|  |  |  |
| --- | --- | --- |
|  |  |  |
| МИНОБРНАУКИ РОССИИ | | |
| Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | |
| Институт кибернетики | | |
| Кафедра программного обеспечения систем радиоэлектронной аппаратуры | | |

|  |  |
| --- | --- |
|  | |
| **ЛАБОРАТОРНАЯ РАБОТА № 1** | |
| Вариант № 12 | |
|  | |
| Студент группы: КМБО-02-19   Курса: 3 | *С. А. Минеев* |
| Руководитель практики \_\_\_\_\_\_\_\_\_\_\_\_  (Должность) | *А. А. Липатов* |

|  |  |  |
| --- | --- | --- |
| «Отчет представлен к рассмотрению» | «\_\_\_» \_\_\_\_\_\_\_\_\_\_2021 г. | *(подпись студента)* |
|  |  |  |
| «Отчет утвержден.  Допущен к защите» | «\_\_\_» \_\_\_\_\_\_\_\_\_\_2021 г. | *(подпись руководителя)* |

Москва 2021

# СОДЕРЖАНИЕ

[СОДЕРЖАНИЕ 2](#_Toc88925054)

[1 Текст задания 3](#_Toc88925055)

[1.1 Задание № 1 0](#_Toc88925056)

[1.2 Задание № 2 0](#_Toc88925057)

[1.3 Задание № 3 0](#_Toc88925058)

[2 Теоретическая часть 0](#_Toc88925059)

[2.1 Задание № 1 0](#_Toc88925060)

[2.2 Задание № 2 0](#_Toc88925061)

[2.3 Задание № 3 0](#_Toc88925062)

[3 Практическая часть 0](#_Toc88925063)

[3.1 Задание № 1 0](#_Toc88925064)

[3.2 Задание № 2 0](#_Toc88925065)

[3.3 Задание № 3 0](#_Toc88925066)

[4 Приложения 0](#_Toc88925067)

[4.1 Задание № 1 0](#_Toc88925068)

[4.2 Задание № 2 0](#_Toc88925069)

[4.3 Задание № 3 0](#_Toc88925070)

[Список использованных источников 0](#_Toc88925071)

[Как раскрыть все заголовки в документе: 0](#_Toc88925072)

# Текст задания

* 1. Задание № 1

1) Определить, какое равенство точнее.

2) Округлить сомнительные цифры числа, оставив верные знаки, и

определить абсолютную погрешность результата.

3) Найти предельные абсолютные и относительные погрешности чисел, если они имеют только верные цифры.

4) Вычислить и определить погрешность результата, если исходные

числа заданы в верных знаках. Записать результат в верных знаках.

|  |  |
| --- | --- |
| № | Данные |
| 1 |  |
| 2 |  |
| 3 | ; |
| 4 |  |

* 1. Задание № 2

1) Найти границы действительных корней, используя схему Горнера. Отделить корни и уточнить каждый из них методом итерации с точностью до 0.001.

2) Отделить корни уравнения графически и уточнить один из них

методом итерации с точностью до 0.001.

|  |  |
| --- | --- |
| № | Данные |
| 1 |  |
| 2 |  |

* 1. Задание № 3

Отделить корни уравнений и уточнить по одному из них с точностью до 0.001:

α) Методом хорд и методом касательных;

б) Комбинированным методом.

|  |  |
| --- | --- |
| п. | Данные |
| α) | *;* |
| б) |  |

# Теоретическая часть

* 1. Задание № 1

### 3.1.1] Пункт 1

#### Тема: Погрешность. Понятие погрешности, абсолютной и относительной погрешности.

В математике при вычислениях часто приходится встречаться с ситуациями, когда либо в результате вычислений получается некоторое точное значение *х*, либо приближенное значение . Во втором случае приближенное значение зависит от точного значения a по формуле:

В численных методах встречаются определенные виды погрешностей, которые рассмотрены далее.

Для оценки погрешности вводятся понятия ***абсолютной*** и

***Относительной*** погрешности. Пусть *х* — точное значение некоторой величины (нам оно неизвестно и никогда не будет известно, поскольку определяется с помощью измерений, страдающих неточностями); — приближенное значение той же величины ( ≈ *х*).

***Определение № 1:***

***Абсолютная погрешность* [АП]** приближенного числа определяется как:

; . (1)

Но поскольку *х* неизвестно, то и абсолютную погрешность мы узнать не можем! Чтобы разрешить парадокс, вводят ***предельную абсолютную погрешность***  .

***Определение № 2:***

***Предельную абсолютную погрешность* [ПАП]** — такое значение, которое абсолютная погрешность заведомо не превзойдет при данном способе измерений:

. (2)

Из выражения (1) следует, что, поэтому желательно возможно меньшее значение — это уменьшит длину интервала, содержащего искомое значение *х* и, следовательно, понизит неопределенность в наших знаниях об этой величине.

В технике формулу (1) часто записывают в виде , причем называется ***допусками***. Никакое изделие не может быть изготовлено с абсолютно точным соблюдением номинальных размеров, допуски показывают возможные (допустимые) отклонения от ***номинала***.

Итак, абсолютная погрешность оценивает точность измерений, но эта оценка неполна, поскольку не учитывает характерный размер изучаемого явления (объекта). Так, например, абсолютная погрешность в 1 см при измерении длины комнаты — вероятно, вполне приемлемая точность, но при измерении роста человека эта же погрешность будет сочтена непозволительно грубой.

Более информативным показателем качества измерений является относительная погрешность (соответственно предельная относительная погрешность ) приближенного числа *а* как отношение абсолютной погрешности (предельной абсолютной погрешности) к модулю числа *а.*

***Определение № 3:***

***Относительная погрешность* [ОП]** приближенного числа определяется как:

. (3)

***Определение № 4:***

***Предельную относительную погрешность* [ПОП]** — приближенного числа определяется как:

; . (4)

Относительная погрешность является величиной безразмерной, т. е. не зависит от выбора системы единиц измерения, что позволяет сравнивать качество измерений разнородных величин (бессмысленным является вопрос о том, что больше: 1 кг или 1 м, — но сравнение качества измерений массы и длины в терминах относительной погрешности вполне допустимо). Измеряется ( ) в долях единицы или в процентах.

**Пример.** Согласно ныне действующим (2015 г.) определениям международного Комитета по константам для науки и технологии входящая в закон всемирного тяготения гравитационная постоянная

= (6.67259 ± 0.00085)'10-11 м 3· кг–1· с–2,

а заряд электрона

*e* = (1.60217733± 0.00000049)'10-19 Кл.

Сравнить точность определения этих фундаментальных физических постоянных.

**Решение.** Для гравитационной постоянной предельная относительная погрешность

= 1.27 •;

= 3.1 •;

Таким образом, в последнем случае относительная погрешность оказывается на три порядка меньшей, т. е. заряд электрона определен существенно точнее, чем гравитационная постоянная.

### 3.1.2] Пункт 2

#### Тема: Понятия верных и значащих цифр числа.

С понятиями абсолютной и относительной погрешности связаны понятия верных и значащих цифр.

