Министерство науки и образования Российской Федерации

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Институт математики и информационных технологий

Кафедра программного обеспечения вычислительной техники и автоматизированных систем

**ОТЧЕТ**

по лабораторной работе № 1

по дисциплине «Программная инженерия задач вычислительной математики»

**Численные методы решения нелинейных уравнений.**

ОГУ 09.03.04.4024. 704 ПЗ

Руководитель

канд. техн. наук, доцент

\_\_\_\_\_\_\_\_\_\_\_\_\_ Е. А. Шнякина

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г.

Исполнитель

Студент группы 22ПИнж(б)РПиС-1

\_\_\_\_\_\_\_\_\_\_\_\_\_\_ И.В. Федоров

«\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2024 г.

Оренбург 2024

**Постановка задачи. Отделение корней**

Пусть функция *f(x)* определена и непрерывна в некоторой области *D*. Необходимо найти корни уравнения

*f(x)=0*

на отрезке

**Аналитический метод отделения корней**

Рассмотрим уравнение

на отрезке [-2;3] вычислим корни с точностью *ε = 10-3*.

Определим количество корней на заданном отрезке и разобьём его на части, содержащие по одному корню.

1. Находим производную

2) Решаем уравнение *f `(x)=0*:

3) Проанализируем поведение функции на отрезке [-2; 3].

На отрезке [-2; 3] функция имеет одинаковые знаки на концах

.

Следовательно, решая по аналитическому методу данное уравнение мы не можем понять есть ли у него корни.

Результаты проведения аналитического метода отделения корней:

1) уравнение на отрезке [-2;3] не имеет корней ( по крайней мере судя по аналитическому методу)

**Графический метод отделения корней**



Рисунок 2. График функции на отрезке [-2;3].

Из графика видно, что функция

пересекается с осью ОХ два раза. Выделим наименьшие отрезки, опираясь на цену деления оси х построенного графика, который содержит корни. Получаем отрезки [-1,4; -1,2] и [2;2,2].

Графический метод отделения корней дал информацию о количестве корней уравнения и позволил увидеть длину отрезка, содержащих корни уравнения.

**Табулирование функции**

1. Проведем табулирование функции

с шагом .

Результаты табулирования функции приведены в таблице 1.

Таблица 1 - Табулирование функции 

|  |  |  |
| --- | --- | --- |
| -2 | 13,5522 |  |
| -1,8 | 9,313182 |  |
| -1,6 | 5,389942 |  |
| **-1,4** | 1,780395 | `+ |
| **-1,2** | -1,51827 | `- |
| -1 | -4,51 |  |
| -0,8 | -7,20068 |  |
| -0,6 | -9,59986 |  |
| -0,4 | -11,7253 |  |
| -0,2 | -13,62 |  |
| 0 | -15,61 |  |
| 0,2 | -15,22 |  |
| 0,4 | -14,9253 |  |
| 0,6 | -14,3999 |  |
| 0,8 | -13,6007 |  |
| 1 | -12,51 |  |
| 1,2 | -11,1183 |  |
| 1,4 | -9,41961 |  |
| 1,6 | -7,41006 |  |
| 1,8 | -5,08682 |  |
| **2** | -2,4478 | `- |
| **2,2** | 0,508614 | `+ |
| 2,4 | 3,783686 |  |
| 2,6 | 7,378443 |  |
| 2,8 | 11,29373 |  |
| 3 | 15,53025 |  |

Вывод: нашли отрезки, содержащие корни: [-1,4; 1,2], [2;2,2].

Для дальнейшего этапа решения нелинейных уравнений – уточнения корней – из найденных разными методами отрезков, выбираем тот, который имеет наименьшую длину: [2; 2,2].

**Уточнение корней**

**Метод бисекций (половинного деления)**

Суть метода заключается в последовательном делении отрезка, содержащего единственный корень уравнения, пополам. Середина отрезка, длина которого не превышает заданную точность, является корнем уравнения.

Условие применимости метода: существование единственного корня на отрезке.

