Выполнил:

Студент 1 курса

Группы 18ПМИ-2

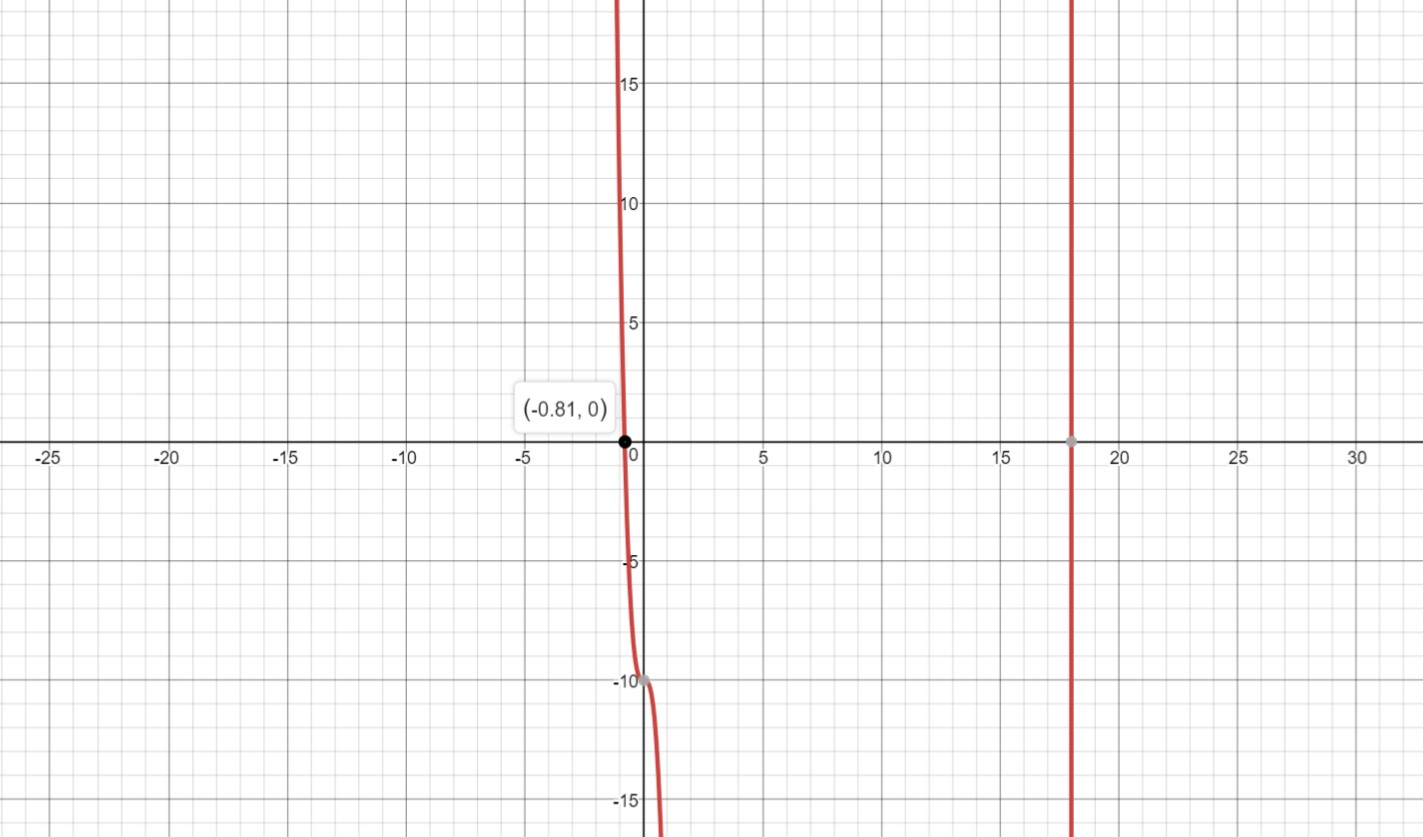
Мусатов Антон

**Отчет о лабораторной работе номер 1  
«Численные методы решения нелинейного уравнения с одним неизвестным»  
Вариант 18**

1. Цель работы: отделить корни уравнений *x4 -18x3-10=0* и *arctg(x2+1/x) = x* на отрезке [a,b] и уточнить корни этих уравнений методами половинного деления, хорд, Ньютона и простой итерации.

