Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра информатики

Отчет по лабораторной работе №3

Численное решение нелинейных уравнений

Выполнил:

студент группы 053506

Ермолович Д.C

Руководитель:

доцент

Анисимов В.Я.

Минск 2022

Оглавление

[Цель работы 3](#_Toc98491710)

[Теоретический сведения 3](#_Toc98491711)

[**Метод простых итераций**. 6](#_Toc98491712)

[**Метод хорд**. 7](#_Toc98491713)

[**Метод Ньютона** (**касательных**). 8](#_Toc98491714)

[**Метод релаксации:** 9](#_Toc98491715)

[**Метод половинного деления** 10](#_Toc98491716)

[**Метод Эйткена-Стеффенсона** 11](#_Toc98491717)

[**Модификации метода Ньютона** 12](#_Toc98491718)

[**Упрощенный метод Ньютона** 12](#_Toc98491719)

[**Метод ложного положения** 12](#_Toc98491720)

[**Метод секущих** 13](#_Toc98491721)

[Программная реализация 13](#_Toc98491722)

[Код: 13](#_Toc98491723)

[Тестовые примеры 19](#_Toc98491724)

[Вывод: 21](#_Toc98491725)

# Цель работы

1. Изучить методы численного решения нелинейных уравнений –методов бисекции, хорд, простой итерации, релаксации, метода Ньютона и его модификаций.
2. Исследовать скорость сходимости итерационных процедур
3. Изучить метод Эйткена ускорения сходимости
4. Составить программу численного решения нелинейных уравнений методами бисекции, хорд, Ньютона
5. Проверить правильность работы программы на тестовых примерах
6. Численно решить нелинейное уравнение заданного варианта
7. Сравнить число итераций, необходимого для достижения заданной точности вычисления разными методами

# Теоретический сведения

Численное решение нелинейного уравнения f(x)=0 заключается в вычислении с заданной точностью значения всех или некоторых корней уравнения и распадается на несколько задач: во-первых, надо исследовать количество и характер корней (вещественные или комплексные, простые или кратные), во-вторых, определить их приближенное расположение, т.е. значения начала и конца отрезка, на котором лежит только один корень, в- третьих, выбрать интересующие нас корни и вычислить их с требуемой точностью. Вторая задача называется отделением корней. Решив ее, по сути дела, находят приближенные значения корней с погрешностью, не превосходящей длины отрезка, содержащего корень. Отметим два простых приема отделения действительных корней уравнения - табличный и графический. Первый прием состоит в вычислении таблицы значений функции f(x) в заданных точках xt и использовании следующих теорем

математического анализа:

1. *Если функция y=f(x) непрерывна на отрезке [а,b] и f(a)f(b)<0, то внутри отрезка [a,b] существует по крайней мере один корень уравнения f(x)=0.*
2. *Если функция y=f(x) непрерывна на отрезке [а,b], f(a)f(b) < 0 и f '(x) на интервале (a,b) сохраняет знак, то внутри отрезка [a,b] существует единственный корень уравнения f(x)=0.*

Таким образом, если при некотором k числа f(xk) и f(xk+1) имеют разные знаки, то это означает, что на интервале (xk,xk+1) уравнение имеет по крайней мере один действительный корень нечетной кратности (точнее – нечетное число корней). Выявить по таблице корень четной кратности очень сложно.

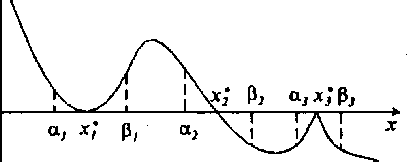


Рис. 1

На рис.1 представлены три наиболее часто встречающиеся ситуации:

а) кратный корень: *f '(x\*)=0, f(a1)\* f(b1) > 0;*

б) простой корень: *f '(x\*)=0, f(a2)\* f(b2) < 0;*

в) вырожденный корень: *f '(x\*)* не существует, f(a3)\* f(b3)>0.

Как видно из рис.1, в первых двух случаях значение корня совпадает с точкой экстремума функции и для нахождения таких корней рекомендуется использовать методы поиска минимума функции.

Для определения числа корней на заданном промежутке используется Теорема Штурма: Если *f(x)* является многочленом и уравнение *f(x)=0* не имеет кратных корней на промежутке [а, b], то число корней этого уравнения, лежащих на таком промежутке, совпадает с числом *N(a) – N(b)*, где функция *N* определяется следующим образом.