Алгоритм решения нелинейного уравнения методом бисекций

1. *i=1* (номер итерации)*,*
2. *ai=a; bi=b*
3. если *,* то: *сi* - корень уравнения

иначе: шаг 5

1. если *|bi - ai|≤ ε*, то: – корень уравнения

иначе: *i:=i+1,* шаг 6

1. если *f(ai-1)·f(ci-1)<0,* то: *bi=ci-1 ai=ai-1,* шаг 3

иначе: *ai=ci-1 bi=bi-1*, шаг 3

Алгоритм был реализован. Результат уточнения на отрезке [2;2,2] с точностью изображен на рисунке 3.

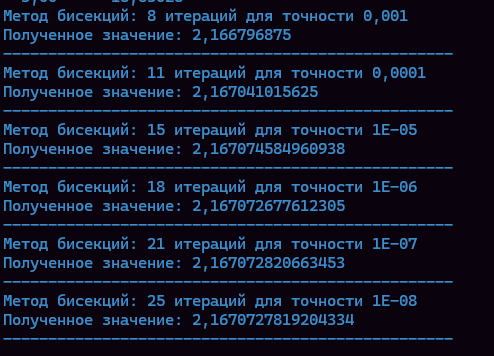
****

Рисунок 3 – Уточнение корня на отрезке [2;2,2] методом бисекций

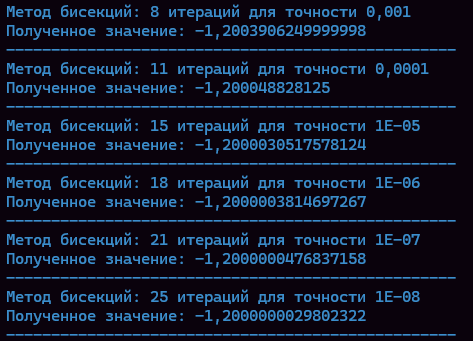


Рисунок 4 – Уточнение корня на отрезке [-1,4;-1,2] методом бисекций

**Метод Ньютона**

Идея метода Ньютона (касательных) заключается в последовательном приближении к корню с помощью касательных, проведённых к графику функции *f(x)* в текущей точке итерационной последовательности.

Условия применимости метода:

Пусть *f(x)* на концах отрезка [a; b] имеет значения разных знаков, т.е.

,

причем и в любой точке отрезка (т.е. сохраняют знак на отрезке [a; b]), тогда, исходя из начального приближения х0, удовлетворяющего условию , можно вычислить корень уравнения *f(x)=0* методом Ньютона с точностью ε.

В качестве начальн*ого приближения выбирается одна из границ отрезка: если , иначе .*

Алгоритм решения нелинейного уравнения методом Ньютона

* + - 1. если *,* то *,* иначе
      2. *i=1* (номер итерации)
      3. если , то: – корень уравнения,

иначе: , шаг 3

Для отрезка [2;2,2] проверяем условие для того чтобы понять какое значение брать в .

Так как то .

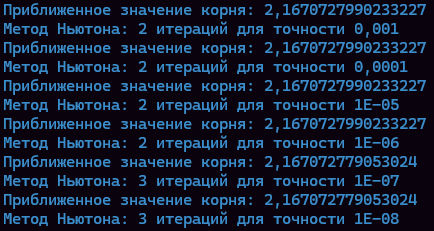
****

Рисунок 5 - Уточнение корня на отрезке [2;2,2] методом Ньютона

Теперь уточним корень на отрезке [-1,4;-1,2]. Опять проверим условие :

Так как то .

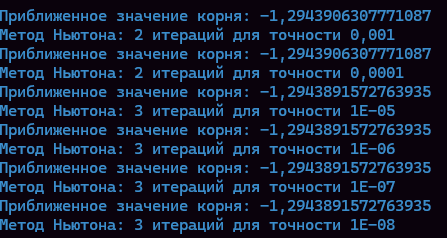


Рисунок 6 - Уточнение корня на отрезке [-1,4;-1,2] методом Ньютона

**Метод хорд**

Суть метода заключается последовательном приближении к корню с помощью хорд - прямых, которыми заменяется нелинейная функция *f(x).* Хорды проводятся через точки с координатами (a; f(a)), (b; f(b)), где а и b границы отрезков, полученных на каждой итерации, и содержащих корень уравнения. Точки пересечения хорд с осью абсцисс составляют сходящуюся к корню последовательность.

Условие применимости метода: существование единственного корня на отрезке.