**Определение № 5:**

Если абсолютная погрешность приближенного числа не превышает единицы последнего (самого правого) разряда его десятичной записи, то цифры числа называют **верными** (или **точными**).

По умолчанию десятичная запись приближенного числа должна содержать только верные цифры, и тогда по записи числа сразу можно узнать предельную абсолютную погрешность, с которой оно известно.

Цифры, не являющиеся верными, называются **сомнительными**.

В общем случае значащие цифры числа определяются через неравенство:

.(3)

Где = заведомо известное число, *n – разряд определяемой цифры числа, абсолютная погрешность.*

**Определение № 6:**

***Значащими*** цифрами приближенного числа называются в с е цифры его десятичной записи, кроме нулей, находящихся левеe первой отличной от нуля цифры.

### 3.1.3 - 4] Пункт 3 + Пункт 4

#### Тема: Операции над приближенными значениями.

Теорема № 1:

Абсолютная погрешность алгебраической суммы нескольких приближенных чисел не превышает суммы абсолютных погрешностей этих чисел.

В частности, для суммы двух чисел *а* и *b* любого знака получаем:

;

Из этой теоремы следует, что абсолютная погрешность алгебраической суммы не меньше абсолютной погрешности наименее точного из слагаемых, т. е. увеличение точности за счет других слагаемых невозможно. Поэтому бессмысленно сохранять излишние десятичные знаки в более точных слагаемых. Отсюда вытекает следующее:

---------------------------------------------------------

Правилами подсчета верных знаках[ППВЗ]

1. При сложении (+) и вычитании (-) приближенных чисел младший сохраненный десятичный разряд результата должен являться старшим среди десятичных разрядов, выражаемых последними верными значащими цифрами исходных данных.

Пример:

⇒

1. При умножении (•) и делении () в результате следует сохранять столько значащих цифр, сколько их в приближенном, данном с наименьшим числом верных значащих цифр.

Пример:

⇒ у

⇒

⇒

⇒

1. При возведении числа в квадрат (x2) или куб (x3) в результате следует сохранять столько цифр, сколько верных значащих цифр имеет основание степени.

Пример:

⇒

----------------------------------------------------

Теорема № 2:

Относительная погрешность произведения (частного) приближенных чисел не превышает суммы относительных погрешностей этих чисел.

В частности, для трех чисел:

Из теоремы следует, что относительная погрешность произведения и частного не может быть меньше относительной погрешности наименее точного из исходных чисел (т. е. имеющего меньше всего верных значащих цифр). Поскольку относительная погрешность числа определяется количеством его верных значащих цифр, то при умножении и делении бессмысленно оставлять значащих цифр больше, чем их было в исходном числе с наименьшим количеством верных значащих цифр.

* 1. Задание № 2

### 3.2.1] Пункт 1 (QProject\_4)

#### Тема: Алгоритм отделения корней при помощи схемы Горнера. Метод итераций.

##### Алгоритм отделения корней при помощи схемы Горнера:

Пусть нам известен полином порядка *n*:

(1)

И для некоторого неизвестного корня определенны нижняя и верхняя границы т. е.:

< < , ;

Тогда:

• Для нахождения значения нижней границы требуется представить полином в виде:

(2)

И предполагая значения для корня(нижней границы) воспользоваться схемой Горнера.

Важно!: если *x* = – нижняя граница полинома, то в схеме Горнера:

(3)

, *i >0;*

Система нахождения коэффициентов схемы Горнера:

(4)

• Для нахождения значения верхней границы требуется представить полином в виде:

(5)

И воспользовавшись аналогичным алгоритмом, что описан ранее подобрать значения для границы *x* = .

##### Метод итераций(Дихотомии):

***Метод итерации*** — численный метод решения математических задач, используемый для приближённого решения алгебраических уравнений и систем. Суть метода заключается в нахождении по приближённому значению величины следующего приближения (являющегося более точным). Метод позволяет получить решение с заданной точностью в виде предела последовательности итераций. Характер сходимости и сам факт сходимости метода зависит от выбора начального приближения решения. Функциональное уравнение может быть записано в виде:

x=f(x)

Функцию ***f(x)*** называют ***сжимающим отображением***.  
  
Последовательность чисел ,  , … ,  называется ***итерационной***, если для любого номера ***n* > 0** элемент   выражается через элемент  по рекуррентной формуле:

xn=f(xn-1)

а в качестве  взято любое число из области задания функции ***f(x)*** *.*

### 3.2.2] Пункт 2 (QProject\_8)

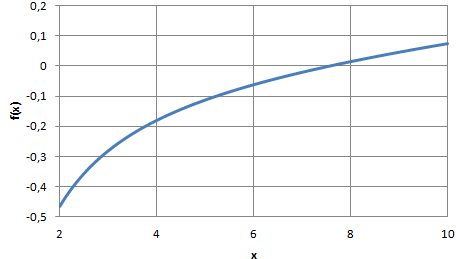
#### Тема: Отделение корней уравнения графически и уточнение методом итерации.

Отделение корней можно проводить графически и аналитически.  
Для того, чтобы графически отделить корни уравнения, необходимо построить график функции **f(x)**. Абсциссы точек его пересечения с осью Ox являются действительными корнями уравнения.  
  
Для примера рассмотрим задачу решения уравнения

Уравнение

где угол **x** задан в градусах. Указанное уравнение можно переписать в виде:

f(x)=0  
  
Для графического отсечения корней достаточно построить график функции:



Из рисунка видно, что корень уравнения лежит в промежутке

*x* ∈ (6;8).

Про метод итерации описано в предыдущем пункте.

* 1. Задание № 3

### 3.3.1] Пункт α (QProject\_5) / (QProject\_7)

#### 3.3.1.α.1] Подпункт 1 (QProject\_7)

##### Тема: Метод хорд.(Метод секущих)

Если ,   - приближенные значения корня уравнения  и выполняется условие:

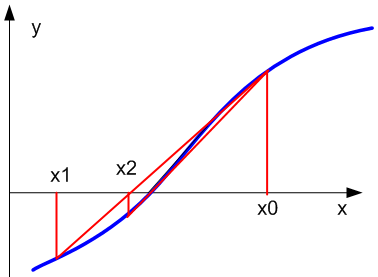
f(a)f(b)  
  
то последующие приближения находят по формуле:

Метод хорд

Методом хорд называют также метод, при котором один из концов отрезка закреплен, т.е. вычисление приближения корня уравнения  производят по формулам:

Метод секущих

Геометрическая интерпретация метода хорд:



#### 3.3.1.α.2] Подпункт 2 (QProject\_5)

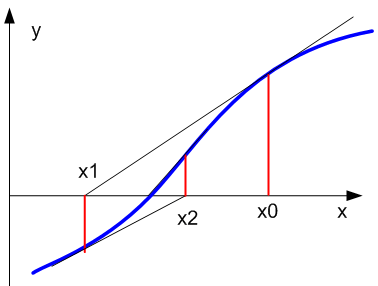
##### Тема: Метод касательных.(Метод Ньютона)

Если известно начальное приближение **x0** корня

уравнения , то последовательные приближения находят по формуле:

метод Ньютона  
  
Графическая интерпретация метода касательных имеет

вид:



# Практическая часть

* 1. Задание № 1

### 3.1.1] Пункт 1

#### Дано:

|  |  |
| --- | --- |
| № | Данные |
| a |  |
| b |  |

#### Определить:

Какое равенство точнее - ?