Строим ряд Штурма *f0*(x), *f1*(x)*, f2*(x), *…,* *fm*(x), где

*f0*(x) = f(x),

*f1*(x) = f '(x),

*fi*(x) = *остаток от деления fi-2*(x) на *fi-1*(x), взятый с обратным знаком

Функция *N(x)* определяется как число перемен знака в ряде Штурма, если подставить в функции ряда значение *x*

Для отделения корней можно использовать график функции y=f(x). Корнями уравнения являются те значения х, при которых график функции пересекает ось абсцисс. Построение графика функции даже с малой точностью обычно дает представление о расположении и характере корней уравнения (иногда позволяет выявить даже корни четной кратности). Если построение графика функции y=f(x) вызывает затруднение, следует преобразовать исходное уравнение к виду φ1(х)=φ2(х) таким образом, чтобы графики функций у=φ1(х) и у=φ2(х) были достаточно просты. Абсциссы точек пересечения этих графиков и будут корнями уравнения.

Допустим, что искомый корень уравнения отделен, т.е. найден отрезок [а, b], на котором имеется только один корень уравнения. Для вычисления корня с требуемой точностью ε обычно применяют какую-либо итерационную процедуру уточнения корня, строящую числовую последовательность значений xn, сходящуюся к искомому корню уравнения. Начальное приближение х0 выбирают на отрезке [а, b], продолжают вычисления, пока не выполнится неравенство | xn-1 – xn | < ε, и считают, что xn есть корень уравнения, найденный с заданной точностью. Имеется множество различных методов построения таких последовательностей и выбор алгоритма – весьма важный момент при практическом решении задачи. Немалую роль при этом играют такие свойства метода, как простота, надежность, экономичность, важнейшей характеристикой является его скорость сходимости. Последовательность хп, сходящаяся к пределу *x\**, имеет скорость сходимости порядка α, если при . При α=1 сходимость называется линейной, при 1<α<2 – сверхлинейной, при α=2 – квадратичной. С ростом α алгоритм, как правило, усложняется и условия сходимости становятся более жесткими. Рассмотрим наиболее распространенные итерационные методы уточнения корня.

## Метод простых итераций.

Вначале уравнение f(x)=0 преобразуется к эквивалентному уравнению вида х=φ(х). Это можно сделать многими способами, например, положив *φ*(х)=х+♦(x)f(x), где ♦(х) – произвольная непрерывная знакопостоянная функция. Выбираем некоторое начальное приближение х0 и вычисляем дальнейшие приближения по формуле

*хk= φ(хk-1), k=1,2, ...*

Метод простых итераций не всегда обеспечивает сходимость к корню уравнения. Достаточным условием сходимости этого метода является выполнение неравенства на отрезке, содержащем корень и все приближения хп. Метод имеет линейную скорость сходимости и справедливы следующие оценки:

*,*

*.*

Метод имеет простую геометрическую интерпретацию: нахождение корня уравнения f(х)=0 равносильно обнаружению неподвижной точки функции х= φ(х), т.е. точки пересечения графиков функций у= φ(х) и у=х. Если производная φ'(х)<0, то последовательные приближения колеблются около корня, если же производная φ'(х)>0, то последовательные приближения сходятся к корню монотонно.

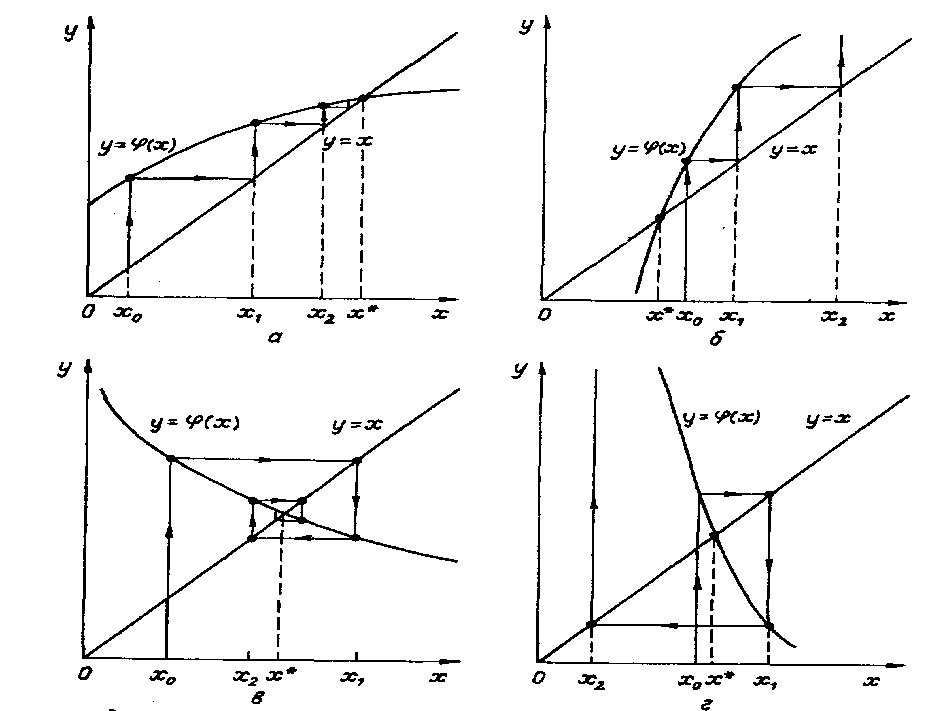


Рис. 2. Метод простых итераций: а - односторонний сходящийся процесс; б - односторонний расходящийся процесс; в - двухсторонний сходящийся процесс; г - двухсторонний расходящийся процесс