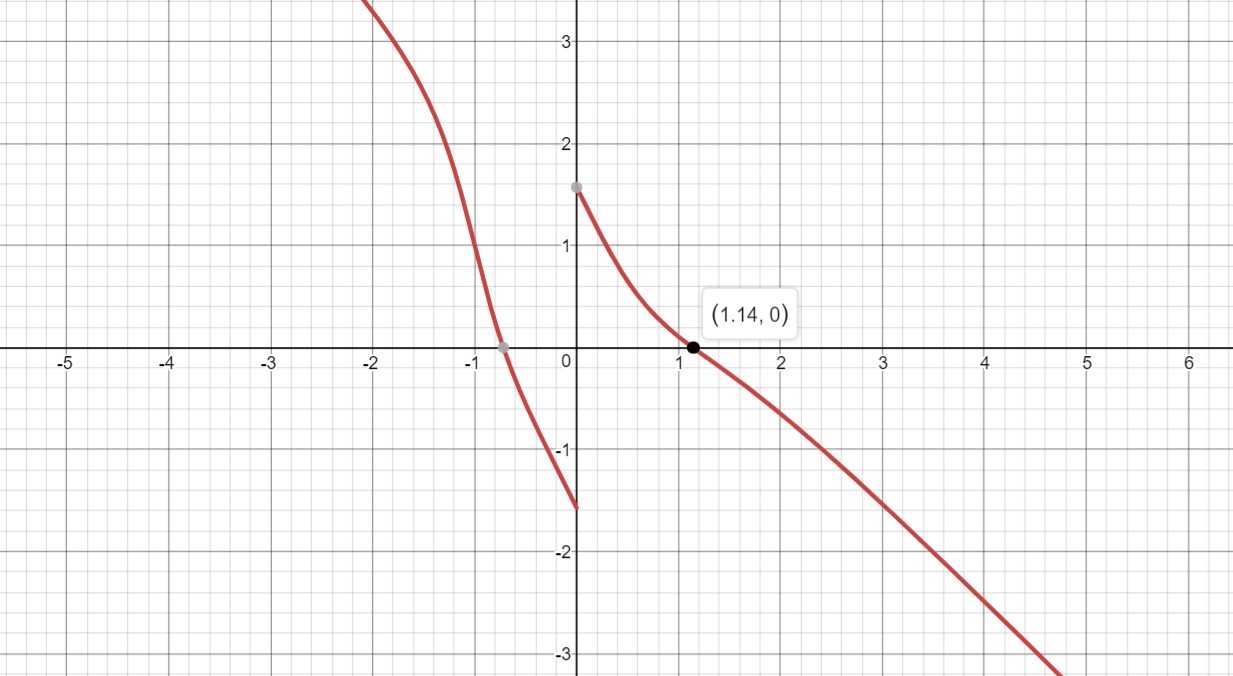
*y = x4 -18x3-10*

В ходе работы уточню корень *x = - 0,81* на отрезке *[-1.5;0]*



*y = arctg(x2+1/x) – x*

В ходе работы уточню корень *x = 1,14* на отрезке *[0.5;2]*



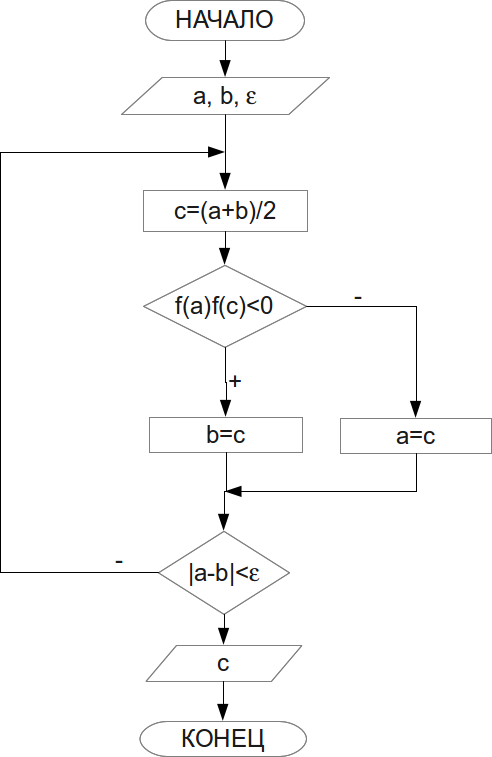
2. 1. Метод половинного деления

**Алгоритм:**

Пусть был выбран интервал изоляции [a, b]. Примем за первое приближение корня точку c, которая является серединой отрезка [a, b]. Далее, будем действовать по следующему алгоритму:

1. Находим точку ;
2. Находим значение ;
3. Если , то корень лежит на интервале [a, c], иначе корень лежит на интервале [c, b];
4. Если величина интервала меньше либо равна , то найден корень с точностью , иначе возвращаемся к п.1.

