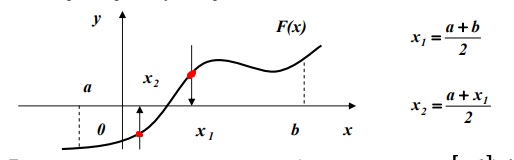
**ЧИСЛЕННЫЕ РЕШЕНИЯ НЕЛИНЕЙНЫХ УРАВНЕНИЙ**

Метод ДИХОТОМИИ

Штож, начнем с метода дихотомии

Начнем самого начала, если на границах некого отрезка функция F(x) имеет разные по знаку ответы, то это значит что функция хотя бы раз да переходит через ось 0Х и нам нужно найти эту точку прохождения через ось. Делать мы это будем постепенно деля отрезок, сохраняя при этом различие знаков на новых получившихся отрезках, тем самым приближаясь к самой точке. Перейдем к графику:



Что же мы тут делаем, делим отрезок [a,b] пополам и получаем отрезки [a,x1] & [x1,b]. Теперь мы смотрим на каком из этих отрезках сохраняется различие знака:

F(x1) > 0 & F(b) > 0 знак на концах отрезков везде положительный, значит не подходит.  
F(a) < 0 & F(x1) > 0 знаки различаются, значит корень который мы ищем находиться ТУТЬ.

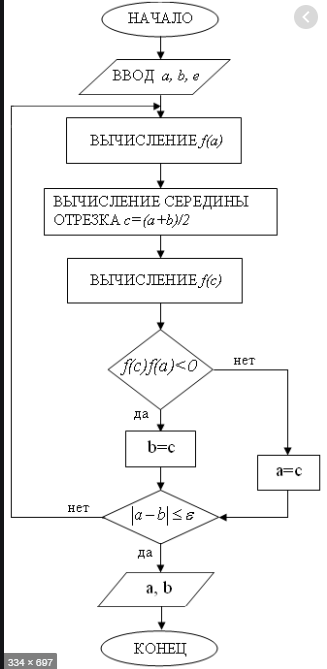
Продолжаем деление но уже отрезка [a,x1], и получаем два новых отрезка [a,x2] & [x2,x1].

Теперь мы смотрим на каком из этих отрезках сохраняется различие знака:

F(a) < 0 & F(x2) < 0 знак на концах отрезков везде отрицательный, значит не подходит.

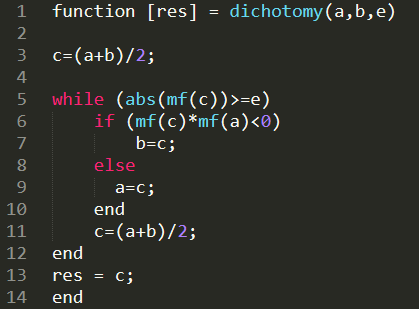
F(x2) < 0 & F(x1) > 0 знаки различаются, значит корень который мы ищем находиться ТУТЬ.

И так мы делаем до тех пор, пока не выполнится один из или сразу оба критерия завершения процедура.

 e - погрешность  
 xk – корень удовлетворяющий условию  
 n – кол-во итераций ну или делений, кому как удобно.  
  
1 критерий: 2 критерий:   
| F(xk) | < e (b – a ) / (2^n) < e

Ну а теперь перейдем к блоксхеме:

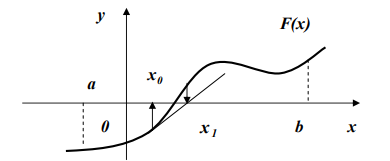
и к коду



Метод НЬЮТОНА

ШТОШ, тут не сильно сложнее, данный метод не требует чтобы на концах отрезка функция принимала разные значения, в основе этого метода лежит такая дичь как разложение функции в ряд Телора, но как бы закрывая глаза на члены со второй и более степенями, получая вот етто:

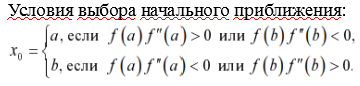
геометрический смысл таков, из первой точки прближения X0 мы проводим касательную и на месте пересечения касательной и 0X мы находим следующую точку приближения

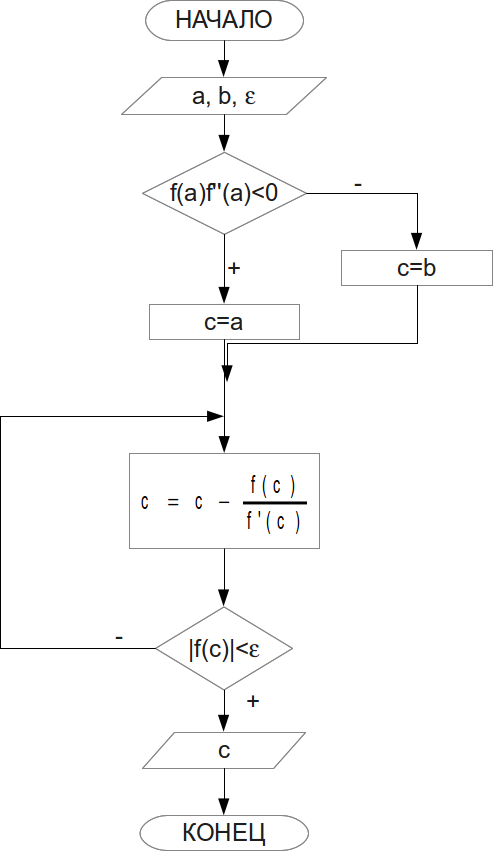


и так до выполнения критерия остановки:

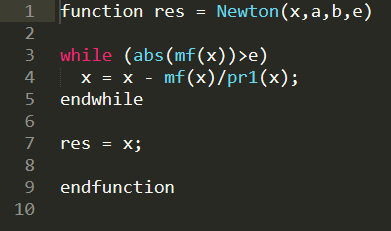
| F(xk) | < e

так же можно сделать выбор начальноего приближения, но я этого не делал)



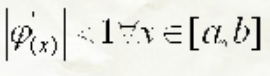


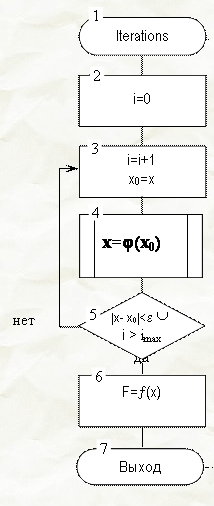
блок схема и код:

pr1 – первая производная от функции

Метод ПРОСТОЙ ИТЕРАЦИИ

ШТОШ, это самый непонятный для меня метод, хоть и название говорит что он простой ¯\\_(ツ)\_/¯

Кароч, нам дана функция F(x), нам нужно привести ее к эквивалентному виду и называть ФИ, при условии что

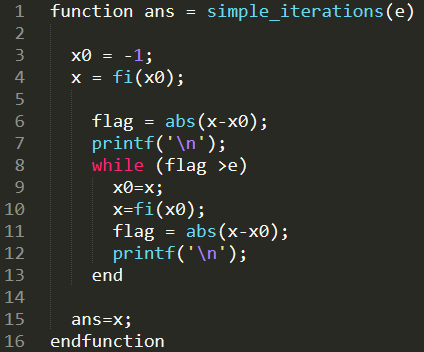


это нужно просто осознать и смириться.

Что же дальше?  
Просто по блок схеме пишем этот код и все

ну по переменным  
e – погрешность

X0 – начальное приближение



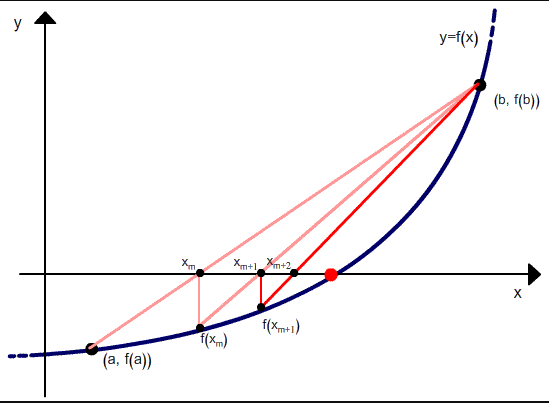


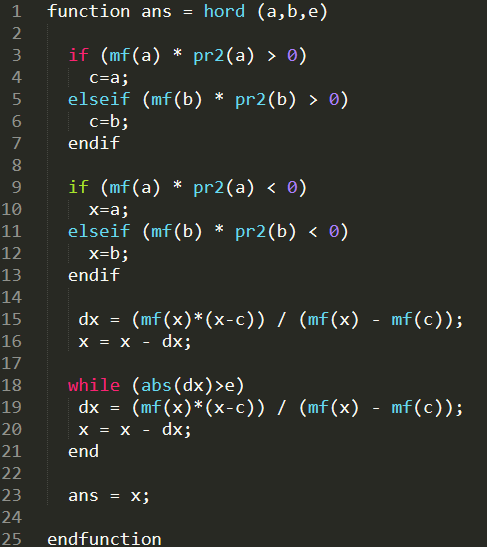
НЕ ПОНЯТНО НО ИНТЕРЕСНО….

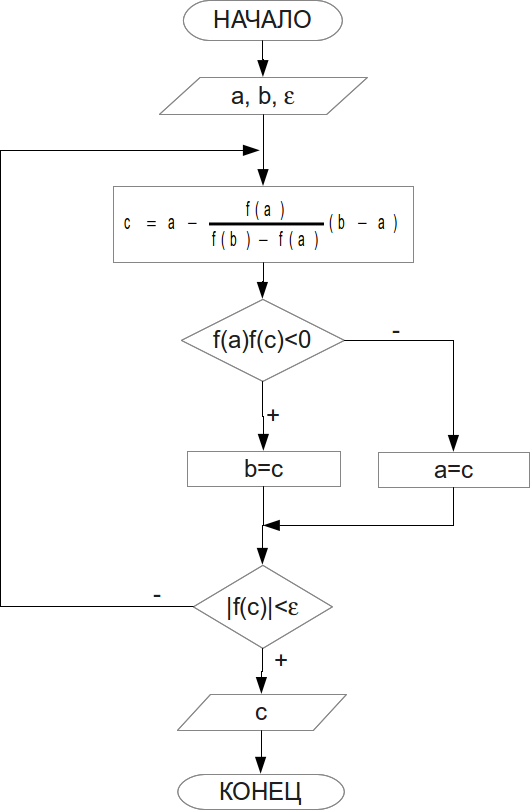
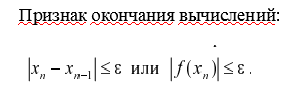
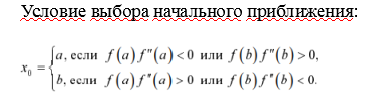
Метод ХОРД

Ничего сложного, дан отрезок, берем две точки x1 & x2, теперь чертим линию между ними и на пересечении линии и оси 0X мы получаем новую точку X3, дальше находим значение функции в это точке и получаем точку C1 ну и теперь выбираем какая из точек x1 & x2 имеет противоположный знак, и делаем процедуру заново, постепенно находя новые С и приближаясь к корню. И делаем мы все это до выполнения критерия завершения.

Ну и куда же без рисууууууночка)



Теперь к алгоритму, коду, и схеме

1 – выбираем приближение 2 — считаем по формуле до признака окончания вычислений