#### Решение:

##### Вычислим с точностью превышающую точность числа 2.63 на 3 разряда, т. е. с точностью .

###### 1.a) Вычислим значения .

Ручное вычисление:

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| \_ | 50 | | | | | 19 | | | |  |
| 38 | | | | | Ответ: 2.63157 | | | |  |
| \_ | 120 | | | | |  | | | | 19•2 = 38 |
| 114 | | | | | 19•3 = 57 |
| \_ | | 60 | | | | 19•6 = 114 |
| 57 | | | | 19•7 = 133 |
| \_ | | | 30 | | | |  | | | 19•3 = 57 |
| 19 | | | | 19\*4 = 76 |
| \_ | | | | 110 | | | |  | | 19•1 = 19 |
| 95 | | | | 19•2 = 38 |
| \_ | | | | | 150 | | |  | | 19•5 = 95 |
| 133 | | | 19•6 = 114 |
| Остаток: | | | | | | | | 17 |  | 19•7 = 133 |

###### 1.в) Вычислим значения

Ручное вычисление:

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | | | | | | Ответ: 5.19615 | | | |  |
|  | | 25 | | |  | | 1]5•5 =25 | | | |  | |
| \_ | | 2.00 | | | 2]10? • ? ={? = 1} = 101 | | | | 101•1 = 101 | | |
| 1.01 | | | 3]10? • ? ={? = 9} = 9’261 | | | | 102•2 = 204 | | |
| \_ | | | 99’00 | |  | | 4]10? • ? ={? = 6} = 62’316 | | | | 1029•9 = 9’261 | | |
| 92’61 | |  | | 5]10? • ? ={? = 1} = 103’921 | | | | 10210•10 = 102’100 | | |
| \_ | | | | 6’39’00 | | | 6]10? • ? ={? = 5} = 5’196’125 | | | | 10386•6 = 62’316 | | |
| 6’23’16 | | |  | | | | 10387•7 = 72’109 | | |
| \_ | | | | | 158’400 | |  | | | | 103’921•1 = 103’103921 | | |
| 103’921 | | 103’922•2 = 207’844 | | |
| \_ | | | | | | 5’447’900 | | |  | | 1’039’225•5 = 5’196’125 | | |
| 5’196’125 | | | 1’039’226•6 = 6’235’356 | | |
| Корневой Остаток: | | | | | | | | 25’177’500 | |  |  | | |
| Проверка на калькуляторе:  (Программное вычисление) | | | | | | | |  | |  |  | | |

##### Вычисление предельных абсолютных погрешностей, округляя их с избытком.

###### 2.a) Вычисление

###### 2.b) Вычисление

##### Вычисление предельных относительных погрешностей.

###### 3.a) Вычисление

###### 3.b) Вычисление

#### Ответ:

т. к. .

т. е.

### 3.1.2] Пункт 2

#### Дано:

|  |  |
| --- | --- |
| № | Данные |
| 2 |  |

#### Определить:

Сомнительные цифры числа, оставив верные знаки и определить абсолютную погрешность результата.

#### Решение:

##### Определим сомнительные цифры числа, используя следующие формулы:

###### 1.1) -.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | ***0*** | ***-1*** | ***-2*** | ***-3*** | ***-4*** | ***-5*** | ***: n*** |
|  | *0 , 8* | | *5* | *6* | *3* | *7* |  |

*где n – разряд определяемой цифры числа.*

###### 1.2) -.

*• Используем неравенство:*

|  |
| --- |
|  |

|  |  |  |
| --- | --- | --- |
| ***n*** |  | ***Статус*** |
| *0* |  | ***Верно*** |
| *-1* |  | ***Верно*** |
| *-2* |  | ***Верно*** |
| *-3* |  | ***Неверно*** |

*⇒ В числе верными являются только первые 3 цифры.*

*⇒ Определяем число*



*Три цифры*



*⇒ Все цифры в числе верны!*

#### Ответ:

### 3.1.3] Пункт 3

#### Дано:

|  |  |
| --- | --- |
| № | Данные |
| 2 |  |

#### Найти:

#### Решение:

##### Вычисления и

(По условию). В числе все цифры верны.

⇒

#### Ответ:

Для :

### 3.1.4] Пункт 4

#### Дано:

|  |  |
| --- | --- |
| № | Данные |
| 4 |  |

#### Условия:

• Вычислить и определить погрешность результата, если исходные числа заданы в верных знаках.

• Записать результат в верных знаках.

#### Решение:

##### Для вычисления значения переменной V в верных знаках воспользуемся правилами подсчета верных знаках[ППВЗ]:

---------------------------------------------------------

###### Правилами подсчета верных знаках[ППВЗ]

1. При сложении (+) и вычитании (-) приближенных чисел младший сохраненный десятичный разряд результата должен являться старшим среди десятичных разрядов, выражаемых последними верными значащими цифрами исходных данных.

Пример:

⇒

1. При умножении (•) и делении () в результате следует сохранять столько значащих цифр, сколько их в приближенном, данном с наименьшим числом верных значащих цифр.

Пример:

⇒ у

⇒

⇒

⇒

1. При возведении числа в квадрат (x2) или куб (x3) в результате следует сохранять столько цифр, сколько верных значащих цифр имеет основание степени.

Пример:

⇒

---------------------------------------------------------

• Возврат к задаче и применение ППВЗ для вычисления значения переменной V.

##### Для вычисления погрешностей(предельной абсолютной и предельной относительной) переменной V воспользуемся следующей таблицей “Погрешность результатов арифметических операций”.

###### **Таблица № 1 - Погрешность результатов арифметических операций.**

|  |  |
| --- | --- |
| **№** | **Формула** |
| 1 |  |
| 2 |  |
| 3 |  |
| 4 |  |
| 5 |  |
| 6 |  |
| 7 |  |
| 8 | = |
| 9 | = |

•

т. к.

•

•



*d*

• [*h•s•d*]:

*h =3.81; s = 45.8; d = 1.004;*

•

~

#### Ответ:

V = ;

* 1. Задание № 2

### 3.2.1] Пункт 1 (QProject\_4)

#### Дано:

|  |  |
| --- | --- |
| № | Данные |
| 1 |  |

#### Найти:

• Границы действительных корней, используя схему Горнера.

• Отделить корни и уточнить каждый из них методом

итерации с точностью до 0.001.