**Блок-схема:**



2.2. Метод хорд

**Алгоритм:** Этот метод отличается от метода дихотомии тем, что очередное приближение берём не в середине отрезка, а в точке пересечения с осью Xпрямой, соединяющей точки (a, f (a)) и (b, f (b)).

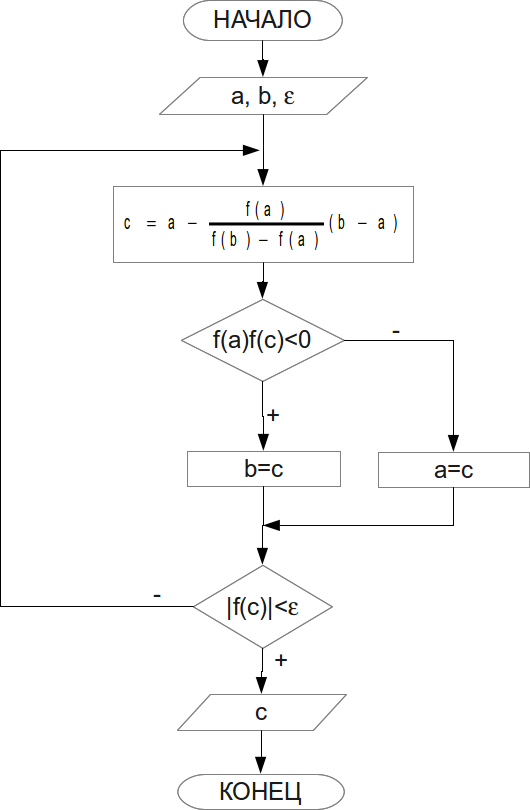
Запишем уравнение прямой, проходящей через точки с координатами (a, f (a)) и (b, f (b)) :

|  |  |
| --- | --- |
| \frac{y-f(a)}{f(b)-f(a)}=\frac{x-a}{b-a},\ \ \  y=\frac{f(b)-f(a)}{b-a}\cdot (x-a)+f(a) |  |

Прямая, заданная уравнением (4.2), пересекает ось Xпри условии y = 0.

Найдём точку пересечения хорды с осью X: y=\frac{f(b)-f(a)}{b-a}\cdot (x-a)+f(a),\ \  x=a-\frac{f(a)\cdot (b-a)}{f(b)-f(a)}.итак, c=a-\frac{f(a)}{f(b)-f(a)}(b-a).

**Блок-схема:**



2.3 Метод Ньютона

**Алгоритм:**

Метод Ньютона (метод касательных) применяется в том случае, если уравнение *f(x) = 0* имеет корень http://statistica.ru/upload/medialibrary/chisl-methods-resh-ur/11.png, и выполняются условия:

1. функция *y= f(x)* определена и непрерывна при http://statistica.ru/upload/medialibrary/chisl-methods-resh-ur/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. http://statistica.ru/upload/medialibrary/chisl-methods-resh-ur/13.png.

Основная идея метода заключается в следующем: на отрезке [*a;b*] выбирается такое число *x0,* при котором *f(x0)* имеет тот же знак, что и *f''(x0),* т. е. выполняется условие *f(x0)·f''(x) > 0*. Таким образом, выбирается точка с абсциссой *x0*, в которой касательная к кривой *y=f(x)* на отрезке [*a;b*] пересекает ось *Ox*. За точку *x0* сначала удобно выбирать один из концов отрезка.

Если http://statistica.ru/upload/medialibrary/chisl-methods-resh-ur/image022.png — некоторое приближение к корню http://statistica.ru/upload/medialibrary/chisl-methods-resh-ur/image024.pngуравнения http://statistica.ru/upload/medialibrary/chisl-methods-resh-ur/image026.png, то следующее приближение определяется как корень касательной к функции http://statistica.ru/upload/medialibrary/chisl-methods-resh-ur/image028.png, проведенной в точке http://statistica.ru/upload/medialibrary/chisl-methods-resh-ur/image022_0000.png.