Алгоритм решения нелинейного уравнения методом хорд

1. x0=a, x1=b
2. *i=1* (номер итерации)
3. если , то: – корень уравнения

иначе:шаг 5

1. если *f(a)·f(xi+1)<0*, то: *b=xi+1*

иначе: *a= xi+1*

1. *i,* шаг 3

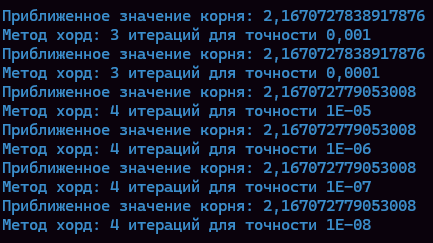
****

Рисунок 7 - Уточнение корня на отрезке [2;2,2] методом хорд

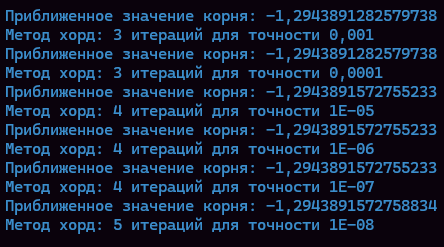


Рисунок 8 - Уточнение корня на отрезке [-1,4;-1,2] методом хорд

**Метод простых итераций**

Заменим уравнение *f(x)=0* равносильным уравнением вида: *x=φ(x)*.

На основе полученного уравнения, строится итерационная последовательность вида:

*xi=φ(xi-1)*.

Пусть функция *φ(x)* определена и дифференцируема на отрезке , тогда, если выполняются следующие условия:

* + - * 1. для любого ;
        2. существует такое вещественное число α, что для любого ,

тогда итерационная последовательность *xi=φ(xi-1)* сходится при любом начальном приближении .

Алгоритм решения нелинейного уравнения методом простых итераций

(выбирается произвольно любое значение из )

* 1. *i=1* (номер итерации)
  2. *xi=φ(xi-1)*
  3. если , то: – корень уравнения,

иначе: , шаг 3

Уточним корень на отрезке [-1,4;-1,2].

От исходного уравнения *f(x)=0* перейдем к уравнению *x=φ(x)*:

Т.е.

Для сходимости вычислительного процесса метода простых итераций, необходимо, чтобы выполнилось условие . Проверим его выполнимость.

Находим

На отрезке [-1,4;-1,2] функция возрастает

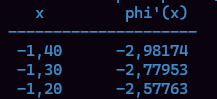


Рисунок 9 – результаты вычисления значения в точке -1,4

.

Также проверим условие для любого .

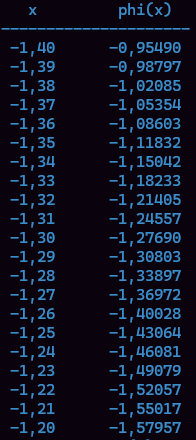


Рисунок 10 – проверка условия сходимости

Как видим условие что для любых принадлежащих отрезку [-1,4;-1,2] выполняется не выполняется.

Поэтому условие сходимости не выполняется.

Возьмем другую функцию:

Перейдем к уравнению вида

Т.е.

Где:

Тогда должно выполняться условие

И c должно равняться:

Тогда функция для метода простых итераций будет выглядеть так:

Применим метод простых итераций для отрезка [-1,4;-1,2]

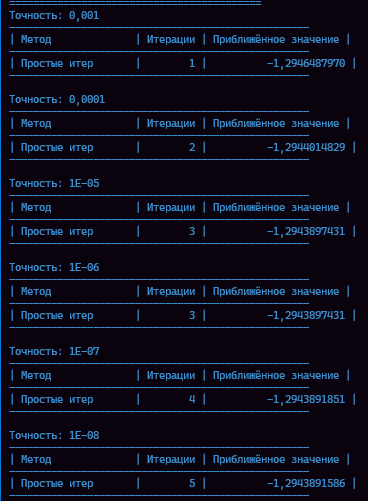


Рисунок 11 – Результат метода простых итераций на отрезке [-1,4;-1,2]

Теперь проверим отрезок [2;2,2].

