|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | |  |  |  |
| МИНОБРНАУКИ РОССИИ | | | |  |
| |  | | --- | |  | | Федеральное государственное бюджетное образовательное учреждение  высшего образования  **«МИРЭА – Российский технологический университет»**  **РТУ МИРЭА** | | | | |  |
|  | **Институт кибернетики**  *(наименование института)* | | | |
|  | **Кафедра высшей математики**  *(наименование кафедры)* | | | |

|  |  |
| --- | --- |
| **Отчет по лабораторной работе №1** | |
| **по дисциплине** | |
| **«** Численные методы **»** | |
| **Вариант 13** | |
| Студент 3-го курса  группы КМБО-2-16 | Савин В. О. |
| Преподаватель | Даева С.Г. |
| Рецензент |  |
|  |  |

|  |  |  |
| --- | --- | --- |
| Работа представлена к защите | «\_\_»\_\_\_\_\_\_\_2018 г. |  |
|  |  |  |
| «Допущен к защите» | «\_\_»\_\_\_\_\_\_\_2018 г. |  |

Москва 2018

Содержание

Задание №1 3

Теоретическая часть 3

Практическая часть 4

Задание №2 5

Теоретическая часть 5

Практическая часть 6

Задание №3 7

Теоретическая часть 8

Практическая часть 10

Приложения 12

# Задание №1

1. Определить, какое равенство точнее:
2. Округлить сомнительные цифры числаоставив верные знаки, и определить абсолютную погрешность результата.
3. Найти предельные абсолютные и относительные погрешности числа если оно имеет только верные цифры.
4. Вычислить и определить погрешность результата, если исходные числа заданы в верных знаках. Записать результат в верных знаках.

где a*=0.643, b=2.17, c=5.843*

## Теоретическая часть

Приближенным числом *a* называется число, отличающееся от точного числа *A* и заменяющее точное число в вычислениях. Тогда числоназывается абсолютной погрешностью приближенного числа *a*.

Предельной абсолютной погрешностьючисла *a* называется любое число не меньшее

Относительной погрешностью приближенного числа а называется число

Предельной относительной погрешностьюприближенного числа а называется любое число не меньшее

Цифра стоящая на n-ом месте в записи приближенного числа называется верной, если абсолютная погрешность этого числа не превышает десятичного разряда, выраженного n-ой цифрой данного числа.

Справедлива теорема, утверждающая, что если приближенное число *a > 0* имеет  *n > 2* верных знаков, то относительная погрешность где — первая значащая цифра числа.

Пусть для функцииизвестны— абсолютные погрешности аргументов. Тогда абсолютная погрешность функции

**Практическая часть**

**1)** a = 13/17 = 0.764 b = √31 = 5.56

**Ответ:** √31 точнее, чем 13/17

**2)**

*1. Найдем абсолютную погрешность числа a:*

*2. Округлим a до десятых:*

*Теперь число записано в верных знаках:* 15.9

**Ответ:** 15.9

**3)**

*Поскольку число дано в верных знаках:*

*Вычислим предельную относительную погрешность:*

**Ответ:**

**4)**  a*=0.643, b=2.17, c=5.843*

*1. Найдем абсолютные погрешности:*

*Поскольку числа даны в верных знаках, то:*

*2. Вычислим выражение:*

*3. Вычислим производные:*

*4. Вычислим предельную погрешность выражения Х:*

*5. Приведем число к верным знакам:*

*Округлим до десятых:* X = 1.1

Ответ: X = 1.1

# 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.

## Теоретическая часть

Пользуясь схемой Горнера можно вычислить границы действительных корней многочлена Если в схеме Горнера длявсе коэффициентыито все действительные корни многочлена, если они есть, расположены не правееДля оценки нижней границы необходимо описанным выше способом найти верхнюю границу многочленапри этом нижняя граница многочленабудет равна

Для отделения корней, можно воспользоваться тем фактом, что непрерывная функция *f(x)* принимает значения разных знаков на концах отрезка [*a, b*], то внутри отрезка содержится по крайней мере один корень уравнения *f(x) = 0*. Причем тот корень будет единственным, если на отрезке существует производная *f΄(x),* и она сохраняет на нем знак.