Рассмотрим процесс графически (рис. 2). Из графиков видно, что при φ'(x)<0 и при φ'(x)>0 возможны как сходящиеся, так и расходящиеся итерационные процессы. Скорость сходимости зависит от абсолютной величины производной φ(х). Чем меньше |φ'(х)| вблизи корня, тем быстрее сходится процесс.

## Метод хорд.

Пусть дано уравнение f*(x*) *=* 0, a ≤ *x* ≤b, где f*(x)* – дважды непрерывно дифференцируемая функция. Пусть выполняется условие f(a)\*f*(b)*<0 и проведено отделение корней, то есть на данном интервале (a, b) находится один корень уравнения. При этом, не ограничивая общности, можно считать, что f(b)>0.

Пусть функция f выпукла на интервале (a, b) (см. рис. 3).

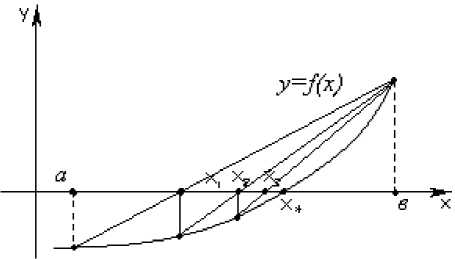


Рис. 3

Заменим график функции хордой (прямой), проходящей через точки *M0(a, f(a))* и *M1(b, f(b))*. Уравнение прямой, проходящей через две заданные точки, можно записать в виде . В нашем случае получим: . Найдем точку пересечения хорды с осью Ox. Полагая у = 0, получаем из предыдущего уравнения: . Теперь возьмем интервал (x1,b) в качестве исходного и повторим вышеописанную процедуру (см. рис. 3). Получим . Продолжим процесс. Каждое последующее приближение вычисляется по рекуррентной формуле (3.1)

,

.

Если же функция вогнута (см. рис. 4),

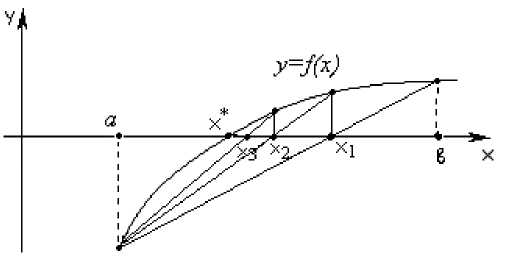


Рис. 4

уравнение прямой соединяющей точки *M0(a, f(a))* и *M1(b, f(b))* запишем в виде . Найдем точку пересечения хорды с осью Ox: . Теперь возьмем интервал (a, x1) в качестве исходного и найдем точки пересечения хорды, соединяющей точки (a, f(a)) и (x1, f(x1)), с осью абсцисс (см. рис. 4). Получим . Повторяя данную процедуру, получаем рекуррентную формулу:

(3.2)

,

.

Описанный выше метод построения рекуррентных последовательностей (3.1) и (3.2) называется методом хорд. Для использования метода хорд нужно было бы предварительно найти точки перегиба и выделить участки, на которых функция не меняет характер выпуклости. Однако на практике поступают проще: в случае для построения рекуррентной последовательности применяются формулы (3.1), а в случае, когда , применяют формулы (3.2).

## Метод Ньютона (касательных).

Для начала вычислений требуется задание одного начального приближения x0, последующие приближения вычисляются по формуле

Метод имеет квадратичную скорость сходимости для простого корня, но очень чувствителен к выбору начального приближения. При произвольном начальном приближении итерации сходятся, если всюду , в противном случае сходимость будет только при x0, достаточно близком к корню. Существует несколько достаточных условий сходимости. Если производные и сохраняют знак в окрестности корня, рекомендуется выбирать x0 так, чтобы . Если, кроме этого, для отрезка [a,b], содержащего корень, выполняются условия то метод сходится для любых a ≤ x0 ≤ b.