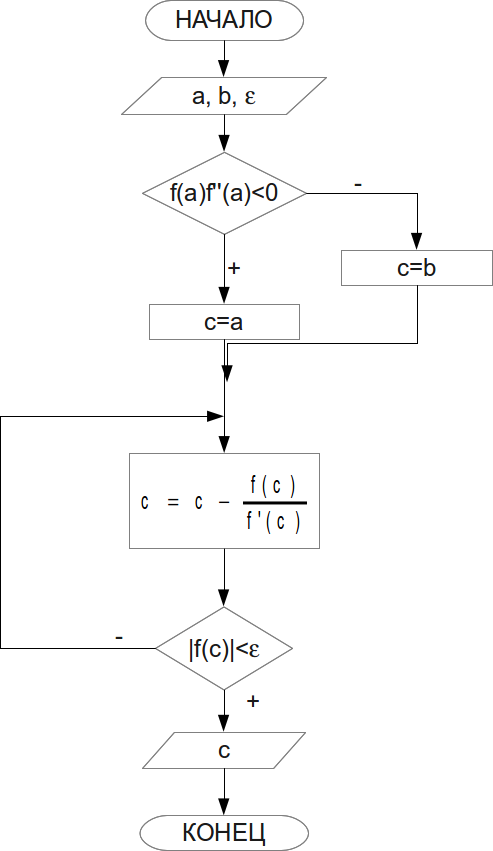
Уравнение касательной к функции http://statistica.ru/upload/medialibrary/chisl-methods-resh-ur/image030.pngв точке http://statistica.ru/upload/medialibrary/chisl-methods-resh-ur/image032.pngимеет вид:

http://statistica.ru/upload/medialibrary/chisl-methods-resh-ur/image034.png

В уравнении касательной положим http://statistica.ru/upload/medialibrary/chisl-methods-resh-ur/image036.png и http://statistica.ru/upload/medialibrary/chisl-methods-resh-ur/image038.png.

Тогда алгоритм последовательных вычислений в методе Ньютона состоит в следующем:

http://statistica.ru/upload/medialibrary/chisl-methods-resh-ur/image040.png

**Блок- схема:**

2.4. Метод простой итерации

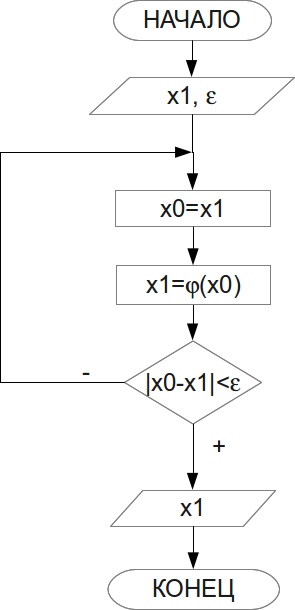
**Алгоритм:**

Метод основан на замене исходного уравнения *F(x)=0* на эквивалентное *x*=ϕ(*x*). Функция ϕ(*x*) выбирается таким образом, чтобы на обоих концах отрезка [a,b] выполнялось условие сходимости ⎢ϕ′(*x*) ⎢< 1. В этом случае в качестве начального приближения можно выбрать любой из концов отрезка. Итерационная формула имеет вид

