ЭКЗ ПО ПРОГЕ

**Методы уточнения корней**

Корень называется простым, если f’(x) ≠ 0, иначе – кратный.

Локализация:

1. **Метод Ньютона (касательных)**

Основная идея метода заключается в следующем: задаётся начальное приближение вблизи предположительного корня, после чего строится касательная к графику исследуемой функции в точке приближения, для которой находится пересечение с осью абсцисс. Эта точка берётся в качестве следующего приближения. И так далее, пока не будет достигнута необходимая точность.

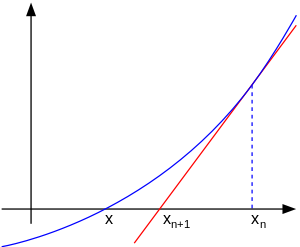
Метод Ньютона (метод касательных) применяется в том случае, если уравнение f(x) = 0 имеет корень x на [a,b], и выполняются условия:

1) функция f(x) определена и непрерывна при /var/folders/31/4dy5ynqd6hb30snv61jqdp700000gn/T/com.microsoft.Word/WebArchiveCopyPasteTempFiles/12.png

2) f(a)·f(b) < 0 (функция принимает значения разных знаков на концах отрезка [a,b])

3) производные f'(x) и f''(x) сохраняют знак на отрезке [a;b] (т.е. функция f(x) либо возрастает, либо убывает на отрезке [a;b], сохраняя при этом направление выпуклости)

4) /var/folders/31/4dy5ynqd6hb30snv61jqdp700000gn/T/com.microsoft.Word/WebArchiveCopyPasteTempFiles/13.png.



Формула: xn+1 = xn – f(xn)/f’(xn)

Критерий окончания: | xn+1 - xn | < e

**Плюсы:**

- Сходимость метода касательных квадратичная, порядок сходимости равен 2. Таким образом, сходимость метода касательных Ньютона очень быстрая.

**Минусы:**

- Необходимость вычисления производных на каждом шаге.

- Если начальное приближение недостаточно близко к решению, то метод может не сойтись.

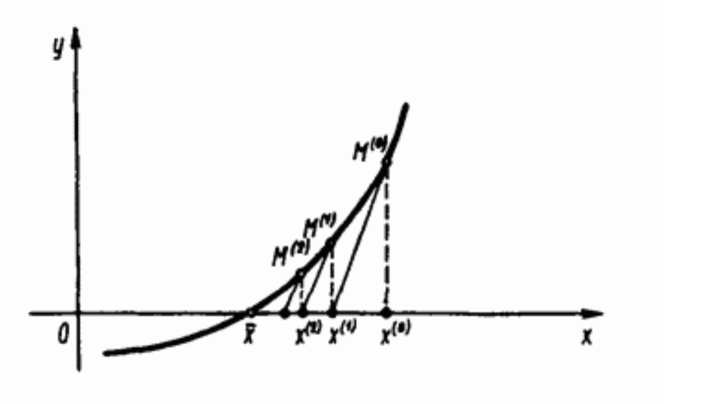
**-** Если корень x является корнем второй кратности и выше, то порядок сходимости падает и становится линейным.

- Много условий для сходимости

1. **Упрощенный метод Ньютона**

Если производная непрерывна, то ее значение вблизи простого корня практически постоянно. Поэтому можем вычислить производную один раз для x0 и подставить ее в формулу классического метода Ньютона.

(Проводим первую касательную, а все последующие – параллельно.)



Формула: xn+1 = xn – f(xn)/f’(x0), где x0 – некоторое начальное приближение к корню

Критерий окончания: | xn+1 - xn | < e

**Плюсы:**

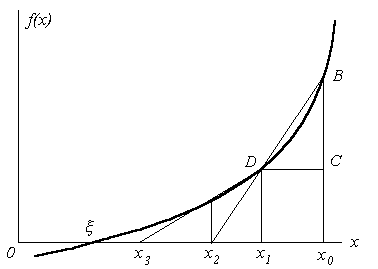
- простота реализации

- возможность обобщения на системы уранвения

**Минусы:**

- более медленную по сравнению с классическим методом Ньютона сходимость (она линейная).(медленнее)

1. **Метод секущих**



В этом алгоритме начинают с двумя исходными числами x1 и х2. На каждом шаге xn+1 получают из xn и xn-1 как единственный нуль линейной функции, принимающей значения f(xn) в xn и f(xn-1) в xn-1. Эта линейная функция представляет секущую к кривой у = f(х), проходящую через ее точки с абсциссами xn и xn-1 - отсюда название метод секущих.