#### Решение:

##### Нахождение нижней (α) и верхней (β) границы некоторого

###### Для поиска нижней границы (α)

*⇒*

*⇒*

* + 1. Предположим,

*Имеем:*

*• Воспользуемся схемой Горнера для определения, является ли нижней* границей, т. е. должно выполняться условие в таб. *Горнера:*

*⇒*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  | *1* | *3* | *6* | *5* |  |
|  | *1* | *4* | *10* | *15* |  |
|  |  |  |  |  | *⇒* |

*⇒* α = 1;

###### Для поиска верхней границы (β)

•

*⇒*

* + 1. Предположим,

*Имеем:*

*• Воспользуемся схемой Горнера для определения, является ли верхней границей, т. е. должно выполняться условие в таб. Горнера:*

*⇒*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  | *1* | *-3* | *6* | *-5* |  |
|  | *1* | *0* | *6* | *13* |  |
|  |  |  |  |  | *⇒* |

*⇒* β = 3;

*• Проверка:*

*(Проверка на наличие одного )*

*• Проверим, что имеет только один вообще!*

###### Ответ № 1:

[α ; β] = [1 ; 3] - *границы какого-либо*

##### Воспользуемся методом итераций для уточнения на отрезке [1 ; 3] с точностью

###### Ручное вычисление:

• т. к.

[α ; β] = [1 ; 3] [α ;] = [1 ; 2];

*Т. к.*

*(Накопительная погрешность формируется в процессе возведения в степень n.)*

*• Определим с помощью следующей формулы число нужных итераций для достижения точности*

*При n*

*При n*

*требуется 6 итераций для вычисления с точностью*

|  |
| --- |
|  |

*Точность достигнута, если:*

*-* ***Верно!***

###### Ответ № 2:



**Рис. 1 – Листинг кода к заданию № 2.1.**

#### Ответ:

### 3.2.2] Пункт 2 (QProject\_8)

#### Дано:

|  |  |
| --- | --- |
| № | Данные |
| 2 |  |

#### Отделить:

• Корни уравнения графически и уточнить один из корней методом итерации с точностью до .

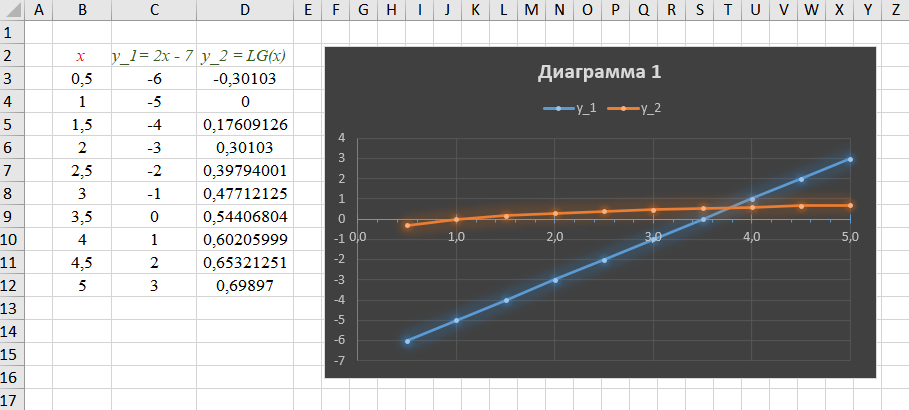
#### Решение:

*• Сведём : .*

*• Эквивалентно записи*

*• Для определения отрезка, в котором находится решение уравнения , нарисуем графики обеих функций.*

###### Графическое рисование:



*Рисунок № 1 – Диаграмма двух функций.*

• Из рисунка № 1 видно, что x

• Воспользуемся методом итераций для его уточнения с точностью .

*Т. к.*

*(Накопительная погрешность формируется в процессе возведения в степень n.)*

*• Определим с помощью следующей формулы число нужных итераций для достижения точности*

*При n*

*При n*

*требуется 3 итераций для вычисления с точностью*

|  |
| --- |
| *Видно, что после достижения 3 итерации в последующей итерации значение в 3 разряде не меняется.* |

*Точность достигнута, если:*

*-* ***Верно!***

#### Ответ:

* 1. Задание № 3

### 3.3.1] Пункт α (QProject\_5) / (QProject\_7)

#### 3.3.1.α.1] Подпункт 1 (QProject\_7)

##### Дано:

|  |  |
| --- | --- |
| № | Данные |
| α) | *;* |

##### Условие:

• Отделить корни уравнений и уточнить по одному из них с точностью до 0.001: методом хорд.

##### Решение:

###### Вычисление

* 1. Поиск отрезка [α; β]:

Ручное вычисление:

• Условие*:*

⇒ ;

*•*

⇒ т. к.

* + 1. Проверка того, что на отрезке имеется ровно 1 корень.

*Теорема № 1:*

*Если функция непрерывна и дифференциируема на отрезке* [α; β] и сохраняет знак производной внутри отрезка [α; β]

т. е. то на отрезке [α; β] имеется ровно 1 корень.

*• Исходя из теоремы № 1 имеем:*

*• Обозначим*

*• Для доказательства требуется найти корни и с помощью метода интервалов определить знаки в промежутке между корнями функции Если на одном из промежутков функция имеет отрицательное значение, то для функции теорема № 1 не выполняется.*

Теория

*п.1) При этом, стоит отметить, что уравнении позволяет определить критические(стационарные и особые) точки. Особенно точки стационарные(точки экстремума), которые меняют знак функции на противоположный.*

*⇒ Если – корень*

*п.2) В случае, если – корень на отрезке [a;b], то для функции*

*требуется сдвинуть отрезок в сторону корня с той стороны отрезка [a;b] на которой*

*п.3) В случае, если – корней на отрезке [a;b] для функции больше одного, то возможны два случая:*

1. *Если – корней на отрезке [a;b] для функции четное:*

*Пример:*

+

+

-

-

+

[

]

*a*

*b*

-

-

+

+

-

*v.1*

*v.2*

*⇒*

*• В этом случае придётся на отрезке [a;b] исследовать все интервалы положительного и для каждого их них проверять для функции существования корня, т. е.*

1. *Если – корней на отрезке [a;b] для функции нечетное:*

*Пример:*

+

+

-

-

+

[

]

*a*

*b*

-

-

+

+

-

*v.1*

*v.2*

*⇒*

*• В этом случае придётся поступать также, как и в случае 1).*

*⇒ Из выше написанного:*

*• Определим количество – корней для функции*

*• В общем:*

*Для доказательства требуется найти корень . И если корень один, то с помощью метода интервалов определить знаки слева и справа от корня.*

*⇒*

*// Определяем количество – корней для*

*•*

*⇒ ⇒ Имеется на [0:1] покрасней мере 1 корень.*

*• Проверка того, что на отрезке [a;b] функция имеет ровно один корень.*

*⇒ Согласно Теореме № 1 требуется проверить, что*

*• Обозначим: = ;*

*⇒*

*⇒*

*• Либо не имеет корней на отрезке [0;1].*

*• Либо:*

[0;s(0)]

[1;s(1)]

*т. е. имеется точка экстремума, т. е.*

*• Проверим на отрезке [; ].*

-

+

+

*⇒*

*⇒ на отрезке [0;1] не имеет корней.*

*⇒ = имеет на отрезке [0;1] один знак, т. е.*

*⇒ имеет ровно один корень на отрезке [0;1].*

*• Найдем этот корень с помощью метода хорд:*

*⇒ Имеем:*

*• Будем искать корень для с точностью*

*•⇒*

*• Формула итерационного процесса:*

*⇒*

*• Проверка;(Достигнута ли точность )*

*• Условия достижения точности:*

*⇒ Точность не достигнута.*

*Проверка;(Достигнута ли точность )*

*• Условия достижения точности:*

*⇒*

*Проверка;(Достигнута ли точность )*

*• Условия достижения точности:*

*⇒*

*⇒ - корень с точностью*

*• Воспользуемся методом интервалов для определения знака функции .*

*⇒*

[

+

*a=*

*b=*

]