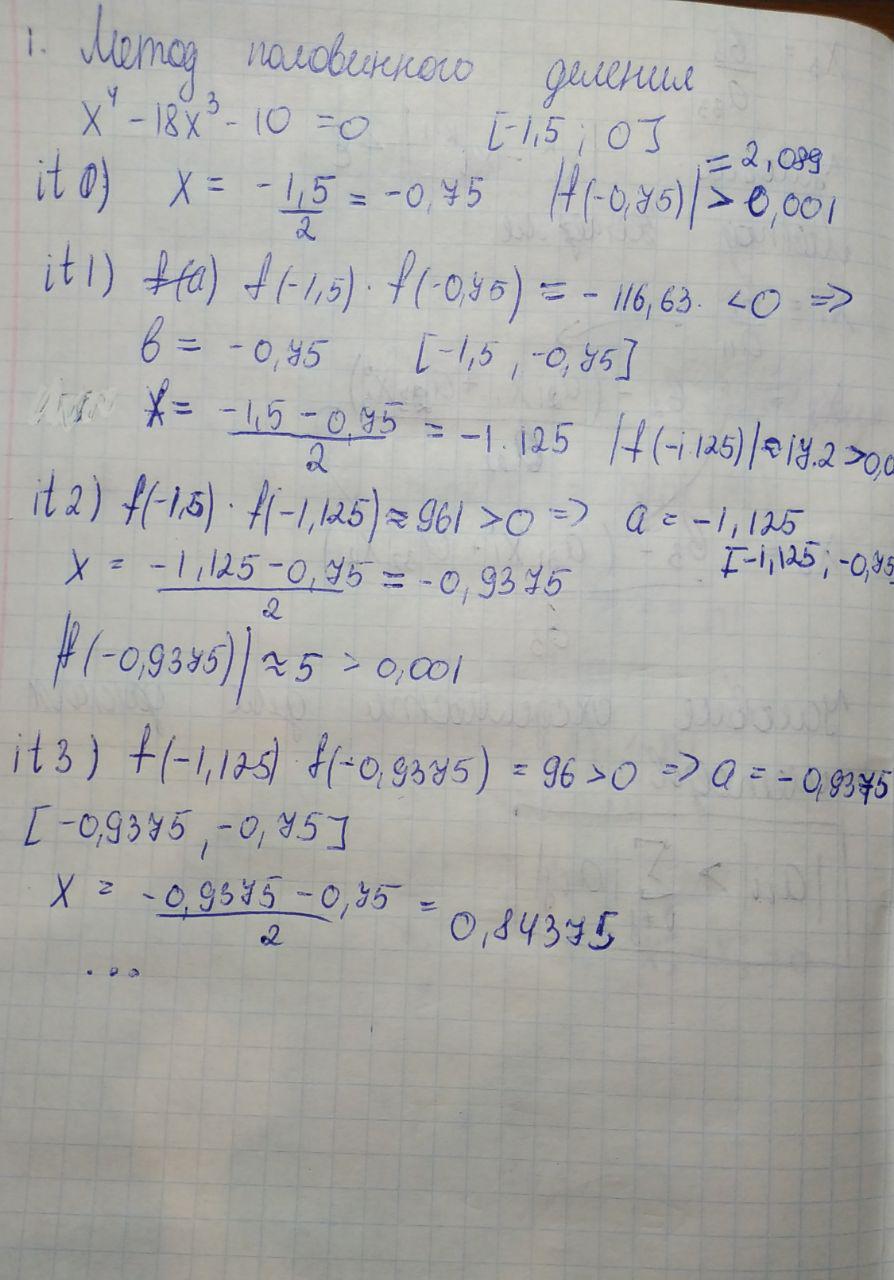
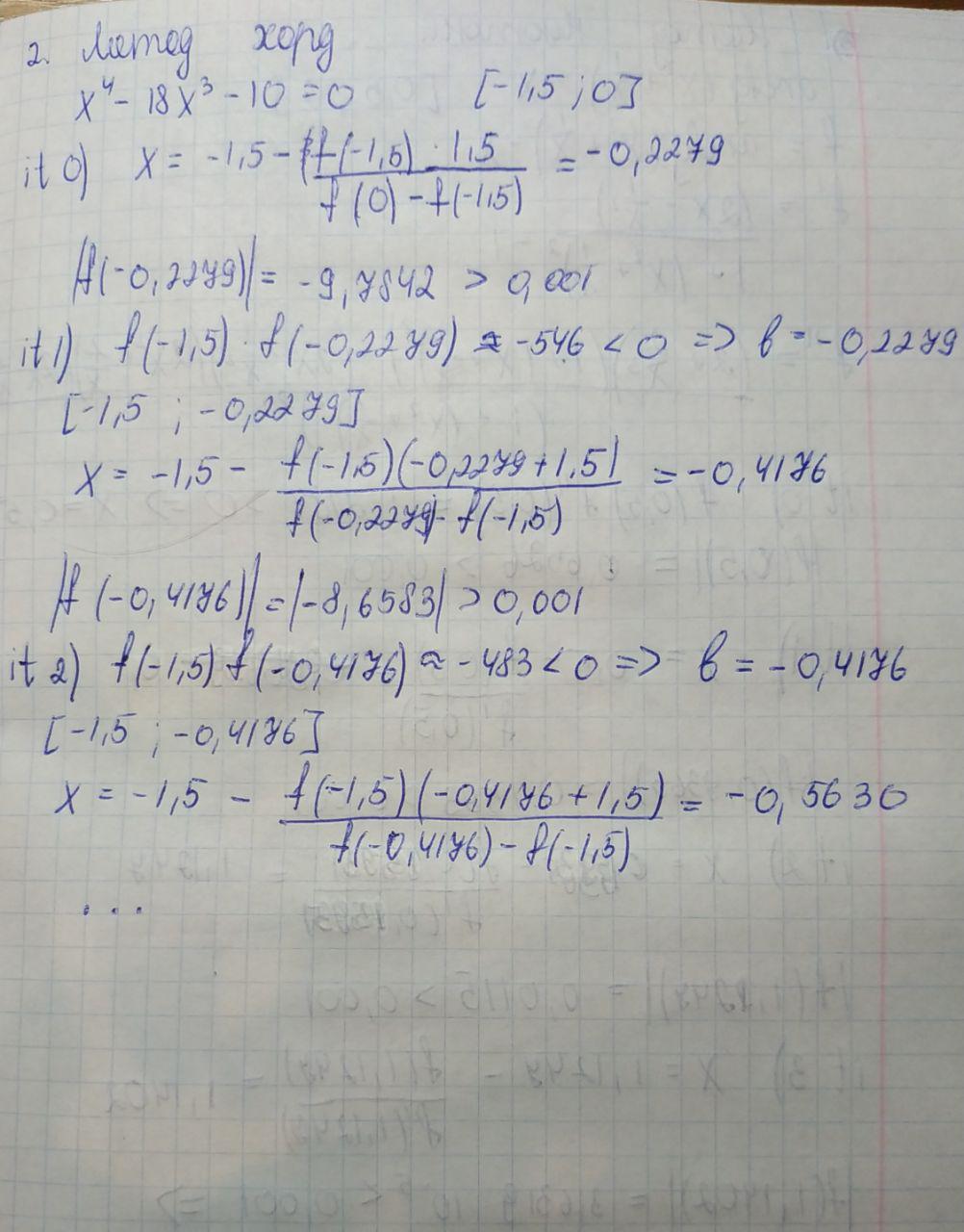
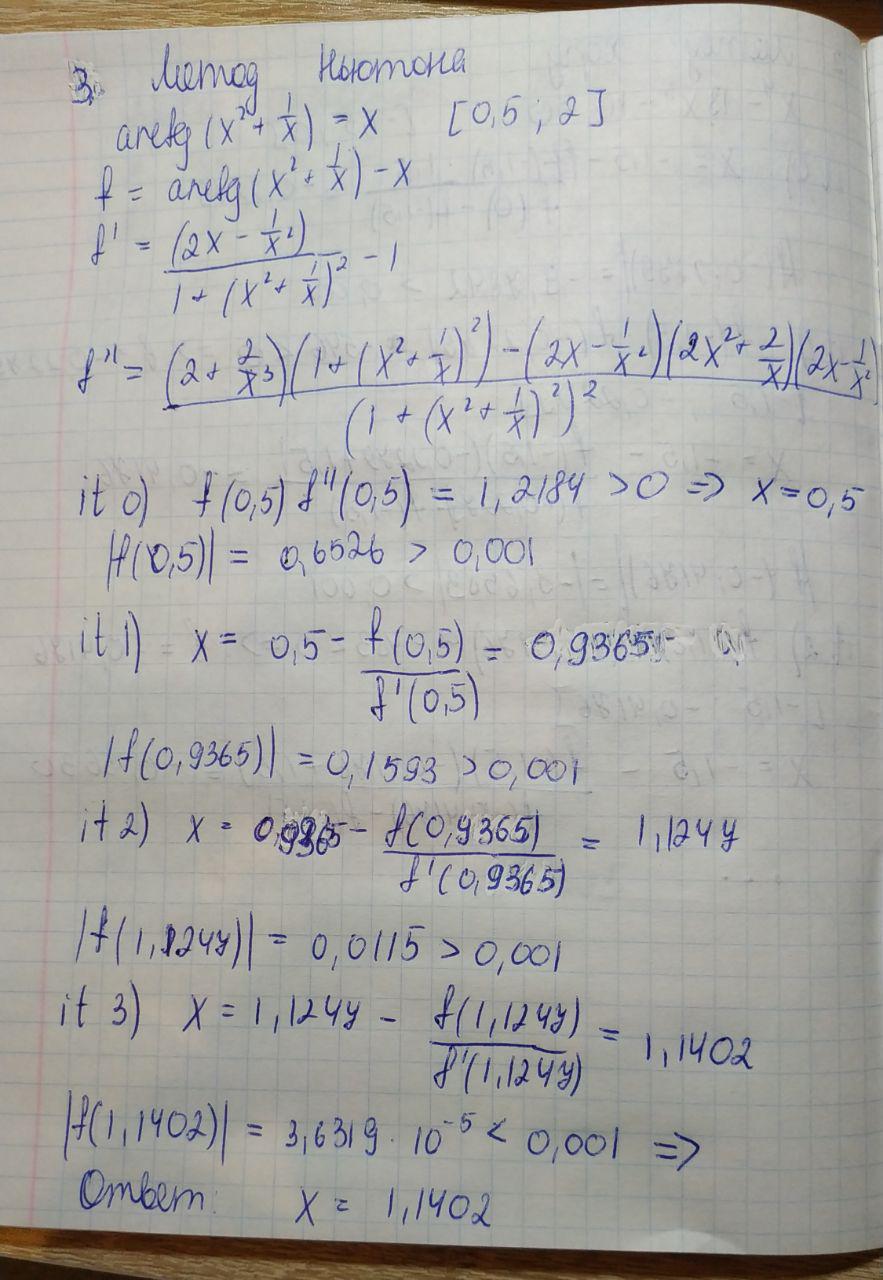
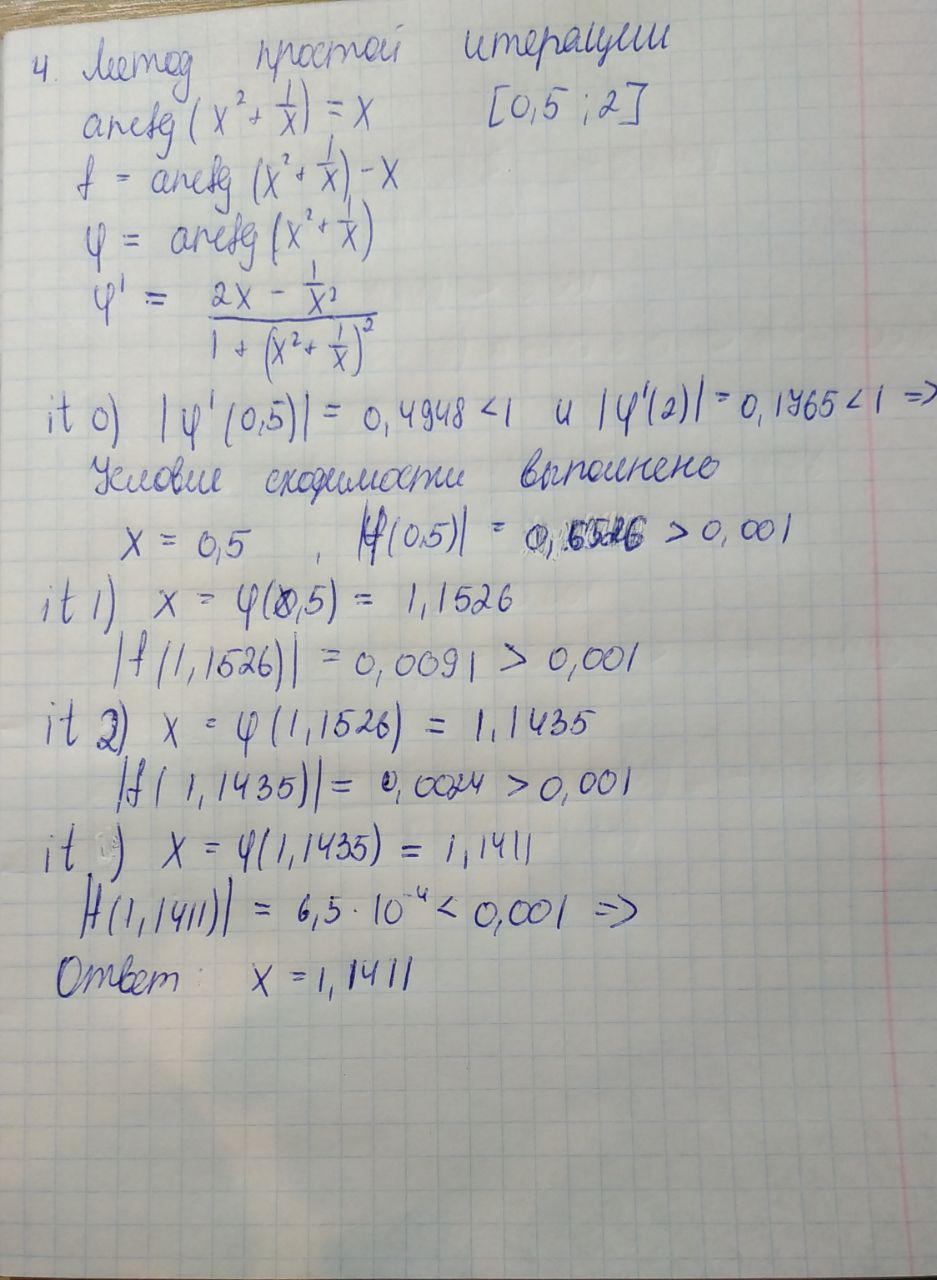


Итерационный процесс продолжается до тех пор, пока не будет выполнено условие ⎪*F(x)|*<ε, где ε - заданная точность.

**Блок-схема:**



3. **Ручной счет показан на изображениях: dihot.jpg, newton.jpg, hord.jpg, simple\_it.jpg.**



4. **Исходный код в файлах: dihot.m, newton.m, hord.m, hord2.m, simple\_it.m**

Метод половинного деления:

function [ x ] = dihot( a,b,e )

x = (a+b)/2;

i = 0;

while abs(f1(x)) > e

if (f1(a)\*f1(x) < 0)

b = x;

else

a = x;

end

x = (a+b)/2;

i = i + 1;

end

fprintf('x= %2.4f it = %d\n',x,i);

end

Метод хорд (1):

function [ x ] = hord( a,b,e )

x=a-(f1(a)./(f1(b)-f1(a))).\*(b-a);

i=0;

while (abs(f1(x))>e)

if f1(a)\*f1(x)<0

b=x;

else

a=x;

end

x=a-(f1(a)./(f1(b)-f1(a))).\*(b-a);

i=i+1;

end

fprintf('x = %f it = %d \n',x,i);

end

Метод хорд (2):

function [ x ] = hord2( a,b,e )

if f1(a)\*f1pr2(b) < 0

xn= b;

x0= a;

else

xn=a;

x0=b;

end

i = 0;

x1=x0-((xn-x0)/(f1(xn)-f1(x0)))\*f1(x0);

x0=x1;

while (abs(f1(x1))>e)

x1=x0-((xn-x0)/(f1(xn)-f1(x0)))\*f1(x0);

x0=x1;

i = i + 1;

end

fprintf('x = %2.4f it = %d\n',x1,i);

end

Метод Ньютона:

function [ x ] = newton( a,b,e )

if f2(a)\*f2pr2(a) > 0

x = a;

else

x = b;

end

i = 0;

while abs(f2(x)) > e

x = x - f2(x)/f2pr1(x);

i = i + 1;

end

fprintf('x = %2.4f it = %d \n',x,i);

end

Метод простой итерации:

function [ x ] = simple\_it( a,b,e )

i = 0;

if (abs(phi2\_pr1(a)) < 1 && abs(phi2\_pr1(b))< 1)

x = a;

while abs(f2(x)) > e

x = phi2(x);

i = i + 1;

end

else

fprintf('Введите новую функцию Phi ');

end

fprintf('x = %2.4f it = %d \n',x,i);

end

5. Для уравнения *x4 -18x3-10=0* метод дихотомии оказался самым производительным, найдя корень на отрезке [-1,5;0] за 14 итераций, в то время как метод хорд справился с данной задачей лишь за 18 итераций.  
Для функции *arctg(x2+1/x)=x* методы Ньютона и простой итерации нашли корень на отрезке [0,5;2] за 3 итерации.  
Можно сделать вывод, что методы простой итерации и Ньютона являются более производительными, чем остальные.

6. **Для получения результатов каждой задачи нужно запустить скрипты (task1\_script.m и task2\_script.m).**