0, tasknum), minix);

maxix = max(func(i, 0, tasknum), maxix);

miniy = min(func(i, 1, tasknum), miniy);

maxiy = max(func(i, 1, tasknum), maxiy);

}

bonds[0] = minix;

bonds[1] = maxix;

bonds[2] = miniy;

bonds[3] = maxiy;

}

void graphic(double\* bonds, double h, double e, int tasknum) {

//double x[3];

int j = 0;

for (double i = bonds[0]; i < bonds[1]; i = i + h) {

for (double j = bonds[2]; j < bonds[3]; j = j + h) {

if (fabs(i - func(j, 0, tasknum)) < e && fabs(j - func(i, 1, tasknum)) < e) {

//x[j] =

cout << "уточнённый корень равен:\n x = " << func(j, 0, tasknum) << "\n y = " << func(i, 1, tasknum) << "\n";

}

}

}

}

//юзлесс

void func2(double x, double y, int num) {

double f[4];

f[0] = sin(2 \* x - y) - 1.2 \* x - 0.4;

f[1] = 0.8 \* x \* x + 1.5 \* y \* y - 1;

}

// борды для второго задания

void borders2(double\* bonds, double a, double b, double h, int tasknum) {

double minix = 100, maxix = -100, miniy = 100, maxiy = -100; //f[0] x =

double minixx = 100, maxixx = -100, miniyy = 100, maxiyy = -100; //f[1] y =

for (double i = a; i < b; i = i + h) {

if (func(i, 0, tasknum) != func(i, 0, tasknum)) continue;

//cout << "func0 = " << func(i, 0, tasknum) << ", i = " << i << "\n";

minix = min(min(func(i, 0, tasknum), minix), min(-func(i, 0, tasknum), minix));

maxix = max(func(i, 0, tasknum), maxix);

miniy = min(i, miniy);

maxiy = max(i, maxiy);

//cout << "minx = " << minix << ", maxx = " << maxix << ", miny = " << miniy << ", maxy = " << maxiy << "\n";

}

for (double i = a; i < b; i = i + h) {

if (func(i, 1, tasknum) != func(i, 1, tasknum)) continue;

//cout << "func1 = " << func(i, 1, tasknum) << ", i = " << i << "\n";

minixx = min(i, minixx);

maxixx = max(i, maxixx);

miniyy = min(func(i, 1, tasknum), miniyy);

maxiyy = max(func(i, 1, tasknum), maxiyy);

//cout << "minxx = " << minixx << ", maxxx = " << maxixx << ", minyy = " << miniyy << ", maxyy = " << maxiyy << "\n";

}

bonds[0] = max(minix, minixx);

bonds[1] = min(maxix, maxixx);

bonds[2] = max(miniy, miniyy);

bonds[3] = min(maxiy, maxiyy);

/\*

for (int i = 0; i < 4; i++) {

cout << "bonds[" << i << "] = " << bonds[i] << "\n";

}

\*/

}

// эквивалент main(args)

void run() {

double\* f = new double[2];

double\* bonds = new double[4];

double h = 0.001;

const int n = 10;

double e = 0.001;

//borders(bonds, -10, 11, h, 1);

//borders2(bonds, -1, 1, h, 2);

//graphic(bonds, h, e, 2);

cout << "Введите номер задания(1 или 2): \n";

int input;

cin >> input;

while (input == 1 || input == 2) {

if (input == 1) {

borders(bonds, -10, 11, h, 1);

graphic(bonds, h, e = 0.001, 1);

}

if (input == 2) {

borders2(bonds, -1, 1, h, 2);

graphic(bonds, h, e = 0.002, 2);

}

cout << "введите не номер задания для выхода \n";

cin >> input;

}

cout << "выход\n";

}

};

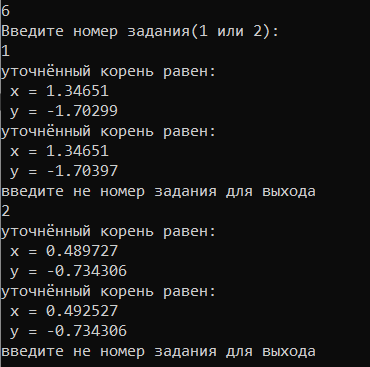


Рисунок 40.-Работа 6

## ***Класс для работы 7***

#pragma once

#include <iostream>

#include <math.h>

#include <algorithm>

#include <string>

using namespace std;

class Zadanie7 {

public:

void func(int\* f) {

f[0] = 0;

f[1] = 1;

f[2] = -31; //Вариант 10

f[3] = 0;

f[4] = -26;