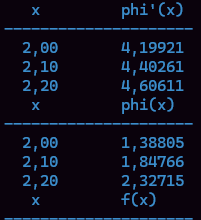


Рисунок 12 – проверка условия сходимости на отрезке [2;2,2]

Для этой функции которую мы подобрали условия применимости метода простых итераций не выполняются. Найдем другую функцию

От исходного уравнения *f(x)=0* перейдем к уравнению *x=φ(x)*:

Т.е.

Для сходимости вычислительного процесса метода простых итераций, необходимо, чтобы выполнилось условие . Проверим его выполнимость.

Находим

На отрезке [-1,4;-1,2] функция убывает

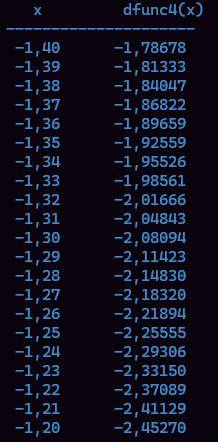


Рисунок 13 – результаты вычисления значения на отрезке [-1,4;-1,2].

Условие сходимости не выполняется на отрезке [-1,4;-1,2].

Теперь проверим отрезок [2;2,2].

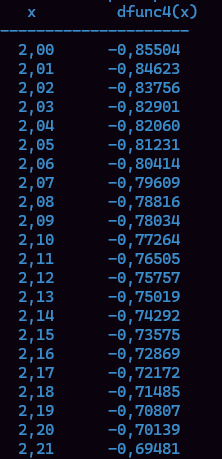


Рисунок 14 – проверка условия сходимости на отрезке [2;2,2]

На этом отрезке мы можем применить метод простых итераций, так как для этого отрезка мы можем найти такое альфа, что все значения производной на этом отрезке будут меньше альфа и 1 по модулю.

Отрезок [-1,4;-1,2]

Таблица 1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Метод    Точность | Метод бисекций | | Метод Ньютона | | Метод хорд | | Метод простых итераций | |
| Значение корня | Количество итераций | Значение корня | Количество итераций | Значение корня | Количество итераций | Значение корня | Количество итераций |
| *10-3* | -1,2941406250 | 8 | -1,2943891283 | 2 | -1,2943891283 | 3 | -1,2946487970 | 1 |
| *10-4* | -1,2943847656 | 11 | -1,2943891283 | 2 | -1,2943891283 | 3 | -1,2944014829 | 2 |
| *10-5* | -1,2943878174 | 15 | -1,2943891573 | 3 | -1,2943891573 | 4 | -1,2943897431 | 3 |
| *10-6* | -1,2943889618 | 18 | -1,2943891573 | 3 | -1,2943891573 | 4 | -1,2943897431 | 3 |
| *10-7* | -1,2943892002 | 21 | -1,2943891573 | 3 | -1,2943891573 | 4 | -1,2943891851 | 4 |
| *10-8* | -1,2943891555 | 25 | -1,2943891573 | 3 | -1,2943891573 | 5 | -1,2943891586 | 5 |

Отрезок [2;2,2]

Таблица 2

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Метод    Точность | Метод бисекций | | Метод Ньютона | | Метод хорд | | Метод простых итераций | |
| Значение корня | Количество итераций | Значение корня | Количество итераций | Значение корня | Количество итераций | Значение корня | Количество итераций |
| *10-3* | 2,1667968750 | 8 | 2,1670727990 | 2 | 2,1670727839 | 3 | 2,1676080788 | 15 |
| *10-4* | 2,1670410156 | 11 | 2,1670727990 | 2 | 2,1670727839 | 3 | 2,1670171064 | 22 |
| *10-5* | 2,1670745850 | 15 | 2,1670727990 | 3 | 2,1670727791 | 4 | 2,1670785702 | 29 |
| *10-6* | 2,1670726776 | 18 | 2,1670727990 | 3 | 2,1670727791 | 4 | 2,1670732150 | 37 |
| *10-7* | 2,1670728207 | 21 | 2,1670727791 | 3 | 2,1670727791 | 4 | 2,1670727337 | 44 |
| *10-8* | 2,1670727819 | 25 | 2,1670727791 | 3 | 2,1670727791 | 5 | 2,1670727838 | 51 |

Рисунок 11 – График зависимости количества итераций на отрезке [-1,4;-1,2]

Рисунок 12 – График зависимости количества итераций на отрезке [2;2,2]