Формула:

xn+1 = xn – (xn-1 - xn)/(f(xn-1)-f(xn)\*f(xn)

**Минусы:**

- Скорость сходимости всегда будет линейной

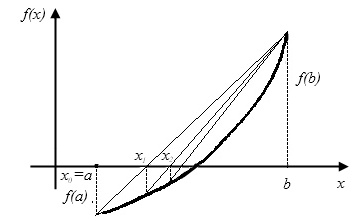
**Плюсы:**

- Больше возможность сходимости

**4. Метод Хорд**

Отличие от метода секущих состоит в том, что если в методе секущих в качестве точек следующей итерации выбираются последние рассчитанные точки, то в методе хорд выбираются те точки, в которых функция имеет разный знак, и соответственно, выбранный интервал содержит корень.

Рассматриваемый метод так же, как и метод половинного деления, предназначен Очередное приближение в отличие от метода половинного деления берем не в середине отрезка, а в точкеx0, где пересекает ось абсцисс прямая линия (хорда), проведенная через точки aи b.



Условия:

1. На концах отрезка [a,b] функция имела значения разных знаков

2. На [a,b] f’(x) и f’’(x) сохраняли постоянный знак

Формула: xn+1 = xn – (f(xn)/(f(xn) – f(a)))\*(xn - a)

xn+1 = xn – (f(xn)/(f(b) – f(xn)))\*(b - xn)

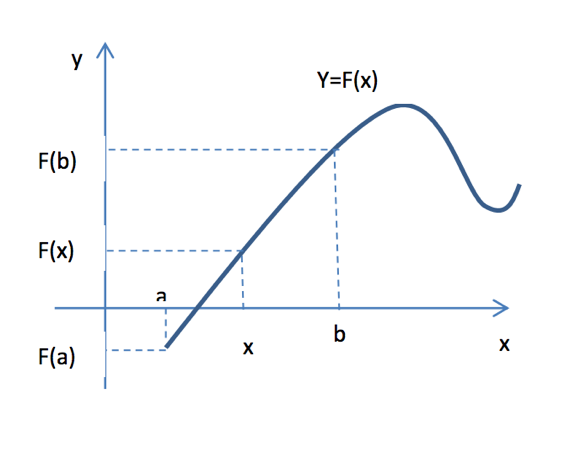
**Плюсы:**

Порядок сходимости больше линейного, но не квадратичен, как у Метода Ньютона (но все равно быстрый).

**Минусы:**

Много условий для сходимости метода

1. **Метод половинного деления**



Метод половинного деления один из методов решения нелинейных уравнений и основан на последовательном сужении интервала, содержащего единственный корень уравнения f(x)=0 до того времени, пока не будет достигнута заданная точность ɛ. Пусть задан отрезок [а,b], содержащий один корень уравнения. Предварительно необходимо определить области локализации корней данного уравнения. Если на отрезке [а,b] содержится более одного корня, то метод не работает.

**Плюсы:**

- Прост в реализации

**Минусы:**

**- Сходимость** линейна

1. **Комбинированный метод хорд и касательных**

Суть комбинированного метода состоит в разбиении отрезка [a,b] (при условии f(a)f(b)<0) на три отрезка с помощью хорды и касательной и выборе нового отрезка от точки пересечения хорды с осью абсцисс до точки пересечения касательной с осью абсцисс, на котором функция меняет знак и содержит решение.

Поэтому их часто применяют в сочетании друг с другом, и уточнение корня происходит быстрее.

Формула:

1. Если f(a)\*f’’(a)<0, то a = a-f(a)\*(a-b)/(f(a)-f(b))

Если f(a)\*f’’(a)>0, то a = a-f(a)/f’(a)

2. Если f(b)\*f’’(b)<0, то b = b-f(b)\*(b-a)/(f(b)-f(a))

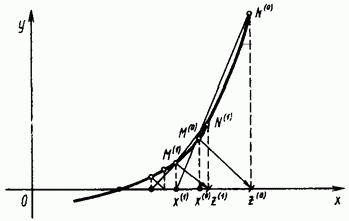
Если f(b)\*f’’(b)>0, то b = b-f(b)/f’(b)

X = (a+b)/2, вывод x

Плюсы:

- Объединяя метод хорд и метод Ньютона, можно ускорить сходимость итерационного процесса поиска корня.