}

int border(int\* f, int a, int b) {

for (int x = a; x < b + 1; x++) {

//cout << "x = " << x << ", f = " << f[1] \* pow(x, 3) + f[2] \* pow(x, 2) + f[3] \* x + f[4] << "\n";

//cout << "x = " << x << ", " << f[1] << " \* pow(x, 3) = " << f[1] \* pow(x, 3) << ", " << f[2] << " \* pow(x, 2) = " << f[2] \* pow(x, 2) << ", " << f[3] << " \* x = " << f[3] \* x << ", f[4] = " << f[4] << "\n";

//cout << "x = " << x << ", sign= " << (f[1] \* pow(x, 3) + f[2] \* pow(x, 2) + f[3] \* x + f[4]) \* (f[1] \* pow(x + 1, 3) + f[2] \* pow(x + 1, 2) + f[3] \* (x + 1) + f[4]) << "\n";

if ((f[1] \* pow(x, 3) + f[2] \* pow(x, 2) + f[3] \* x + f[4]) \* (f[1] \* pow(x + 1, 3) + f[2] \* pow(x + 1, 2) + f[3] \* (x + 1) + f[4]) < 0) {

//cout << x << "\n";

return x;

}

}

}

int sign(int\* f) {

int res = 1;

f[2] = -f[2];

f[4] = -f[4];

if (border(f, -35, 35) < 0) res = -1;

f[2] = -f[2];

f[4] = -f[4];

return res;

}

void GornerFreeman(int\* f) {

int bord = border(f, 0, 32);

int nums[3];

int signa = sign(f);

//cout << "signa = " << signa << "\n";

//f[0] = bord;

for (int i = 0; i < 3; i++) {

f[0] = bord;

int comparable = f[1] \* pow(bord, 3) + f[2] \* pow(bord, 2) + f[3] \* bord + f[4];

do {

f[2] = (f[1] \* bord) + f[2];

f[3] = (f[2] \* bord) + f[3];

f[4] = (f[3] \* bord) + f[4];

//for (int j = 0; j < 5; j++) cout << "f[" << j << "] = " << f[j] << "\n";

//cout << "f(bord) = " << f[1] \* pow(bord, 3) + f[2] \* pow(bord, 2) + f[3] \* bord + f[4] << "\n";

} while (f[4] != comparable);

f[2] = (f[1] \* bord) + f[2];

f[3] = (f[2] \* bord) + f[3];

f[2] = (f[1] \* bord) + f[2];

for (int j = 2; j < 5; j++) f[j] \*= pow(10, j - 1);

//for (int j = 0; j < 5; j++) cout << "f[" << j << "] = " << f[j] << "\n";

nums[i] = bord;

bord = border(f, 0, 32);

}

nums[0] = nums[0] \* signa;

string str = to\_string(double(abs(f[4])) / double(abs(f[3])) \* 10000);

string str2;

for (int i = 0; i < 3; i++) { str2 = str2 + str[i]; }

//cout << "str = " << str << "\n";

cout << "x = " << nums[0] << ".";

for (int i = 1; i < 3; i++) cout << nums[i];

for (int i = 0; i < 3; i++) cout << str[i];

cout << "\n";

}

void run() {

int n = 5;

int\* f = new int[n];

func(f);

//sign(f);

//border(f, -10, 10);

GornerFreeman(f);

system("pause");

}

};

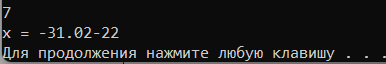


Рисунок 41.-Работа 7

# **Реализация в Java**

## ***Меню пользователя***

package Lab4;

import java.util.Scanner;

public class Menu {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

task1 zadanie1 = new task1();

task2 zadanie2 = new task2();

task3 zadanie3 = new task3();

task4 zadanie4 = new task4();

task5 zadanie5 = new task5();

task6 zadanie6 = new task6();

task7 zadanie7 = new task7();

System.out.println("Введите номер работы (с 1 по 7)");