-

*⇒ ⇒ на отрезке [0;1] может иметься больше одного корня или точка экстремума, которые находятся по формуле: и есть корень этого уравнения и*

*⇒*

*⇒ Согласно п.2 [теории] сдвигаем отрезок [0;1] в отрезок [0,2351;1].*

*⇒ [0;1] [0,2351;1];*

*• Для полученного отрезка [0,2351;1] проделываем расчеты, начиная с проверки условия существования на этом отрезке хотя бы одного корня.*

*⇒*

*⇒ ⇒ На отрезке [0,2351;1] имеется хотя бы один корень.*

*• Проверка того, что на отрезке [0,2351;1] имеется ровно один корень.*

*⇒ Согласно Теореме № 1 требуется проверить, что .*

*⇒ ⇒ на отрезке [0,2351;1] имеет ровно 1 корень.*

*• Найдем корень с помощью метода хорд с точностью до*

*⇒*

*⇒*

*• Формула итерационного процесса:*

*⇒*

*• Проверка;(Достигнута ли точность )*

*• Условия достижения точности:*

*⇒ Точность не достигнута.*

*• Проверка;(Достигнута ли точность )*

*• Условия достижения точности:*

*⇒ Точность не достигнута.*

*• Проверка;(Достигнута ли точность )*

*• Условия достижения точности:*

*⇒ Точность не достигнута.*

*•*

*• Проверка;(Достигнута ли точность )*

*• Условия достижения точности:*

*⇒ Точность* ***достигнута****.*

*Ответ:*

Программное вычисление:

*< Код описан в главе “Приложения” пункте 4.3.1 Задание № 3, подпункте 1. >*

##### Ответ:

#### 3.3.1.α.2] Подпункт 2 (QProject\_5)

##### Дано:

|  |  |
| --- | --- |
| № | Данные |
| a |  |

##### Определить:

• Отделить корни уравнений и уточнить по одному из них методом касательных с точностью до 0.001 - ?

##### Решение:

###### Поиск отрезка [α; β] для одного из корней:

• Условие наличие хотя-бы одного корня на отрезке [α; β]:

• Пусть

• Проверка наличие ровно одного коря на отрезке:

• Если на отрезке [0; 2] имеется более одного корня, то на этом отрезке имеется критическая(стационарная или особая) точка.

• Проверка наличия критической точки на отрезке [0; 2]:

критических точек нет*.* И по *теореме № 1* имеет ровно один корень.

•

• Проверка: (Достигнута ли точность 0.001)

• Условия точности:

*•*

*•*

*•*

##### Ответ:

### 3.3.2] Пункт б (QProject\_6)

#### Дано:

|  |  |
| --- | --- |
| № | Данные |
| б) |  |

#### Определить:

Отделить корни уравнений и уточнить по одному из них с точностью до 0.001: Комбинированным методом.

#### Решение:

##### Поиск отрезка [α; β]:

Таб. 1

|  |  |  |
| --- | --- | --- |
|  |  |  |
| -∞ | -∞ | ⊝ |
| -4 | -24 | ⊝ |
| -3 | +10 | ⊕ |
| -1 | +12 | ⊕ |
| 0 | -8 | ⊝ |
| 1 | -34 | ⊝ |
| 6 | -44 | ⊝ |
| 7 | 20 | ⊕ |

Имеем три корня с интервалами: (-4; -3), (-1;0), (6;7);

• Найдем корень с точностью до 0.001, воспользовавшись комбинированным методом.

• Из таб. 1 имеем:

##### Формулы:

• Для нахождения и воспользуемся формулами из методов *хорд*(*секущих*) и *касательных*(*Ньютона*) соответственно.

* + Корень вычисляется по формуле:

• Формула достижения точности:

##### Расчет:

• Проверка:

• Проверка:

• Корень :

#### Ответ:

# Приложение 1 – примеры работы программы.

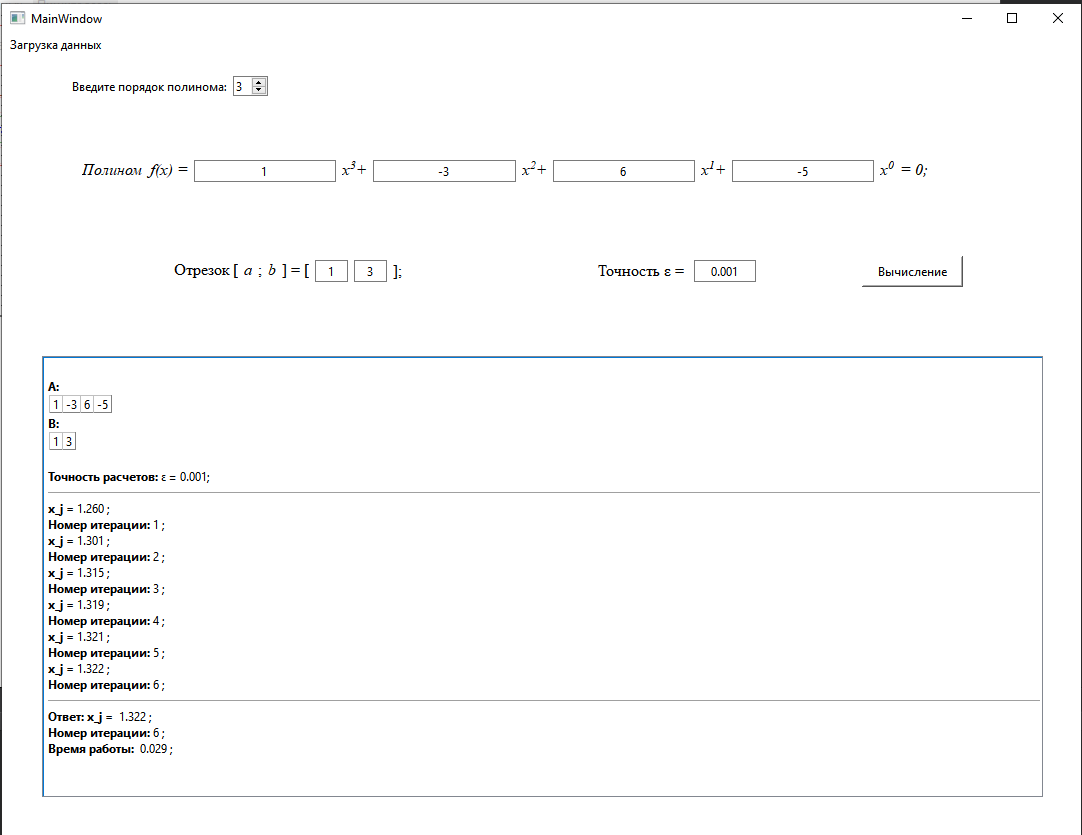
4.1 Задание № 1

- - -

4.2 Задание № 2

### Пункт № 1:

Ниже представлен пример работы программы с исходными данными и листинг основного кода, а также ссылка на GitHub.