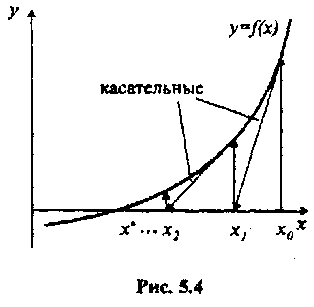


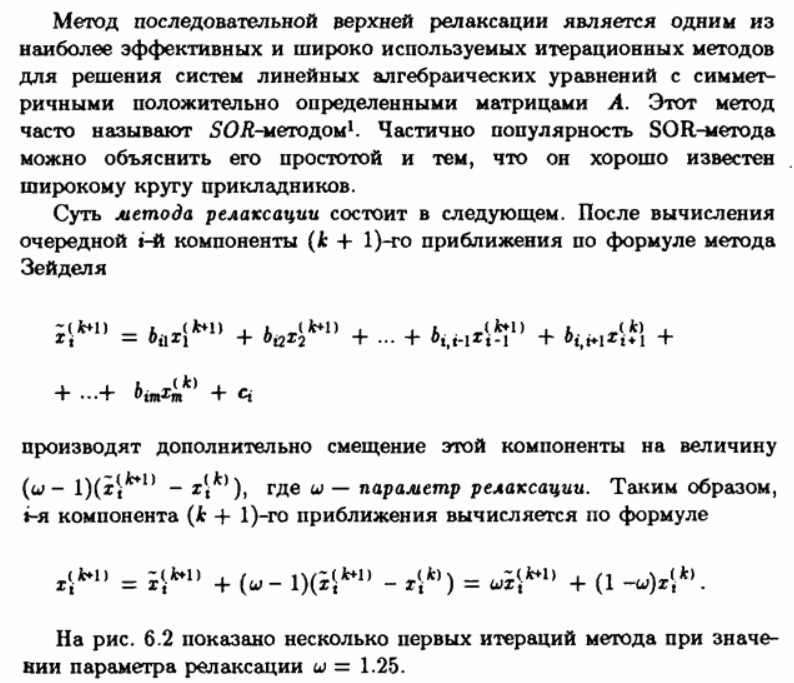
Рис. 5

Метод Ньютона получил также второе название метод касательных благодаря геометрической иллюстрации его сходимости, представленной на рис. 5.

Метод Ньютона позволяет находить как простые, так и кратные корни. Основной его недостаток – малая область сходимости и необходимость вычисления производной.

## **Метод релаксации:**

Метод релаксаций применяется при решении множества близких алгебраических систем линейных уравнений.



На первом этапе проводится решение одной из систем с различными значениями итерационного параметраи из анализа скорости сходимости итерационного процесса выбирается оптимальное значение этого параметра. Затем все остальные системы решаются с выбранным значением .

Еще одно достоинство итерационного метода верхних релаксаций состоит в том, что при его реализации на ЭВМ алгоритм вычислений имеет простой вид и позволяет использовать всего один массив для неизвестного вектора

Скорость сходимости зависит от параметра релаксации.

## **Метод половинного деления**

Перед применением метода для поиска корней функции необходимо отделить корни одним из известных способов, например, графическим методом. Отделение корней необходимо в случае, если неизвестно на каком отрезке нужно искать корень.

Будем считать, что корень t функции f(x)=0 отделён на отрезке [a,b]. Задача заключается в том, чтобы найти и уточнить этот корень методом половинного деления. Другими словами, требуется найти приближённое значение корня с заданной точностью \eps.

Пусть функция f непрерывна на отрезке [a,b],

f(a)\cdot f(b)<0, \; \eps=0,01 и t\in[a,b] - единственный корень уравнения f(x)=0, \; a\le t\le b.

(Мы не рассматриваем случай, когда корней на отрезке [a,b] несколько, то есть более одного. В качестве \eps можно взять и другое достаточно малое положительное число, например, 0,001.)

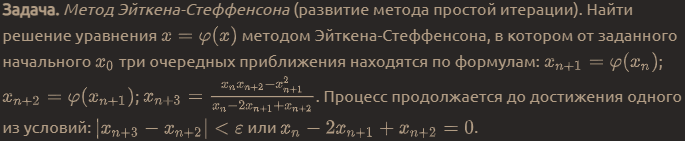
Поделим отрезок [a,b] пополам. Получим точку c= \frac {a+b}{2}, \; a<c<b и два отрезка [a,c], \; [c,b].

* Если f(c)=0, то корень t найден (t=c).
* Если нет, то из двух полученных отрезков [a,c] и [c,b] надо выбрать один [a_1;b_1] такой, что f(a_1)\cdot f(b_1)<0, то есть
  + [a_1;b_1] = [a,c], если f(a)\cdot f(c)<0 или
  + [a_1;b_1] = [c,b], если f(c)\cdot f(b)<0.

Новый отрезок [a_1;b_1] делим пополам. Получаем середину этого отрезка c_1=\frac {a_1+b_1}{2} и так далее.

Для того, чтобы найти приближённое значение корня с точностью до  \eps >0, необходимо остановить процесс половинного деления на таком шаге n, на котором |b_n-c_n|<\eps и вычислить x=\frac {a_n+b_n}{2}. Тогда можно взять t\approx x.