Для уточнения корняуравнения *f(x) =* 0, где *f(x)* — непрерывная на данном отрезке функция,с заданной точностью *ε*, можно использовать метод итерации. Его суть заключается в эквивалентном преобразовании исходного уравнения к виду *x = φ(x),*  выборе начального приближенияи вычислением последовательностидо тех пор, покатак как можно показать, чтоПри этом преобразование законно только тогда, когда *φ(x)* дифференцируема на [*a, b*] ив этом случае метод сходится независимо от выбора начального приближения.

## Практическая часть

1) Уточним корнииспользуя схему Горнера.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1 | -0.2 | 0.5 | -1.4 |
| 2 | 1 | 1.8 | 4.1 | 6.8 |

Верхняя граница *P(x):*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 1 | 0.2 | 0.5 | 1.4 |
| 1 | 1 | 1.2 | 1.7 | 3.1 |

Нижняя граница *P(x):* Получили:

Проверим: *P(1) = -0.1 < 0; P(2) = 6.8 > 0;* следовательно на данном отрезке есть хотя бы один корень.

Уточним корень методом итерации. Преобразуем исходное уравнение:

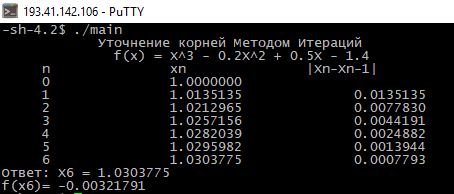
на [1, 2], следовательновозрастает на данном отрезке, тогда:

Оценим

следовательно метод итерации сходится независимо от начального приближения.

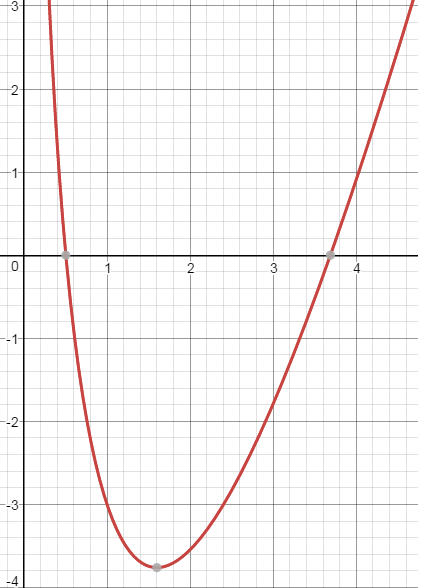
Для расчета последовательностидо тех пор, покапри данных точности *ε,* функции *φ(x)* и начальном приближении была написана программа «Lab1». Программа написана на языке «C++» в операционной системе «Windows7» с использованием кроссплатформенной IDE «CodeBlocks» и компилятора «GNU GCC». Для отладки проводились тесты программы на примерах с заранее известным ответом.

Для данной задачи имеем: необходимую точность *ε = 0.001,* функциюВ качестве начального приближения берем точку



Ответ: приближенный корень

2)

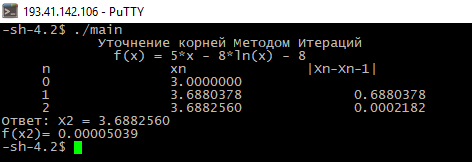
*****График функции f(x):*

Графически отделим корни.

Рассмотрим корень на отрезкеметодом итераций.