*Рис.1 – Пример работы программы.*

### Пункт № 2:

Ниже представлен пример работы программы с исходными данными и листинг основного кода, а также ссылка на GitHub.



Рис. 2 – Пример работы программы.

4.3 Задание № 3

### Пункт a.

#### Подпункт 1 (QProject\_7)



Рис. 3 – Пример работы программы.

Код:

*auto* f = [&](double x) -> double{

double xi = 0;

size\_t ik = 0;

std::function<double(int index, int i)> C\_f\_fun;

*auto* C\_a\_fun = [&](int index, int i, double buff) -> double {

*if*(buff == 0){

*switch*(index)

{

*case*(0):

*return* 1.0;

*case*(1):

*return* 1.0;

*case*(2):

*return* 1.0;

*default*:

*return* 0;

};

}*else* {

*switch*(index)

{

*case*(0):

ik++;

*return* 1.0f + ( ( A\_f[i + 1] \* C\_f\_fun(C\_f[i + 1], i + 1) ) / buff );

*case*(1):

*return* 1;

*case*(2):

*return* 1;

*default*:

*return* 0;

};

}

*return* 0;

};

C\_f\_fun = [&](int index, int i) -> double {*//auto*

*switch*(index)

{

*case*(3):

*if*((x == 0) && (ORD - i) == 0){

*return* A\_a[i]\*1\*C\_a\_fun(C\_a[i], i, 0.0);

}*else*{

*return* A\_a[i]\*qPow(x, ORD - i)\*C\_a\_fun(C\_a[i], i, 0.0);

}

*case*(0):

*//double* *X* *=* *A\_a[i]\*qPow(x,* *ORD* *-* *i)\*C\_a\_fun(C\_a[i],* *i);*

*if*((x == 0) && (ORD - i) == 0){

*return* qTan( A\_a[i] \* 1 \* C\_a\_fun( C\_a[i], i, ( A\_a[i] \* qPow(x, ORD - i) )));*//* *m* *\*(1* *+* *(...* */* *(buff* *=* *m)))* *==* *(m* *+* *...);*

}*else*{

*return* qTan( A\_a[i] \* qPow(x, ORD - i) \* C\_a\_fun( C\_a[i], i, ( A\_a[i] \* qPow(x, ORD - i) )));

}

*case*(1):

*if*((x == 0) && (ORD - i) == 0){

*return* std::log10( A\_a[i] \* 1 \* C\_a\_fun( C\_a[i], i, ( A\_a[i] \* qPow(x, ORD - i) )));*//* *m* *\*(1* *+* *(...* */* *(buff* *=* *m)))* *==* *(m* *+* *...);*

}*else*{

*return* std::log10( A\_a[i] \* qPow(x, ORD - i) \* C\_a\_fun( C\_a[i], i, ( A\_a[i] \* qPow(x, ORD - i) )));

}

*case*(2):

*return* 1;

*default*:

*return* 0;

};

};

*for*(ik = 0; ik < qElem\_s\_1.count(); ik++)

{

xi += A\_f[ik] \* C\_f\_fun(C\_f[ik], ik);

}

*return* xi;

};

*auto* F = [&](double x, double b) -> double {

*return* x - (f(x)/ (f(b) - f(x))) \* (b - x);

};

double x\_0(0), x\_i(0), x\_j(0), b(0);

*if*( f(B[0]) < 0){

x\_0 = B[0];

b = B[1];

}*else* *if* ( f(B[1]) < 0){

x\_0 = B[1];

b = B[0];

}

size\_t count = 0;

*do*{

*if*(count == 0){

x\_i = x\_0;

}*else*{

x\_i = x\_j;

}

count++;

x\_j = F(x\_i, b);

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + "<b>x\_j = </b>"

+ QString::number(x\_j, 'f', Epsilon.qLineEdit\_f->text().length() - 2) + " ;");

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + "<b>Номер итерации: </b>" + QString::number(count, 'd', 0) + " ;");

}*while*(qFabs(x\_j - x\_i) >= eps);

***////////////////////////////////////////////////***

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() +"<hr>" + "<b>Ответ: </b><b>x\_j = </b> "

+ QString::number(x\_j, 'f', Epsilon.qLineEdit\_f->text().length() - 2) + " ;");

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + "<b>Номер итерации: </b>" + QString::number(count, 'd', 0) + " ;");

#### Подпункт 2 (QProject\_5)

4.3.1 Листинг основного кода и пример работы программы к пункту 3.3.1.α.2] Подпункт 2.

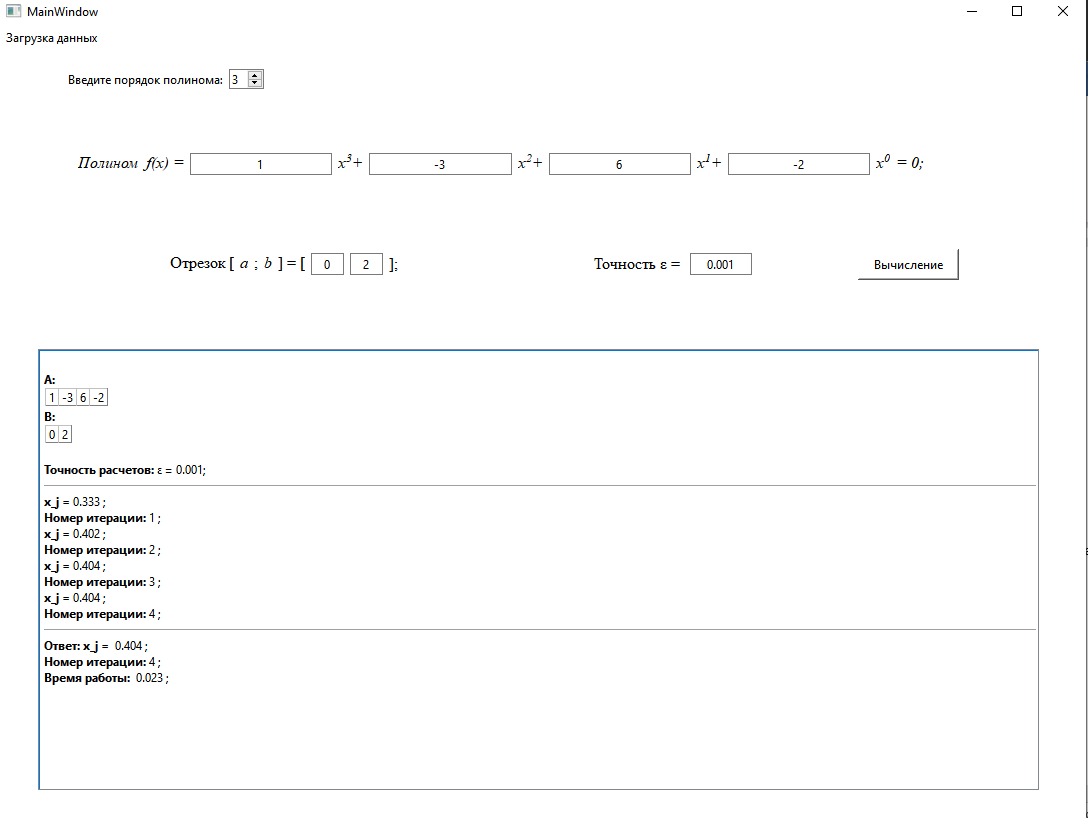


Рис. 3 – Пример работы программы.