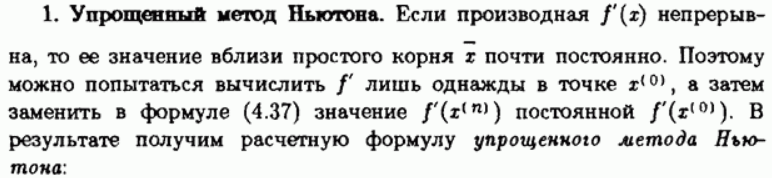
## **Метод Эйткена-Стеффенсона**

****

Скорость сходимости метода Эйткина-Стеффенсена превышает скорость сходимости метода Ньютона.

## **Модификации метода Ньютона**

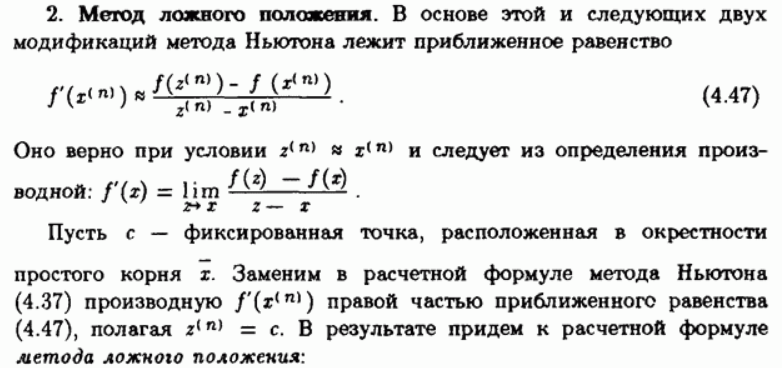
### **Упрощенный метод Ньютона**

****

**C:\Users\Dima\Downloads\Untitled.png**

Линейная сходимость

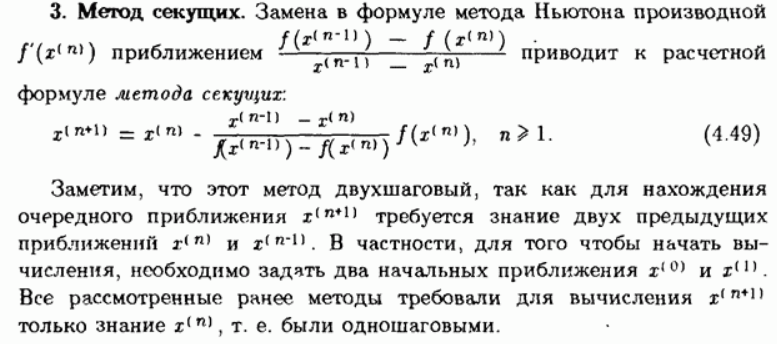
### **Метод ложного положения**

****

**C:\Users\Dima\Downloads\Untitled.png**

Линейная сходимость