От*вет*: приближенный корень

*Вывод программы:*



# Задание №3

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

1. методом хорд:
2. методом касательных (метод Ньютона):
3. комбинированным методом:

## Теоретическая часть

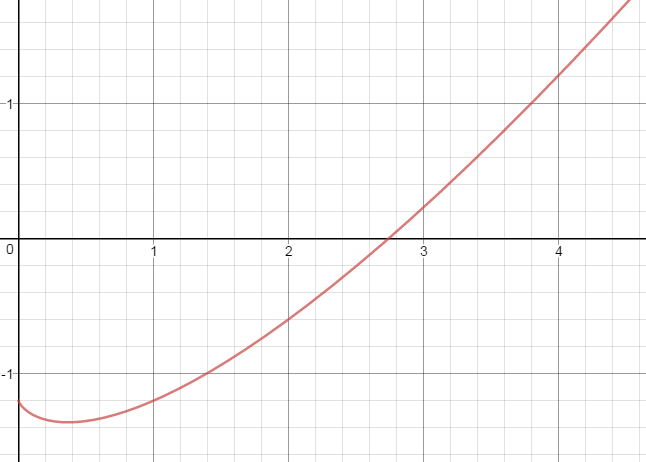
Для уточнения корняуравнения *f(x) =* 0, где функция *f(x)* имеет непрерывные и сохраняющие знак на данном отрезке первую и вторую производные,с заданной точностью *ε*, можно использовать метод хорд. Его суть заключается в построении последовательности приближенийгде - точка пересечения оси абсцисс и хорды между значениями функции в точках *c* иТочка *с —* неподвижный конец отрезка [a, b], в котором знак функции совпадает со знаком второй производной. В качестве начального приближениявыбирается тот конец отрезка [a, b], в котором знак функции не совподает со знаком второй производной. Оценка погрешности метода производится по следующей формуле: Где *M* и *m —* соответственномаксимальное и минимальное значениена отрезке [a, b]. Таким образом, для получения заданной точности *ε,* необходимо рассчитывать приближения до тех пор, пока не будет выполнено неравенство:

Для уточнения корняуравнения *f(x) =* 0, где функция *f(x)* имеет непрерывные и сохраняющие знак на данном отрезке первую и вторую производные, применяется метод касательных. Метод состоит в вычислении последовательности приближенийМожно показать, что если функция на концах отрезка принимает разные знаки, аиотличны от нуля и сохраняют знаки, то при выборе начального приближенияпоследовательность приближений будет сходится. Оценка погрешности метода производится по следующей формуле: Где*-* максимальное значение модуля второй производной, минимальное значение модуля первой производной на отрезке [a, b]. Таким образом, для получения заданной точности *ε,* необходимо рассчитывать приближения до тех пор, пока не будет выполнено неравенство:

Если метод хорд дает приближение по избытку или недостатку, то метод касательных — по недостатку или избытку соответственною. Таким образом для уточнения корня можно использовать оба способа одновременно. Этот метод называется комбинированным. На каждой итерации приближение, полученное методом касательных, используется в качестве неподвижного конца для метода хорд. Погрешность оценивается по следующей формуле:где- приближения, полученные методом касательных,- приближения, полученные методом хорд.

**Практическая часть**

**1)** Графически отделим корни:

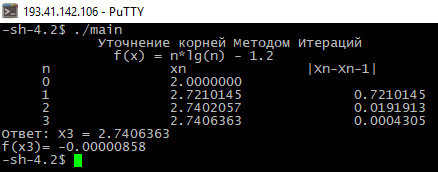


Уточним корень на отрезке методом хорд. Как видно из графика,на данном отрезке, следовательно 3 — неподвижный конец, а 2 - начальное приближение