System.out.println("8 - завершение работы");

int choice;

int exit = 0;

do {

choice = scanner.nextInt();

switch (choice) {

case 1:

zadanie1.run();

break;

case 2:

zadanie2.run();

break;

case 3:

zadanie3.run();

break;

case 4:

zadanie4.run();

break;

case 5:

zadanie5.run();

break;

case 6:

zadanie6.run();

break;

case 7:

zadanie7.run();

break;

case 8:

System.out.println("Завершение работы.\n\n");

System.out.println("Обратная связь:\n");

System.out.println("Студент гр. 21-САИ, Краличев Игорь Евгеньевич, ikralichev@list.ru");

exit = 1;

break;

}

} while (exit == 0);

scanner.close();

}

}

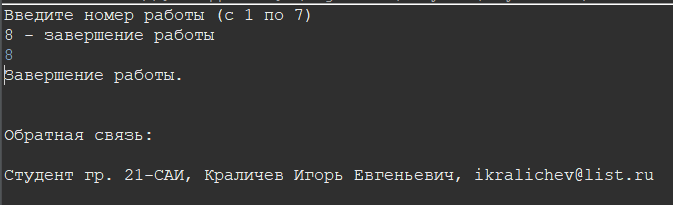


Рисунок 42.-Меню выбора задания

## ***Класс для работы 1***

package Lab4;

import java.util.Scanner;

public class task1 {

public double f(double x, int tasknum) {

if (tasknum == 1) return 2 \* Math.atan(x) - x + 3;

if (tasknum == 2) return 3 \* Math.pow(x, 4) - 8 \* Math.pow(x, 3) - 18 \* Math.pow(x, 2) + 2;

if (tasknum == 3) return 2 \* Math.sin(x + Math.PI / 3.0) - 0.5 \* x \* x + 1;

if (tasknum == 4) return 2 \* Math.log10(x) - x / 2 + 1;

return 0; // default return value

}

public void precise(double x, double e, int tasknum) {

double h = 0.001;

double a = x - h;

double b = x + h;

double c;

while (Math.abs(b - a) > e) {

c = (a + b) / 2;

if (f(a, tasknum) \* f(c, tasknum) > 0) a = c;

else b = c;

}

System.out.println("уточнённый корень x[" + 0 + "] равен " + (a + b) / 2);

}

public void analitic(double[] y, double[] x, double a, double b, double h, double e, boolean key, int tasknum) {

int j = 0;

for (double i = a; i < b + h; i = i + h) {

if (Math.abs(f(i, tasknum)) < e) {

if (j > 0 && Math.abs(x[j - 1] - i) < 1) {

continue;

}

x[j] = i;

y[j] = f(i, tasknum);

System.out.println("x[" + j + "] находится на промежутке: [" + (x[j] - h) + "; " + (x[j] + h) + "]");

j++;

}

}

if (key) {

precise(x[0], e, tasknum);

}

}

public void graphic(double[] y, double[] x, double a, double b, double h, double e, boolean key, int tasknum) {

int j = 0;

for (double i = a; i < b + h; i = i + h) {

if (Math.abs(f(i, tasknum)) < e) {

if (j > 0 && Math.abs(x[j - 1] - i) < 1) {

continue;

}

x[j] = i;

y[j] = f(i, tasknum);

System.out.println("x[" + j + "] находится на промежутке: [" + (x[j] - h) + "; " + (x[j] + h) + "]");

j++;

}

}

if (key) {

precise(x[0], e, tasknum);

}

}

public void run() {

Scanner scanner = new Scanner(System.in, "Cp1251");

double a = -5, b = 5, h = 0.001;

final int n = 10;

final double e = 0.01;

double[] y = new double[n];

double[] x = new double[n];

System.out.println("введите номер задания(1/2/3/4): ");

int input = scanner.nextInt();

while (input == 1 || input == 2 || input == 3 || input == 4) {

if (input == 1) {

analitic(y, x, a, b, h, e, false, 1);

} else if (input == 2) {

analitic(y, x, a, b, h, e, true, 2);

} else if (input == 3) {

graphic(y, x, a, b, h, e, false, 3);

} else if (input == 4) {

graphic(y, x, a, b, h, e, true, 4);

}

System.out.println("введите не номер задания для выхода");

input = scanner.nextInt();