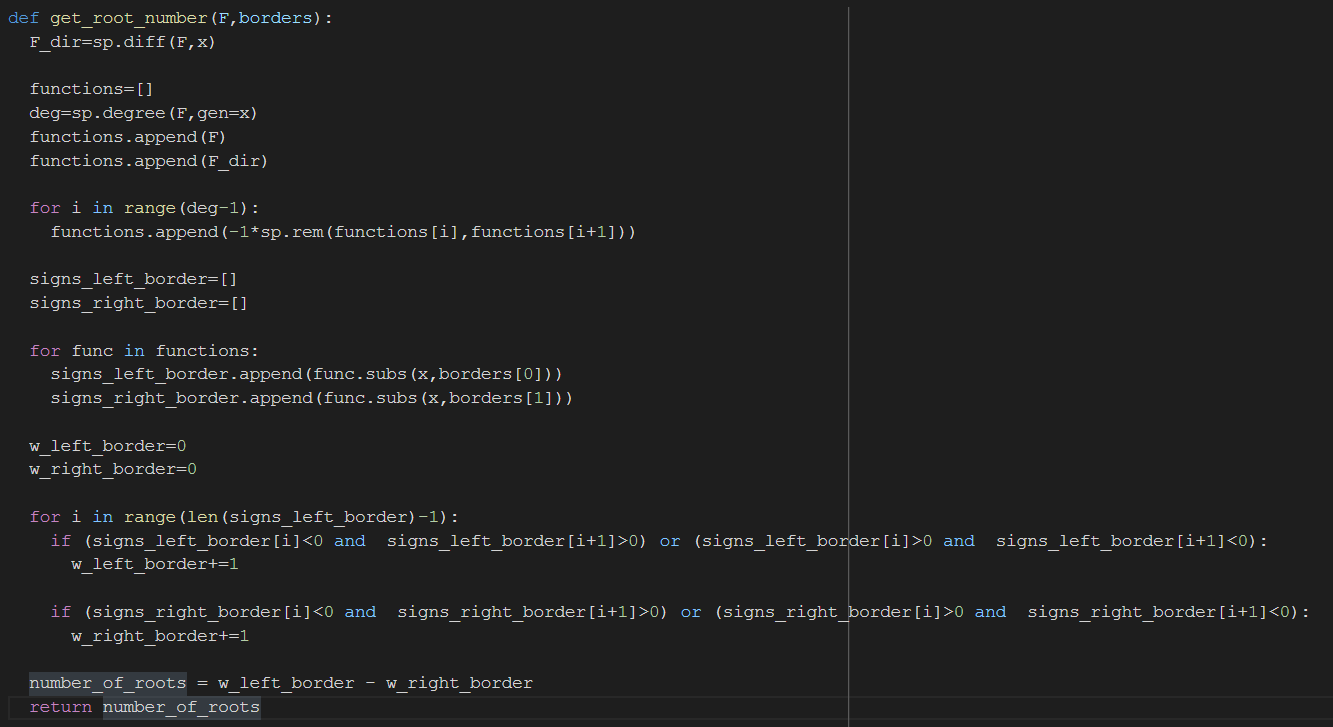
### **Метод секущих**

****

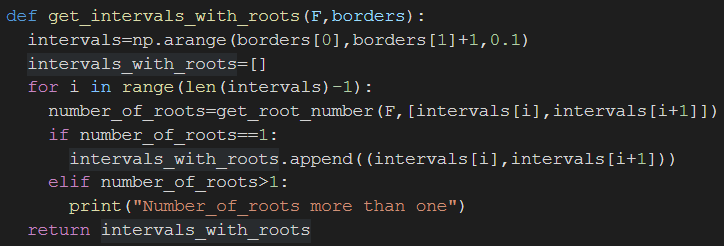
# Программная реализация

## Код:

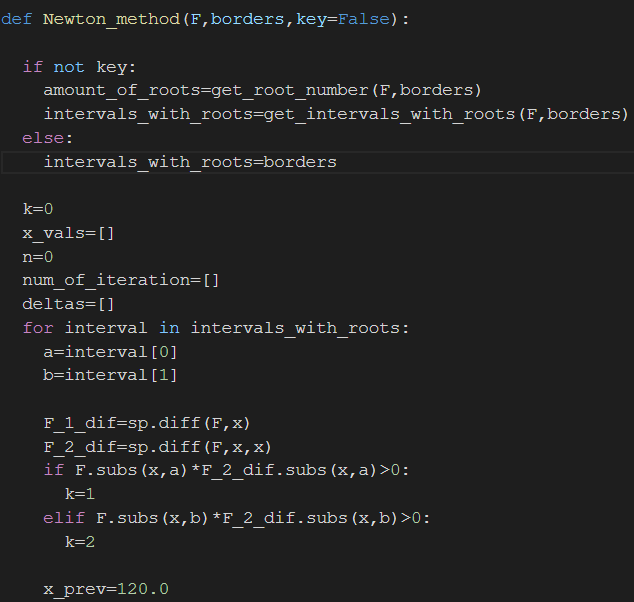
Функция get\_root\_number находит количество корней на заданном интервале методом Штурма.

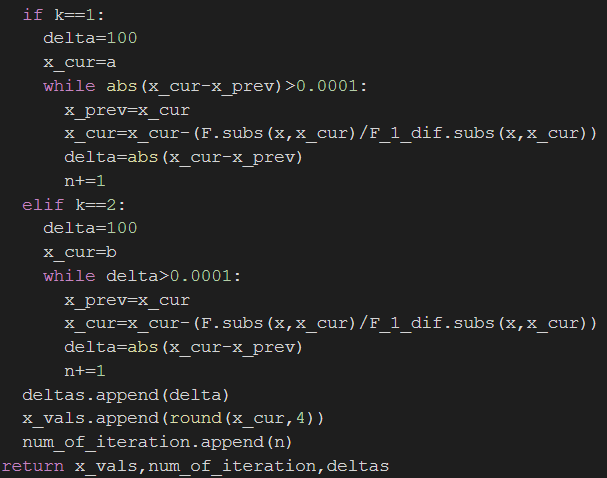


Функция get\_intervals\_with\_roots отделяет все корни лежащие на одном отрезке.