Рассчитаем последовательностьдо тех пор пока не выполнится неравенство

Результат: приближенный корень

*Вывод программы:*

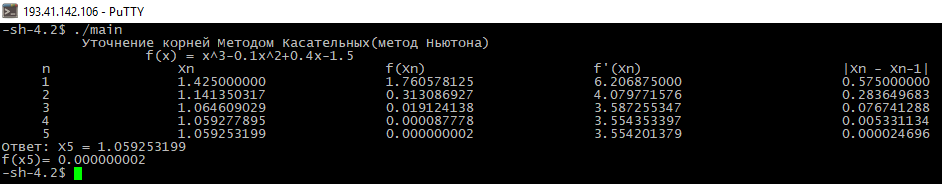


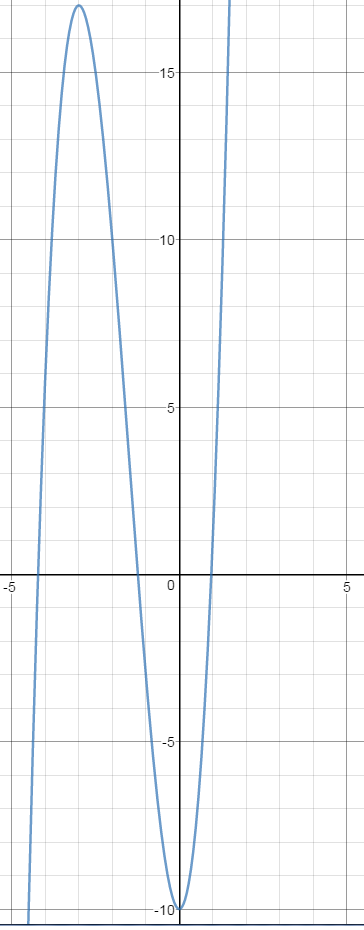
**2)**

Следовательно корень находится на отрезке [0, 1]. Уточним корень используя метод касательных.

Возьмем начальное приближениеРассчитаем последовательностьдо тех пор, пока не выполнится неравенство

**Ответ:** приближенный корень

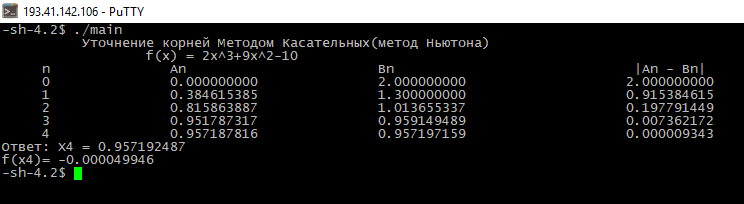


**3)**

Графически отделим корни. Уточним корень на отрезке [0, 2] комбинированным методом. Рассчитаем последовательности

до тех пор, пока не выполнится неравенство

**Ответ:** приближенный корень



# Приложения

*Код программы «» задания 2.1:*

#include <stdio.h>

#include <math.h>

double f(double x)

{

return (x\*x\*x - 0.2\*x\*x + 0.5\*x - 1.4);

};

int main()

{

double arr[50];

arr[0] = 1;

printf( " Уточнение корней Методом Итераций\n f(x) = X^3 - 0.2X^2 + 0.5X - 1.4\n ""n"" ""xn"" ""|Xn-Xn-1|""\n ""0"" %.7f \n",arr[0]);

int i;

for ( i = 1; ; ++i)

{

arr[i] = arr[i-1] - f(arr[i-1])/7.4;

printf(" %d %.7f %.7f\n",i,arr[i],(arr[i]-arr[i-1]));

if ( fabs(arr[i] - arr[i-1]) < 0.001 )

{

printf("Ответ: Х%d = %.7f \n",i,arr[i]);

printf("f(x%d)= %.8f\n",i,f(arr[i]));

break;

}

else continue;

}

return 0;

}

*Код программы «» задания 2.2:*

#include <stdio.h>

#include <math.h>

double f(double x)

{

return (5\*x - 8\*log(x) - 8);

};

int main()

{

double arr[20];

arr[0] = 3;

printf( " Уточнение корней Методом Итераций\n f(x) = 5\*x - 8\*ln(x) - 8\n ""n"" ""xn"" ""|Xn-Xn-1|""\n ""0"" %.7f \n",arr[0]);

int i;

for ( i = 1; ; ++i)

{

arr[i] = arr[i-1] - f(arr[i-1])/2.6;

printf(" %d %.7f %.7f\n",i,arr[i],(arr[i]-arr[i-1]));

if ( fabs(arr[i] - arr[i-1]) < 0.001 )