# Приложение 2 – Листинг кода.

4.1 Задание № 1

- - -

4.2 Задание № 2

### Пункт № 1:

Ниже представлен листинг основного кода, а также ссылка на GitHub.

*Код:*

*1| auto* s = [ORD = ORD,A = A](double x, size\_t i = 1) -> double {

*2|*

*3|*  double xi = 0;

*4|*

*5|*  *for*(size\_t j = i; j < A.count(); j++){

*6|*

*7|*  xi += ( (-1) \* ( A[j] / A[0] ) )\* qPow(x, (ORD - j));

*8|*  }

*9|*

*10|*  *return* qPow(xi, 1.0/(ORD));

*11|*

*12|*  };

*13|*

*14|*  size\_t count = 0;

*15|*  double x\_0 = B[0];

*16|*  double x\_i(0), x\_j(0);

*17|*

*18|*  *do*{

*19|*  *if*(count == 0){

*20|*  x\_i = x\_0;

*21|*  }*else*{

*21|*  x\_i = x\_j;

*22|*  }

*23|*  count++;

*24|*

*25|*  x\_j = s(x\_i);

*26|*

*27|*  ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + "<b>x\_j = *28|* </b>" + QString::number(x\_j, 'f', Epsilon.qLineEdit->text().length() - 2) *29|* + " ;");

*30|*  ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + "<b>Номер *31|* итерации: </b>" + QString::number(count, 'd', 0) + " ;");

*32|*

*33|*  }*while*(qFabs(x\_j - x\_i) >= eps);

*34|*

Листинг кода № 2 – Основной код.

Ссылка на gitHab: QProject\_4// <https://github.com/MineevS/CHM_3_5.git>

### Пункт № 2:

Ниже представлен листинг основного кода, а также ссылка на GitHub.

Код:

*auto* fi = [&](double x) -> double{

double xi = 0;

size\_t ik = 0;

std::function<double(int index, int i)> C\_f\_fun;

*auto* C\_a\_fun = [&](int index, int i, double buff) -> double {

*if*(buff == 0.0){

*switch*(index)

{

*case*(0):

*return* 1.0;

*case*(1):

*return* 1.0;

*case*(2):

*return* 1.0;

*case*(3):

*return* 1.0;

*case*(4):

*return* x; *//* *qPow(x,* *1)* *~* *x*

*case*(5):

*return* 1; *//* *qPow(x,* *1)* *~* *x*

*case*(6):

*return* 1; *//* *qPow(x,* *1)* *~* *x*

*case*(7):

*return* 1; *//* *qPow(x,* *1)* *~* *x*

*default*:

*return* 0;

};

}*else* {

*switch*(index)

{

*case*(0):

ik++;

*return* 1.0f + ( ( A\_f[i + 1] \* C\_f\_fun(C\_f[i + 1], i + 1) ) / buff );

*case*(1):

*return* 1;

*case*(2):

*return* 1;

*case*(3):

*return* 1;

*case*(4):

*return* 1;

*case*(5):

*return* x; *//* *qPow(x,* *1)* *~* *x*

*case*(6):

*return* x; *//* *qPow(x,* *1)* *~* *x*

*case*(7):

*return* 1; *//* *qPow(x,* *1)* *~* *x*

*default*:

*return* 0;

};

}

*return* 0;

};

C\_f\_fun = [&](int index, int i) -> double {*//auto*

*switch*(index)

{

*case*(0):

*return* qTan( A\_a[i] \* C\_a\_fun( C\_a[i], i, ( A\_a[i] )));

*case*(1):

*return* std::log10( A\_a[i] \* C\_a\_fun( C\_a[i], i, ( A\_a[i] )));

*case*(2):

*return* 1;

*case*(3):

*return* A\_a[i]\*C\_a\_fun(C\_a[i], i, 0.0);

*default*:

*return* 0;

};

};

*for*(ik = 1; ik < qElem\_s\_1.count(); ik++)

{

xi += (-1) \* A\_f[ik] \* C\_f\_fun(C\_f[ik], ik) / A\_a[0]\*A\_f[0];

}

*return* xi;

};

double x\_0(0), x\_i(0), x\_j(0), b(0);

x\_0 = B[0];

size\_t count = 0;

*do*{

*if*(count == 0){

x\_i = x\_0;

}*else*{

x\_i = x\_j;

}

count++;

x\_j = fi(x\_i);

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + "<b>x\_j = </b>" + QString::number(x\_j, 'f', Epsilon.qLineEdit\_f->text().length() - 2) + " ;");

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + "<b>Номер итерации: </b>" + QString::number(count, 'd', 0) + " ;");

}*while*(qFabs(x\_j - x\_i) >= eps);

Листинг кода № 2 – Основной код.

Ссылка на gitHab: QProject\_8// <https://github.com/MineevS/CHM_3_5.git>

4.3 Задание № 3

### Пункт a.

#### Подпункт 1 (QProject\_7)

Код:

*auto* f = [&](double x) -> double{

double xi = 0;

size\_t ik = 0;

std::function<double(int index, int i)> C\_f\_fun;

*auto* C\_a\_fun = [&](int index, int i, double buff) -> double {

*if*(buff == 0){

*switch*(index)

{

*case*(0):

*return* 1.0;

*case*(1):

*return* 1.0;

*case*(2):

*return* 1.0;

*default*:

*return* 0;

};

}*else* {

*switch*(index)

{

*case*(0):

ik++;

*return* 1.0f + ( ( A\_f[i + 1] \* C\_f\_fun(C\_f[i + 1], i + 1) ) / buff );

*case*(1):

*return* 1;

*case*(2):

*return* 1;

*default*:

*return* 0;

};

}

*return* 0;

};

C\_f\_fun = [&](int index, int i) -> double {*//auto*

*switch*(index)

{

*case*(3):

*if*((x == 0) && (ORD - i) == 0){

*return* A\_a[i]\*1\*C\_a\_fun(C\_a[i], i, 0.0);

}*else*{

*return* A\_a[i]\*qPow(x, ORD - i)\*C\_a\_fun(C\_a[i], i, 0.0);

}

*case*(0):

*//double* *X* *=* *A\_a[i]\*qPow(x,* *ORD* *-* *i)\*C\_a\_fun(C\_a[i],* *i);*

*if*((x == 0) && (ORD - i) == 0){

*return* qTan( A\_a[i] \* 1 \* C\_a\_fun( C\_a[i], i, ( A\_a[i] \* qPow(x, ORD - i) )));*//* *m* *\*(1* *+* *(...* */* *(buff* *=* *m)))* *==* *(m* *+* *...);*

}*else*{

*return* qTan( A\_a[i] \* qPow(x, ORD - i) \* C\_a\_fun( C\_a[i], i, ( A\_a[i] \* qPow(x, ORD - i) )));

}

*case*(1):