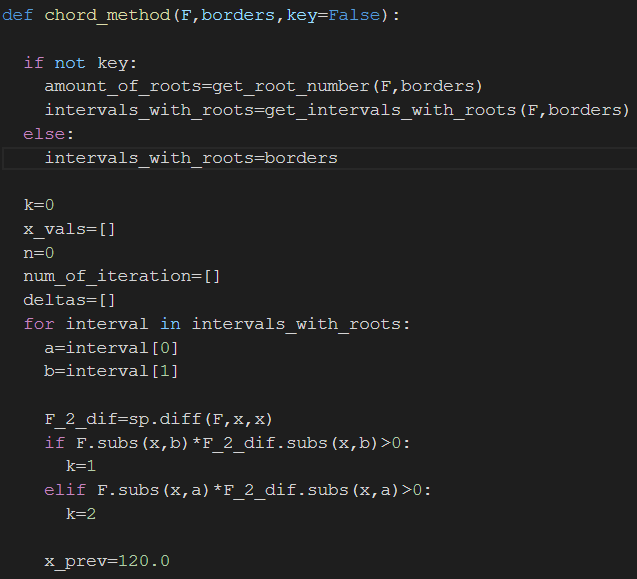


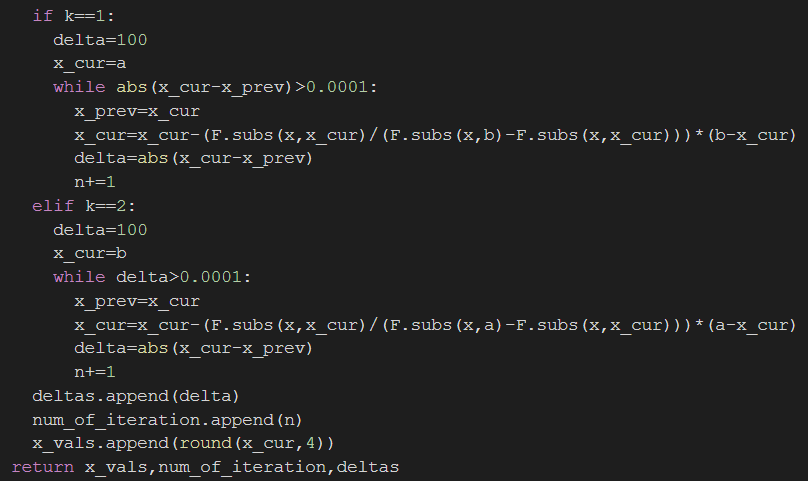
Функция Newton\_method находит корни нелинейного уравнения методом Ньютона.

 \_

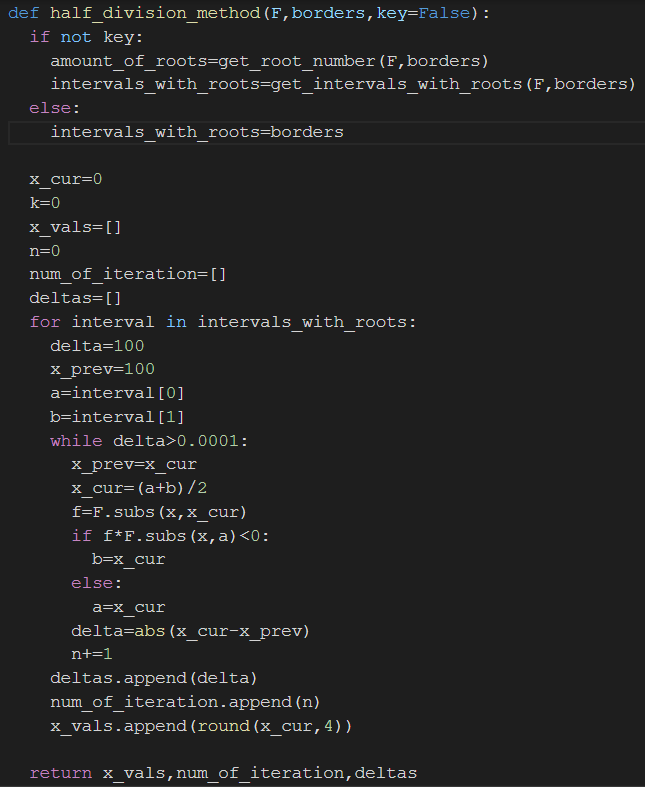


Функция chord\_method находит корни нелинейного уравнения методом Хорд.

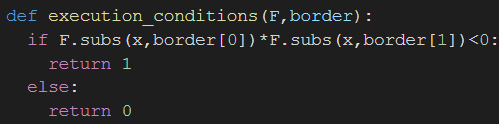




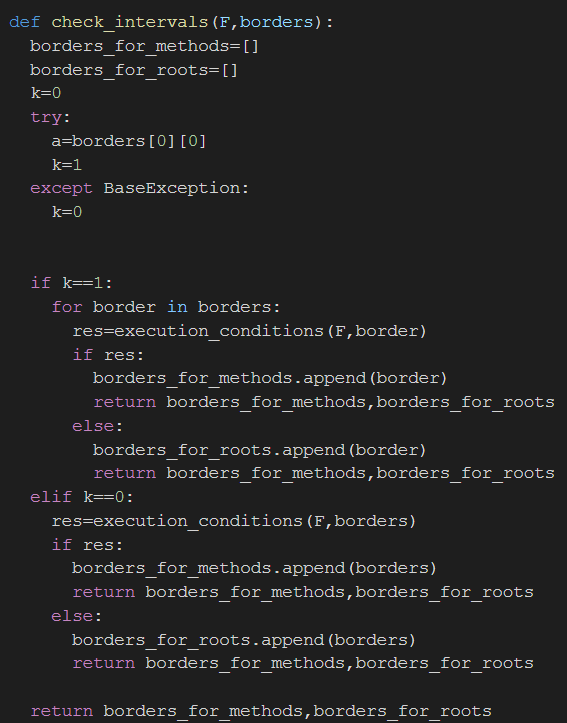
Функция half\_division\_method находит корни нелинейного уравнения методом половинного деления.