1. **Метод Стеффенсена**



Формула: xn+1 = xn – f(xn)/(f(xn+f(xn))-f(xn))\*f(xn)

Плюсы:

- одношаговый

- не требует вычисления производной

- сходится квадратично

Минусы:

- метод Стеффенсена уступает методу секущих, поскольку требует большей вычислительной работы для достижения той же точности

1. **Метод итераций**

Нахождение алгоритма поиска по известному приближению (приближенному значению) искомой величины следующего, более точного приближения. Применяется в случае, когда последовательность приближений по указанному алгоритму сходится.

Характер сходимости и сам факт сходимости метода зависит от выбора начального приближения корня x0.

Для его применения уравнение f(x)=0 представляется в виде x = g(x), где g(x) = x+f(x)

Формула: xn+1 = g(xn).

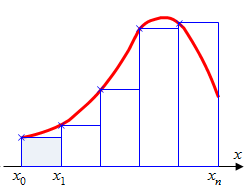
Характер сходимости и сам факт сходимости метода зависит от выбора начального приближения корня x0.

Сравнение методов:

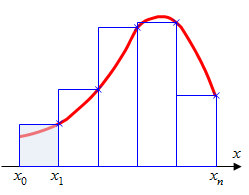
1. Самый медленный и надежный – метод половинного деления. Также он самый простой
2. Метод Ньютона эффективен, график в окрестности корня имеет крутизну. Высокая скорость, но также сходимость зависит от вида функции
3. Метод итераций дает возможность «угадывать» новые значения корня на любом шаге.

Интегралы:

Метод левых прямоугольников:

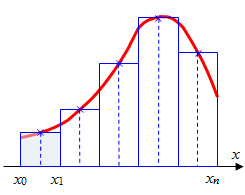


def left(f, a, b, n):  
 h = (b - a)/n  
 s = 0  
 for i in range(n):  
 s += f(a + i\*h)  
 s \*= h  
 return s

Метод правых прямоугольников:  


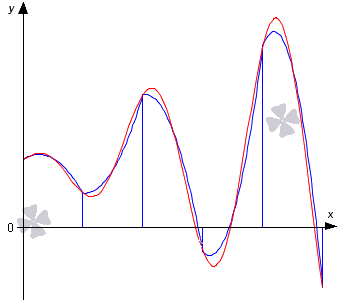
def right(f, a, b, n):  
 h = (b - a)/n  
 s = 0  
 for i in range(1, n + 1):  
 s += f(a + i\*h)  
 s \*= h  
 return s

Метод серединных прямоугольников:



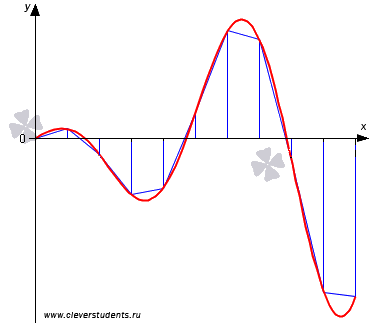
def middle(f, a, b, n):  
 h = (b - a)/n  
 kv = 0  
 for i in range(n):  
 kv += f(a + (i + 0.5)\*h)  
 kv \*= h  
 return s

Метод симпсона:



def simpson(f, a, b, n):  
 if n % 2:  
 return '---'  
 else:  
 h = (b - a)/n  
 s = 0  
 for i in range(n + 1):  
 if not i % n:  
 k = 1  
 elif i % 2:  
 k = 4  
 else:  
 k = 2  
 s += k\*f(a + i\*h)  
 s \*= h/3  
 return s

Метод трапеций:



def trap(f, a, b, n):  
 h = (b - a)/n  
 s = 0  
 for i in range(n + 1):  
 if not i % n:  
 k = 1  
 else:  
 k = 2  
 s += k\*f(a + i\*h)  
 s \*= h/2  
 return s

Метод 3/8:

def cubic(f, a, b, n):  
 if n % 3:  
 return '---'  
 else:  
 h = (b - a)/n  
 s = 0  
 for i in range(n + 1):  
 if not i % n:  
 k = 1  
 elif i % 3:  
 k = 3  
 else:  
 k = 2  
 s += k\*f(a + i\*h)  
 s \*= h\*3/8  
 return s