{

printf("Ответ: Х%d = %.7f \n",i,arr[i]);

printf("f(x%d)= %.8f\n",i,f(arr[i]));

break;

}

else continue;

}

return 0;

}

*Код программы «» задания 3.1:*

#include <stdio.h>

#include <math.h>

double f(double n)

{

return n\*log10(n) - 1.2;

}

int main()

{

double arr[20];

arr[0] = 2;

printf( " Уточнение корней Методом Итераций\n f(x) = n\*lg(n) - 1.2\n ""n"" ""xn"" ""|Xn-Xn-1|""\n ""0"" %.7f \n",arr[0]);

int i;

for (i = 1; ; ++i)

{

arr[i] = arr[i-1] - ( f(arr[i-1]) / ( f(arr[i-1]) - f(3) ) ) \* (arr[i-1] - 3);

printf(" %d %.7f %.7f\n",i,arr[i],(arr[i]-arr[i-1]));

if ( fabs(arr[i] - arr[i-1]) < 0.004452 )

{

printf("Ответ: Х%d = %.7f \n",i,arr[i]);

printf("f(x%d)= %.8f\n",i,f(arr[i]));

break;

}

else continue;

}

return 0;

}

*Код программы «» задания 3.2:*

#include <stdio.h>

#include <math.h>

double func(double n)

{

return (n\*n\*n-0.1\*n\*n+0.4\*n-1.5);

}

double prFunc(double n)

{

return 3\*n\*n-0.2\*n+0.4;

}

int main()

{

double arr[20];

arr[0] = 2;

printf(" Уточнение корней Методом Касательных(метод Ньютона)\n");

printf(" f(x) = x^3-0.1x^2+0.4x-1.5\n");

printf(" ""n"" ""Xn"" ""f(Xn)"" ""f'(Xn)"" ""|Xn - Xn-1|""\n");

int i;

for ( i = 1; i < 12; ++i)

{

arr[i] = arr[i-1] - ( func(arr[i-1]) / prFunc(arr[i-1]) );

printf(" %d %.9f %.9f %.9f %.9f\n",i,arr[i],func(arr[i]),prFunc(arr[i]),fabs(arr[i]-arr[i-1]));

if ( fabs(arr[i] - arr[i-1]) < 0.0002)

{

printf("Ответ: Х%d = %.9f \n",i,arr[i]);

printf("f(x%d)= %.9f\n",i,func(arr[i]));

break;

}

else continue;

}

return 0;

}

*Код программы «» задания 3.3:*

#include <stdio.h>

#include <math.h>

double func(double n)

{

return (2\*n\*n\*n+9\*n\*n-10);

}

double prFunc(double n)

{

return (6\*n\*n+18\*n);

}

int main()

{

double arrA[50];

double arrB[50];

arrA[0] = 0;

arrB[0] = 2;

printf(" Уточнение корней Методом Касательных(метод Ньютона)\n");

printf(" f(x) = 2x^3+9x^2-10\n");

printf(" ""n"" ""An"" ""Bn"" ""|An - Bn|""\n");

printf(" ""0"" %.9f %.9f %.9f\n",arrA[0],arrB[0],fabs(arrA[0]-arrB[0]));

int i;

for (i = 1; ; ++i)

{

arrA[i] = arrA[i-1] - (( func(arrA[i-1]) ) / (func(arrB[i-1]) - func(arrA[i-1]) )) \* (arrB[i-1] - arrA[i-1]) ;

arrB[i] = arrB[i-1] - ( func(arrB[i-1]) ) / (prFunc(arrB[i-1])) ;

printf(" %d %.9f %.9f %.9f\n",i,arrA[i],arrB[i],fabs(arrA[i]-arrB[i]));

if ( fabs(arrA[i] - arrB[i]) < 0.001)

{

double result = 0.5\*(arrA[i]+arrB[i]);

printf("Ответ: Х%d = %.9f \n",i,result);

printf("f(x%d)= %.9f\n",i,func(result));

break;

}

else continue;

}

return 0;

}