}

System.out.println("выход");

}

}

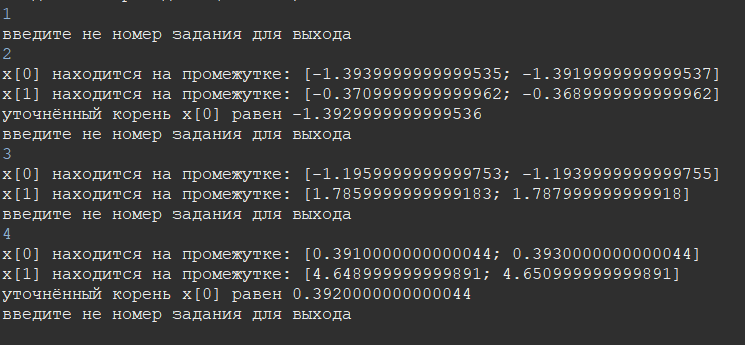


Рисунок 43.-Работа 1

## ***Класс для работы 2***

package Lab4;

import java.util.Scanner;

public class task2 {

public double f(double x, int tasknum) {

if (tasknum == 1) return x \* x + 4 \* Math.sin(x);

if (tasknum == 2) return Math.pow(x, 3) - 3 \* Math.pow(x, 2) + 12 \* x - 9;

return 0;

}

public double dif1(double x, int tasknum) {

if (tasknum == 1) return 2 \* x + 4 \* Math.cos(10 \* x);

if (tasknum == 2) return 3 \* Math.pow(x, 2) + 6 \* x + 6;

return 0;

}

public double dif2(double x, int tasknum) {

if (tasknum == 1) return 2 - 4 \* Math.sin(x);

if (tasknum == 2) return 6 \* x + 6;

return 0;

}

public void precise(double x, double e, int tasknum) {

double h = 0.001;

double a = x - h;

double b = x + h;

double c;

while (Math.abs(b) > e \* 1000) {

if (f(a, tasknum) \* dif2(a, tasknum) > 0) {

b = a - (f(a, tasknum) / (f(b, tasknum) - f(a, tasknum))) \* (b - a);

}

if (f(a, tasknum) \* dif2(a, tasknum) < 0) {

a = a - (f(a, tasknum) / (f(b, tasknum) - f(a, tasknum))) \* (b - a);

}

}

System.out.println("уточнённый корень x[" + 0 + "] равен " + x);

}

public void analitic(double[] y, double[] x, double a, double b, double h, double e, boolean key, int tasknum) {

int j = 0;

for (double i = a; i < b + h; i = i + h) {

if (Math.abs(f(i, tasknum)) < e) {

if (j > 0 && Math.abs(x[j - 1] - i) < 1) {

continue;

}

x[j] = i;

y[j] = f(i, tasknum);

System.out.println("x[" + j + "] находится на промежутке: [" + (x[j] - h) + "; " + (x[j] + h) + "]");

j++;

}

}

if (key) {

precise(x[0], 0.001, tasknum);

}

}

public void graphic(double[] y, double[] x, double a, double b, double h, double e, boolean key, int tasknum) {

int j = 0;

for (double i = a; i < b + h; i = i + h) {

if (Math.abs(f(i, tasknum)) < e) {

if (j > 0 && Math.abs(x[j - 1] - i) < 1) {

continue;

}

x[j] = i;

y[j] = f(i, tasknum);

System.out.println("x[" + j + "] находится на промежутке: [" + (x[j] - h) + "; " + (x[j] + h) + "]");

j++;

}

}

if (key) {

precise(x[0], 0.001, tasknum);

}

}

public void run() {

Scanner scanner = new Scanner(System.in, "Cp1251");

double a = -10, b = 10, h = 0.001;

final int n = 10;

final double e = 0.01;

double[] y = new double[n];

double[] x = new double[n];

System.out.println("введите номер задания(1/2): ");

int input = scanner.nextInt();

while (input == 1 || input == 2) {

if (input == 1) {

analitic(y, x, a, b, h, e, true, 1);

} else if (input == 2) {

graphic(y, x, a, b, h, e, true, 2);

}

System.out.println("введите не номер задания для выхода");

input = scanner.nextInt();

}

System.out.println("выход");

}

}

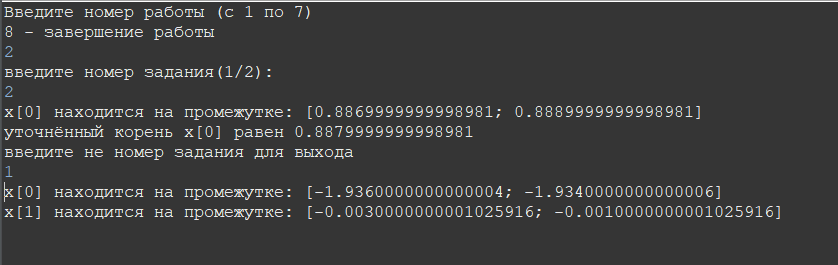


Рисунок 44.-Работа 2

## ***Класс для работы 3***

package Lab4;

import java.util.Scanner;

public class task3 {

public double f(double x, int tasknum) {

if (tasknum == 1) return x \* x + 4 \* Math.sin(x);

if (tasknum == 2) return Math.pow(x, 3) - 3 \* Math.pow(x, 2) + 12 \* x - 9;

return 0;

}

public double dif1(double x, int tasknum) {

if (tasknum == 1) return 2 \* x + 4 \* Math.cos(10 \* x);

if (tasknum == 2) return 3 \* Math.pow(x, 2) + 6 \* x + 6;

return 0;

}

public double dif2(double x, int tasknum) {

if (tasknum == 1) return 2 - 4 \* Math.sin(x);

if (tasknum == 2) return 6 \* x + 6;

return 0;

}

public double precise(double x, double e, int tasknum) {

double h = 0.0001;

double a = x - h;

double b = x + h;

double c1 = 11, c0, c;

if (f(a, tasknum) \* dif2(a, tasknum) > 0) c0 = a;

else c0 = b;

c = c0;

while (Math.abs(c1) < e) {

c1 = c0 - (f(c0, tasknum) / dif1(c, tasknum));

c0 = c1;

}

System.out.println("уточнённый корень x[" + 0 + "] равен " + (x + 3 \* h));

return (x + 3 \* h);

}

public void analitic(double[] y, double[] x, double a, double b, double h, double e, boolean key, int tasknum) {

int j = 0;

for (double i = a; i < b + h; i = i + h) {

if (Math.abs(f(i, tasknum)) < e) {

if (j > 0 && Math.abs(x[j - 1] - i) < 1) {

continue;

}

x[j] = i;

y[j] = f(i, tasknum);

System.out.println("x[" + j + "] находится на промежутке: [" + (x[j] - h) + "; " + (x[j] + h) + "]");

j++;

}

}

if (key) {

precise(x[0], 0.001, tasknum);

}

}

public void graphic(double[] y, double[] x, double a, double b, double h, double e, boolean key, int tasknum) {

int j = 0;

for (double i = a; i < b + h; i = i + h) {

if (Math.abs(f(i, tasknum)) < e) {

if (j > 0 && Math.abs(x[j - 1] - i) < 1) {

continue;

}

x[j] = i;

y[j] = f(i, tasknum);

System.out.println("x[" + j + "] находится на промежутке: [" + (x[j] - h) + "; " + (x[j] + h) + "]");

j++;

}

}

if (key) {

precise(x[0], 0.001, tasknum);

}

}

public void run() {

Scanner scanner = new Scanner(System.in, "Cp1251");

double a = -10, b = 10, h = 0.001;

final int n = 10;

final double e = 0.01;

double[] y = new double[n];

double[] x = new double[n];

System.out.println("введите номер задания(1/2): ");

int input = scanner.nextInt();

while (input == 1 || input == 2) {

if (input == 1) {

analitic(y, x, a, b, h, e, true, 1);

} else if (input == 2) {

graphic(y, x, a, b, h, e, true, 2);

}

System.out.println("введите не номер задания для выхода");

input = scanner.nextInt();

}

System.out.println("выход");

}

}

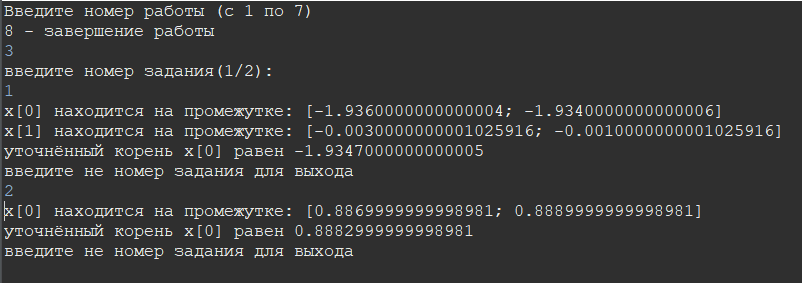


Рисунок 45.-Работа 3

## ***Класс для работы 4***

package Lab4;

import java.util.Scanner;

public class task4 {

public double f(double x, int tasknum) {

if (tasknum == 1) return Math.pow(x, 3) + 3 \* Math.pow(x, 2) - 3.5;

return 0; // default return value

}

public void precise(double x, double e, int tasknum) {

double h = 0.001;

double a = x - h;

double b = x + h;

double c;

while (Math.abs(b - a) > e) {

c = (a + b) / 2;

if (f(a, tasknum) \* f(c, tasknum) > 0) a = c;

else b = c;

}

System.out.println("уточнённый корень x[" + 0 + "] равен " + (a + b) / 2);

}

public void analitic(double[] y, double[] x, double a, double b, double h, double e, boolean key, int tasknum) {

int j = 0;

for (double i = a; i < b + h; i = i + h) {

if (Math.abs(f(i, tasknum)) < e) {

if (j > 0) {

if (Math.abs(x[j - 1] - i) < 1) continue;

}

x[j] = i;

y[j] = f(i, tasknum);

System.out.println("x[" + j + "] находится на промежутке: [" + (x[j] - h) + "; " + (x[j] + h) + "]");

j++;

}

}

if (key) {

precise(x[0], 0.001, tasknum);

}

}

public void run() {

Scanner scanner = new Scanner(System.in, "Cp1251");

double a = -5, b = 5, h = 0.001;

final int n = 10;

final double e = 0.01;

double[] y = new double[n];

double[] x = new double[n];

analitic(y, x, a, b, h, e, true, 1);

}

}

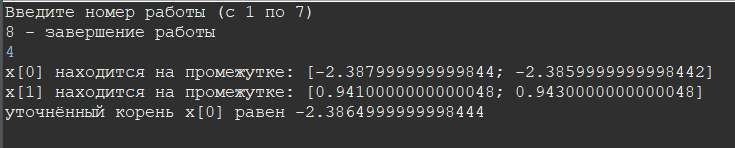


Рисунок 46.-Работа 4

## ***Класс для работы 5***

package Lab4;

import java.util.Scanner;

public class task5 {

public double f(double x, int tasknum) {

if (tasknum == 1) return 2.2 \* x - Math.pow(2, x);

if (tasknum == 2) return Math.pow(x, 3) - 0.2 \* x \* x + 0.5 \* x - 1;

return 0;

}

public void precise(double x, double e, int tasknum) {

double h = 0.001;

double a = x - h;

double b = x + h;

double c;

while (Math.abs(b - a) > e) {

c = (a + b) / 2;

if (f(a, tasknum) \* f(c, tasknum) > 0) a = c;

else b = c;

}

System.out.println("уточнённый корень x[" + 0 + "] равен " + (a + b) / 2);

}

public void analitic(double[] y, double[] x, double a, double b, double h, double e, boolean key, int tasknum) {

int j = 0;

for (double i = a; i < b + h; i = i + h) {

if (Math.abs(f(i, tasknum)) < e) {

if (j > 0 && Math.abs(x[j - 1] - i) < 1) {

continue;

}

x[j] = i;

y[j] = f(i, tasknum);

System.out.println("x[" + j + "] находится на промежутке: [" + (x[j] - h) + "; " + (x[j] + h) + "]");

j++;

}

}

if (key) {

precise(x[0], e, tasknum);

}

}

public void graphic(double[] y, double[] x, double a, double b, double h, double e, boolean key, int tasknum) {

int j = 0;

for (double i = a; i < b + h; i = i + h) {

if (Math.abs(f(i, tasknum)) < e) {

if (j > 0 && Math.abs(x[j - 1] - i) < 1) {

continue;

}

x[j] = i;

y[j] = f(i, tasknum);

System.out.println("x[" + j + "] находится на промежутке: [" + (x[j] - h) + "; " + (x[j] + h) + "]");

j++;

}

}

if (key) {

precise(x[0], e, tasknum);

}

}

public void run() {

Scanner scanner = new Scanner(System.in, "Cp1251");

double a = -5, b = 5, h = 0.001;

final int n = 10;

final double e = 0.01;

double[] y = new double[n];

double[] x = new double[n];

System.out.println("введите номер задания(1/2): ");

int input = scanner.nextInt();

while (input == 1 || input == 2) {

if (input == 1) {

graphic(y, x, a, b, h, e, true, 1);

} else if (input == 2) {

analitic(y, x, a, b, h, e, true, 2);

}

System.out.println("введите не номер задания для выхода");

input = scanner.nextInt();

}

System.out.println("выход");

}

}

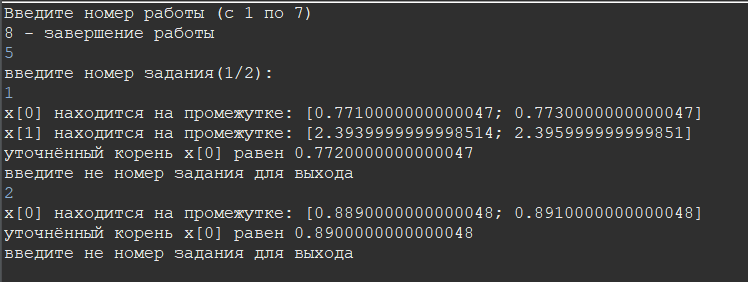


Рисунок 47.-Работа 5

## ***Класс для работы 6***

package Lab4;

import java.util.Scanner;

public class task6 {

public double func(double x, int num, int tasknum) {

double[] f = new double[2];

if (tasknum == 1) {

f[0] = 0.5 - Math.cos(x - 2); // x =

f[1] = Math.sin(x + 2) - 1.5; // y =

} else {

f[0] = Math.sqrt((1 - 1.5 \* x \* x) / 0.8); // x =

f[1] = 2.0 \* x + Math.asin(1.2 \* x + 0.4) - Math.PI; // y =

}

return f[num];

}

public void borders(double[] bonds, double a, double b, double h, int tasknum) {

double minix = b, maxix = a, miniy = b, maxiy = a;

for (double i = a; i < b; i += h) {

minix = Math.min(func(i, 0, tasknum), minix);

maxix = Math.max(func(i, 0, tasknum), maxix);

miniy = Math.min(func(i, 1, tasknum), miniy);

maxiy = Math.max(func(i, 1, tasknum), maxiy);

}

bonds[0] = minix;

bonds[1] = maxix;

bonds[2] = miniy;

bonds[3] = maxiy;

}

public void graphic(double[] bonds, double h, double e, int tasknum) {

for (double i = bonds[0]; i < bonds[1]; i += h) {

for (double j = bonds[2]; j < bonds[3]; j += h) {

if (Math.abs(i - func(j, 0, tasknum)) < e && Math.abs(j - func(i, 1, tasknum)) < e) {

System.out.println("уточнённый корень равен:\n x = " + func(j, 0, tasknum) + "\n y = " + func(i, 1, tasknum));

}

}

}

}

public void run() {

double[] bonds = new double[4];

double h = 0.001;

double e = 0.001;

Scanner scanner = new Scanner(System.in, "Cp1251");

System.out.println("введите номер задания(1/2): ");

int input = scanner.nextInt();

while (input == 1 || input == 2) {

if (input == 1) {

borders(bonds, -10, 11, h, 1);

graphic(bonds, h, e, 1);

} else if (input == 2) {

borders(bonds, -1, 1, h, 2);

graphic(bonds, h, e, 2);

}

System.out.println("введите не номер задания для выхода");

input = scanner.nextInt();

}

System.out.println("выход");

}

}

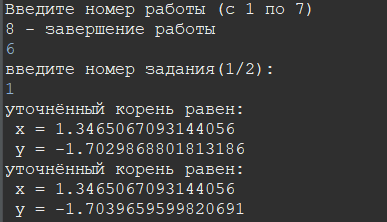


Рисунок 40.-Работа 6

## ***Класс для работы 7***

package Lab4;

public class task7 {

public void func(int[] f) {

f[0] = 0;

f[1] = 1;

f[2] = -31; // Вариант 10

f[3] = 0;

f[4] = -26;

}

public int border(int[] f, int a, int b) {

for (int x = a; x < b + 1; x++) {

if ((f[1] \* Math.pow(x, 3) + f[2] \* Math.pow(x, 2) + f[3] \* x + f[4]) \*

(f[1] \* Math.pow(x + 1, 3) + f[2] \* Math.pow(x + 1, 2) + f[3] \* (x + 1) + f[4]) < 0) {

return x;

}

}

return 0;

}

public int sign(int[] f) {

int res = 1;

f[2] = -f[2];

f[4] = -f[4];

if (border(f, -35, 35) < 0) {

res = -1;

}

f[2] = -f[2];

f[4] = -f[4];

return res;

}

public void gornerFreeman(int[] f) {

int bord = border(f, 0, 32);

int[] nums = new int[3];

int signa = sign(f);

for (int i = 0; i < 3; i++) {

f[0] = bord;

int comparable = f[1] \* (int) Math.pow(bord, 3) + f[2] \* (int) Math.pow(bord, 2) + f[3] \* bord + f[4];

do {

f[2] = (f[1] \* bord) + f[2];

f[3] = (f[2] \* bord) + f[3];

f[4] = (f[3] \* bord) + f[4];

} while (f[4] != comparable);

f[2] = (f[1] \* bord) + f[2];

f[3] = (f[2] \* bord) + f[3];

f[2] = (f[1] \* bord) + f[2];

for (int j = 2; j < 5; j++) {

f[j] \*= Math.pow(10, j - 1);

}

nums[i] = bord;

bord = border(f, 0, 32);

}

nums[0] = nums[0] \* signa;

String str = String.valueOf(doubleAbs(f[4]) / doubleAbs(f[3]) \* 10000);

String str2 = str.substring(0, 3);

System.out.print("x = " + nums[0]);

for (int i = 1; i < 3; i++) {

System.out.print(nums[i]);

}

System.out.print(str2 + "\n");

}

public double doubleAbs(double num) {

return Math.abs(num);

}

public void run() {

int n = 5;

int[] f = new int[n];

func(f);

gornerFreeman(f);

}

}

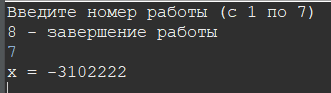


Рисунок 41.-Работа 7

**Список литературы**

1. **Численное решение задач экономики с использованием EXCEL, C++ и MATLAB [Электронные текстовые данные] : Учеб.пособие / Л.Ю. Катаева [и др.]; НГТУ им.Р.Е.Алексеева. - Н.Новгород : [Изд-во НГТУ], 2020. - 230 с. : ил. - Прил.:c.188-230. - Библиогр.:с.187.:**

[**https://fdp.nntu.ru/books/Chisl\_reshenie\_zadach\_economiki/Chisl\_reshenie\_zadach\_economiki/assets/basic-html/index.html#189в**](https://fdp.nntu.ru/books/Chisl_reshenie_zadach_economiki/Chisl_reshenie_zadach_economiki/assets/basic-html/index.html#189%D0%B2)**.**

**2. Численные методы : Курс лекций / В.А. Срочко. - СПб.; М.; Краснодар : Лань, 2010. - 202 с. - (Учебники для вузов. Специальная литература). - Библиогр.:с.200. - ISBN 978-5-8114-1014-9 : 180-00.**

[**https://studfile.net/preview/5793014/page:4/**](https://studfile.net/preview/5793014/page:4/)

**3. Численные методы линейной алгебры : Учеб.пособие / Г.С. Шевцов, О.Г. Крюкова, Б.И. Мызникова. - 2-е изд.,испр.и доп. - СПб.; М.; Краснодар : Лань, 2011. - 495 с. - (Учебники для вузов. Специальная литература). - Предм.указ.:с.491-495. - Библиогр.:с.489-490. - ISBN 978-5-8114-1246-4 : 465-00.:**

[**https://dpm.pstu.ru/images/R/Z/shevcov\_lineynaya\_algebra.pdf**](https://dpm.pstu.ru/images/R/Z/shevcov_lineynaya_algebra.pdf)