*if*((x == 0) && (ORD - i) == 0){

*return* std::log10( A\_a[i] \* 1 \* C\_a\_fun( C\_a[i], i, ( A\_a[i] \* qPow(x, ORD - i) )));*//* *m* *\*(1* *+* *(...* */* *(buff* *=* *m)))* *==* *(m* *+* *...);*

}*else*{

*return* std::log10( A\_a[i] \* qPow(x, ORD - i) \* C\_a\_fun( C\_a[i], i, ( A\_a[i] \* qPow(x, ORD - i) )));

}

*case*(2):

*return* 1;

*default*:

*return* 0;

};

};

*for*(ik = 0; ik < qElem\_s\_1.count(); ik++)

{

xi += A\_f[ik] \* C\_f\_fun(C\_f[ik], ik);

}

*return* xi;

};

*auto* F = [&](double x, double b) -> double {

*return* x - (f(x)/ (f(b) - f(x))) \* (b - x);

};

double x\_0(0), x\_i(0), x\_j(0), b(0);

*if*( f(B[0]) < 0){

x\_0 = B[0];

b = B[1];

}*else* *if* ( f(B[1]) < 0){

x\_0 = B[1];

b = B[0];

}

size\_t count = 0;

*do*{

*if*(count == 0){

x\_i = x\_0;

}*else*{

x\_i = x\_j;

}

count++;

x\_j = F(x\_i, b);

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + "<b>x\_j = </b>"

+ QString::number(x\_j, 'f', Epsilon.qLineEdit\_f->text().length() - 2) + " ;");

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + "<b>Номер итерации: </b>" + QString::number(count, 'd', 0) + " ;");

}*while*(qFabs(x\_j - x\_i) >= eps);

***////////////////////////////////////////////////***

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() +"<hr>" + "<b>Ответ: </b><b>x\_j = </b> "

+ QString::number(x\_j, 'f', Epsilon.qLineEdit\_f->text().length() - 2) + " ;");

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + "<b>Номер итерации: </b>" + QString::number(count, 'd', 0) + " ;");

#### Подпункт 2 (QProject\_5)

Код:

*auto* f = [A = A, ORD = ORD](double x) -> double {

double xi = 0;

*for*(size\_t i = 0; i <= ORD; i++){

*if*(i == ORD){

xi += A[i]\*1;

}*else*{

xi += A[i]\*qPow(x, ORD - i);

}

}

*return* xi;

};

*//d\_1* *-* *derivative* *1* *-* *Первая* *производная:*

*auto* f\_d\_1 = [A = A, ORD = ORD](double x) -> double {

double xi = 0;

*for*(size\_t i= 0; i < ORD; i++){

xi += (ORD - i)\* A[i] \* qPow(x, (ORD - i) - 1);

}

*return* xi;

};

*//d\_2* *-* *derivative* *1* *-* *Первая* *производная:*

*auto* f\_d\_2 = [A = A, ORD = ORD](double x) -> double {

double xi = 0;

*for*(size\_t i= 0; i < ORD; i++){

xi += ((ORD - i) - 1)\*(ORD - i)\* A[i] \* qPow(x, (ORD - i) - 2);

}

*return* xi;

};

*auto* s = [&,ORD = ORD,A = A](double xj, size\_t i = 1) -> double {

*return* ( xj - ( f(xj) / f\_d\_1(xj) ) );

};

size\_t count = 0;

double x\_0 = 0;

*if*(f(B[0])\*f\_d\_2(B[0]) > 0){

x\_0 = B[0];

}*else* *if* (f(B[1])\*f\_d\_2(B[1]) > 0){

x\_0 = B[1];

}*else*{

qDebug() << "Отрезок отделения указан неверно!";

exit(0);

}

double x\_i(0), x\_j(0);

*do*{

*if*(count == 0){

x\_i = x\_0;

}*else*{

x\_i = x\_j;

}

count++;

x\_j = s(x\_i);

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + "<b>x\_j = </b>" + QString::number(x\_j, 'f', Epsilon.qLineEdit->text().length() - 2) + " ;");

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + "<b>Номер итерации: </b>" + QString::number(count, 'd', 0) + " ;");

}*while*(qFabs(x\_j - x\_i) >= eps);

Ссылка на gitHab: QProject\_5// <https://github.com/MineevS/CHM_3_5.git>

### Пункт b.

4.3.2 Листинг основного кода к пункту 3.3.1.b]

*auto* f = [A = A, ORD = ORD](double x) -> double {

double xi = 0;

*for*(size\_t i = 0; i <= ORD; i++){

*if*(i == ORD){

xi += A[i]\*1;

}*else*{

xi += A[i]\*qPow(x, ORD - i);

}

}

*return* xi;

};

*//d\_1* *-* *derivative* *1* *-* *Первая* *производная:*

*auto* f\_d\_1 = [A = A, ORD = ORD](double x) -> double {

double xi = 0;

*for*(size\_t i= 0; i < ORD; i++){

xi += (ORD - i)\* A[i] \* qPow(x, (ORD - i) - 1);

}

*return* xi;

};

*//d\_2* *-* *derivative* *1* *-* *Первая* *производная:*

*auto* f\_d\_2 = [A = A, ORD = ORD](double x) -> double {

double xi = 0;

*for*(size\_t i= 0; i < ORD; i++){

xi += ((ORD - i) - 1)\*(ORD - i)\* A[i] \* qPow(x, (ORD - i) - 2);

}

*return* xi;

};

*auto* s = [&,ORD = ORD,A = A](double xj, double \_xj) -> double {

*return* ( xj - ( f(xj) / (f(\_xj) - f(xj)) ) \* (\_xj - xj) );

};

*auto* \_s = [&](double \_xj) -> double {

*return* ( \_xj - ( f(\_xj) / f\_d\_1(\_xj) ) );

};

size\_t count = 0;

double x\_n = B[0];

double \_x\_n = B[1];

double x\_i(0), x\_j(0), \_x\_i(0), \_x\_j(0);

*do*{

*if*(count == 0){

x\_i = x\_n;

\_x\_i = \_x\_n;

}*else*{

x\_i = x\_j;

\_x\_i = \_x\_j;

}

count++;

x\_j = s(x\_i, \_x\_i);

\_x\_j = \_s(\_x\_i);

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + "<b> x\_j = </b>" + QString::number( x\_j, 'f', Epsilon.qLineEdit->text().length() - 2) + " ;");

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + "<b>\_x\_j = </b>" + QString::number(\_x\_j, 'f', Epsilon.qLineEdit->text().length() - 2) + " ;");

ui->textBrowser\_1->setHtml(ui->textBrowser\_1->toHtml() + "<b>Номер итерации: </b>" + QString::number(count, 'd', 0) + " ;");

}*while*(qFabs(\_x\_j - x\_j) >= eps);

double csi = (x\_j + \_x\_j) / 2;

Ссылка на gitHab: QProject\_6// <https://github.com/MineevS/CHM_3_5.git>

# Список использованных источников

1. <https://vega.fcyb.mirea.ru/disc/disc.php?id=163>
2. <https://online-edu.mirea.ru/course/view.php?id=2154>

# Как раскрыть все заголовки в документе:

