**МФТИ**

Лабораторная работа №1

“Численное решение нелинейного уравнения”

Выполнил

Студент Б03-907

Алиев Артем Эльдарович

Долгопрудный, 2021

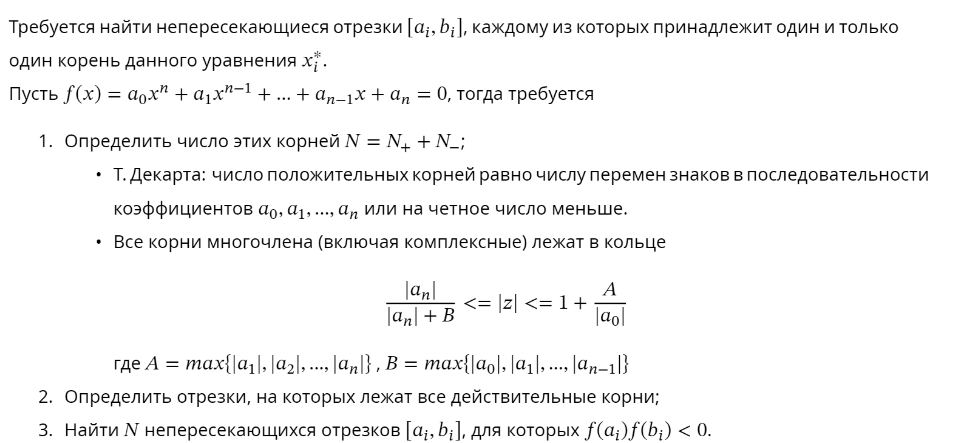
**Задача**

Определить корни уравнений (локализовать), а затем уточнить с помощью методов:

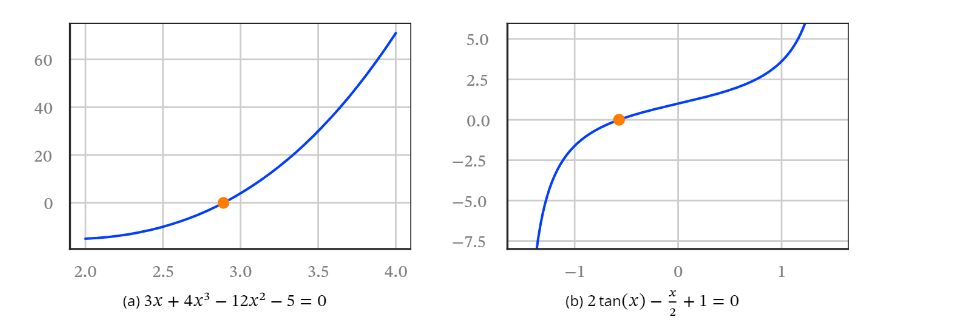
* половинчатое деление
* МПИ
* метод Ньютона
* модифицированный метод Ньютона
* метод секущих

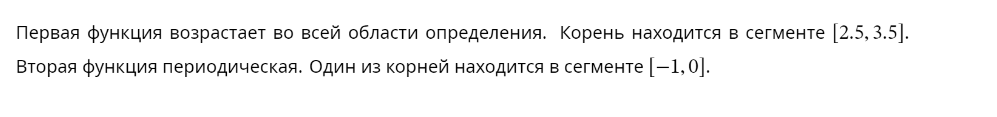
с точностью: **ε = 10-4**

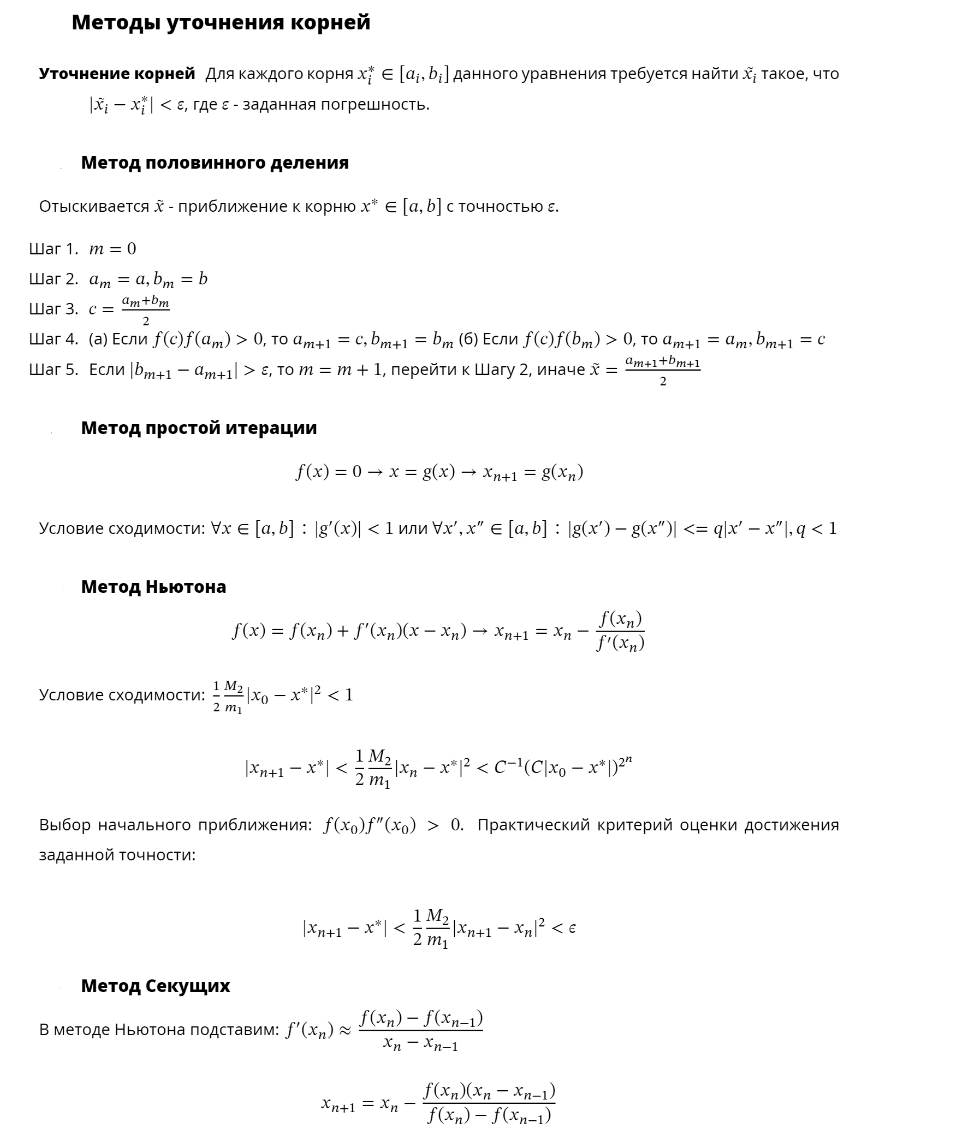
**Локализация**

****

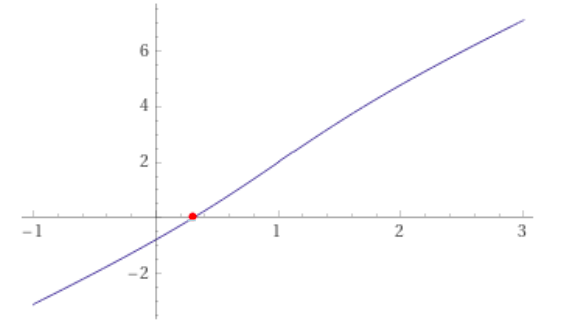
**Графики функций и выбор отрезков**

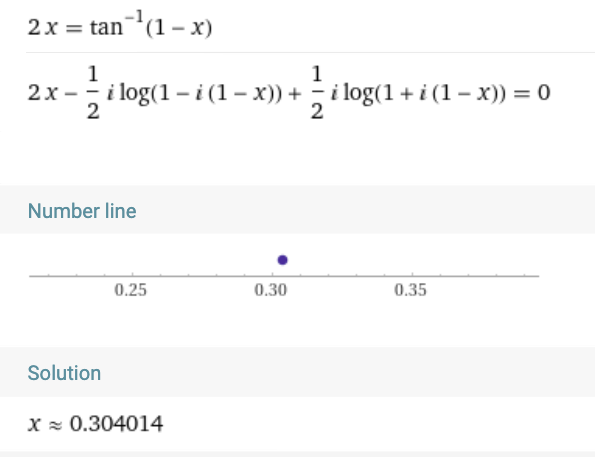
****

****

****

**I. arctg(x-1) + 2x = 0**

****

****

Из графика видно, что отрезок [0; 1] можно назвать отрезком локализации.

1. Решим с помощью бисекции:

Код:

import math

from math import atan

a = 0; b = 1; e = 0.0001; i = 1

def f(x):

return atan(x-1) + 2\*x

y1 = f(a); y2 = f(b)

if (y1 \* y2 > 0) or (y1 \* y2 == 0):

print("no solutions")

else:

n = 1

x = (a+b)/2

y3 = f(x)

print(' ', ' a', ' b', ' (a+b)/2', 'f((a+b)/2)', 'f(a)', ' f(b)')

while (abs(y3) > e):

if i > 9:

print(i, end = '. ')

else:

print(i, end = '. ')

print('%.5f' % round(a, 5), '%.5f' % round(b, 5), end = ' ')

x = (a+b)/2

print('%.5f' % round(x, 5), end = ' ')

if f(x) < 0:

print('%.5f' % round(f(x), 5), end = ' ')

else:

print('%.5f' % round(f(x), 5), end = ' ')

print('%.5f' % round(f(a), 5), '%.5f' % round(f(b), 5))

i += 1

y3 = f(x);

if y1 \* y3 < 0:

b = x

else:

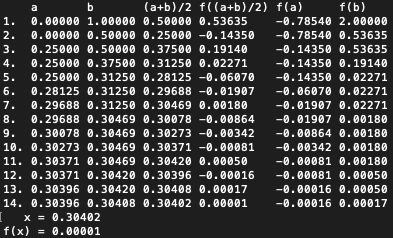
a = x

n += 1

print(' x = %.5f' % round(x, 5))

print('f(x) = %.5f' % round(f(x), 5))

Данные:



1. Решим методом простых итераций:

Код:

import math

from math import atan

e = 0.0001; i = 1

def f(x):

return atan(x-1) + 2\*x

def phi(x):

return -atan(x-1)/2

x0 = 0.5

x1 = phi(x0)

X1 = []

X0 = []

print(' x0', ' f(x0)')

while abs(x0 - x1) > e:

X1.append(x1)

X0.append(x0)

x0 = x1

x1 = phi(x1)

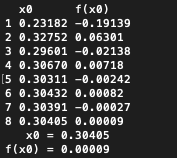
print(i, '%.5f' % round(x0, 5), '%.5f' % round(f(x0), 5))

i += 1

print(' x0 =', '%.5f' % round(x0, 5))

print('f(x0) =', '%.5f' % round(f(x0), 5))

Данные:



1. Решим методом Ньютона:

Код:

import math

from math import atan

from sympy import \*

e = 0.0001; i = 1

x = Symbol('x')

f = atan(x-1) + 2\*x

df = f.diff(x)

x0 = 0.5

x1 = x0 - f/df

X0 = []

F = []

dF = []

print(' x0', ' f(x0)')

while abs(x1.subs(x, x0) - x0) > e:

X0.append(x0)

F.append(f.subs(x, x0))

dF.append(df.subs(x, x0))

x0 = x1.subs(x, x0)

x1 = x0 - f / df

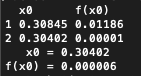
print(i, '%.5f' % round(x0, 5), '%.5f' % round(f.subs(x, x0), 5))

i += 1

print(' x0 =', '%.5f' % round(x0, 5))

print('f(x0) =', '%.6f' % round(f.subs(x, x0), 6))

Данные:



1. А также модифицированным методом Ньютона:

Код:

import math

from math import atan

from sympy import \*

e = 0.0001; i = 1

x = Symbol('x')

f = atan(x-1) + 2\*x

df = f.diff(x)

x0 = 0.5

x1 = x0 - f/df

X0 = []

F = []

dF = []

print(' x0', ' f(x0)')

x\_tmp = x0

flag = True

while abs(x1.subs(x, x0) - x0) > e:

if abs(x1.subs(x, x0) - x0) < e + 0.006:

flag = False

X0.append(x0)

F.append(f.subs(x, x0))

dF.append(df.subs(x, x0))

print(i, '%.5f' % round(x0, 5), '%.5f' % round(f.subs(x, x0), 5))

x0 = x1.subs(x, x0)

i += 1

if flag:

x1 = x0 - f / df

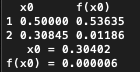
else:

x1 = x0 - f / df.subs(x, x\_tmp)

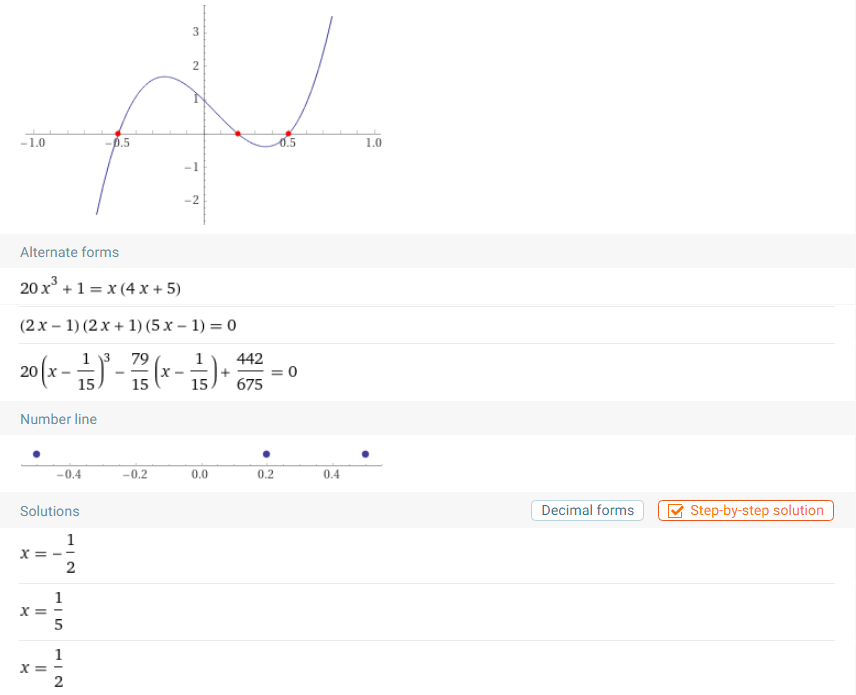
print(' x0 =', '%.5f' % round(x0, 5))

print('f(x0) =', '%.6f' % round(f.subs(x, x0), 6))

Данные:



**II. 20x\*\*3 - 4x\*\*2 -5x + 1 = 0**

****

Из графика видно, что отрезки [-1; 0], [0; 0,4], [0,4; 1] можно назвать отрезками локализации.

1. Решим с помощью бисекции:

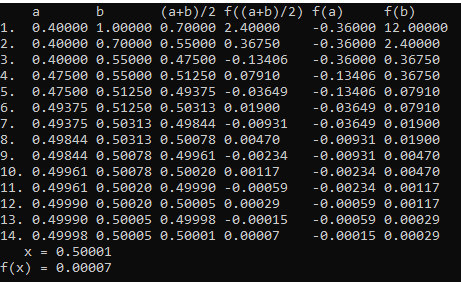
Код:

Аналогичен коду прошлой функции, только теперь вводим другую **f(x)**, и соответственно новые **a** и **b** (концы отрезков локализации)

Данные:







1. Методом простых итераций:

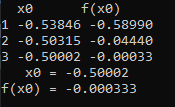
НЕ УДАЛОСЬ ПОДОБРАТЬ ОБРАТНУЮ ФУНКЦИЮ

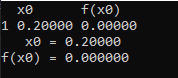
1. Метод Ньютона:

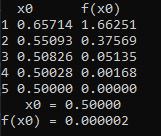
Код:

Аналогичен коду прошлой функции, только теперь вводим другую **f(x)** и разные x0 (в зависимости от отрезков локализации)

Данные:





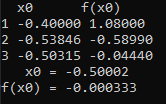


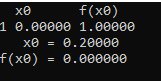
1. Модифицированный метод Ньютона:

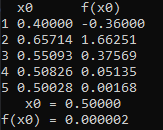
Код:

Аналогичен коду прошлой функции, только теперь вводим другую **f(x)** и разные x0 (в зависимости от отрезков локализации)

Данные:







[*Ссылка*](https://github.com/alievgithub/MIPT_computer_math/tree/main/5th%20semester/Lab%201) *на исходные коды на GitHub*