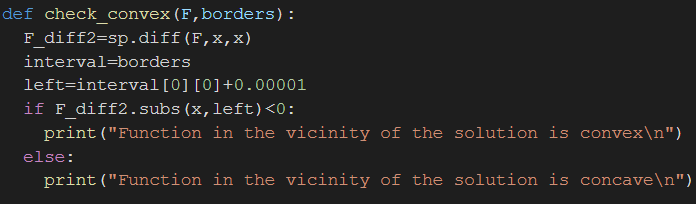
Функция execution\_condition проверяет выполнена ли теорема Больцмана-Коши



Функция check\_intervals определяет типы корней.



Функция check\_convex определяет выпуклость или впуклость функции в окрестности точки.

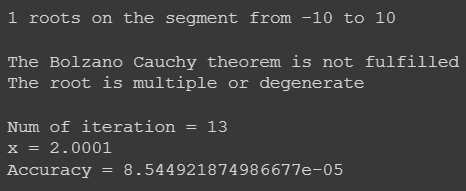


## Тестовые примеры

1.

C:\Users\Dima\Downloads\Untitled.png

Мой ответ



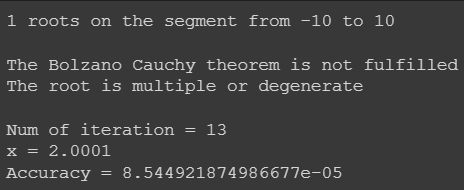
Правильный ответ.

C:\Users\Dima\Downloads\Untitled.png

2.

C:\Users\Dima\Downloads\Untitled.png

Мой ответ:



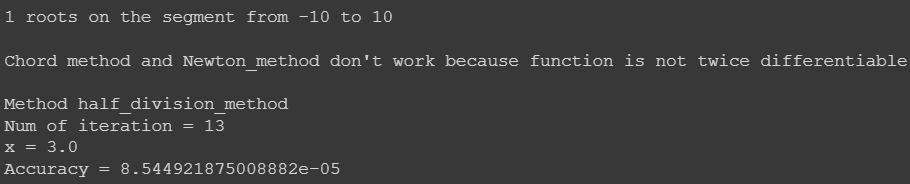
Правильный ответ.

C:\Users\Dima\Downloads\Untitled.png

3.

C:\Users\Dima\Downloads\Untitled.png

Мой ответ:



Правильный ответ.

C:\Users\Dima\Downloads\Untitled.png

4.

C:\Users\Dima\Downloads\Untitled.png

Мой ответ:

C:\Users\Dima\Downloads\Untitled.png

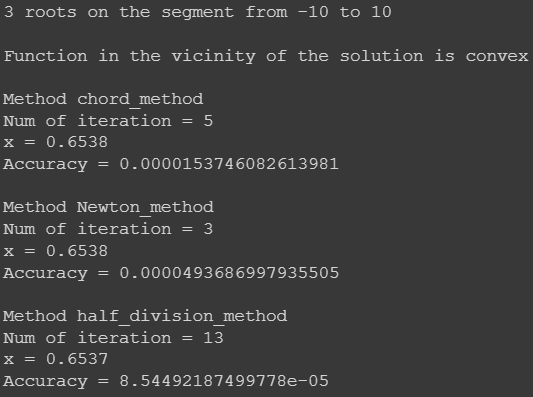
Правильный ответ.

Нет действительных корней.

Вариант 8:

C:\Users\Dima\Downloads\Untitled.png

Мой ответ:



Правильный ответ.

C:\Users\Dima\Downloads\Untitled.png

# Вывод:

После проделанной работы я могу заключить, что среди трех рассматриваемых методов быстрее всего сходится к ответу с заданной точностью метод Ньютона, так как он имеет квадратичную скорость сходимости, потом идет метод хорд, он сходится со скоростью геометрической прогрессии, хуже всего метод половинного деления он сходится с линейной скоростью. Однако метод хорд и метод Ньютона имеет критерии сходимости и если они не выполняются, то сходимость будет только при х близком к решению, метод же половинного деления хоть и долгий, но он найдет решение в любом случае если на отрезеке есть корень.