Лабораторная работа №1

Тема: « Численное решение нелинейных уравнений»

Самоховец Давид

2 курс,1 группа

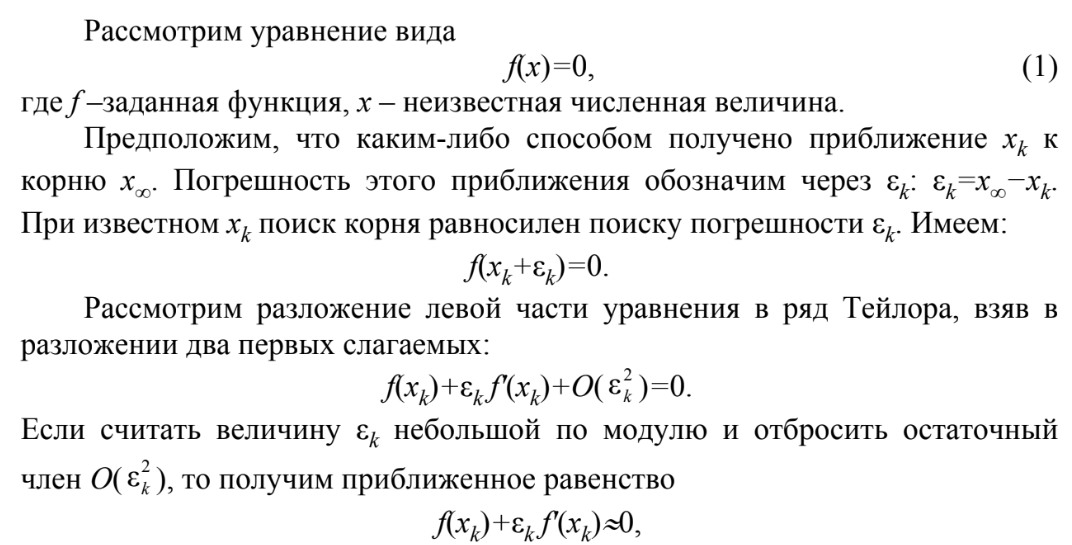
*1) Постановка задачи*

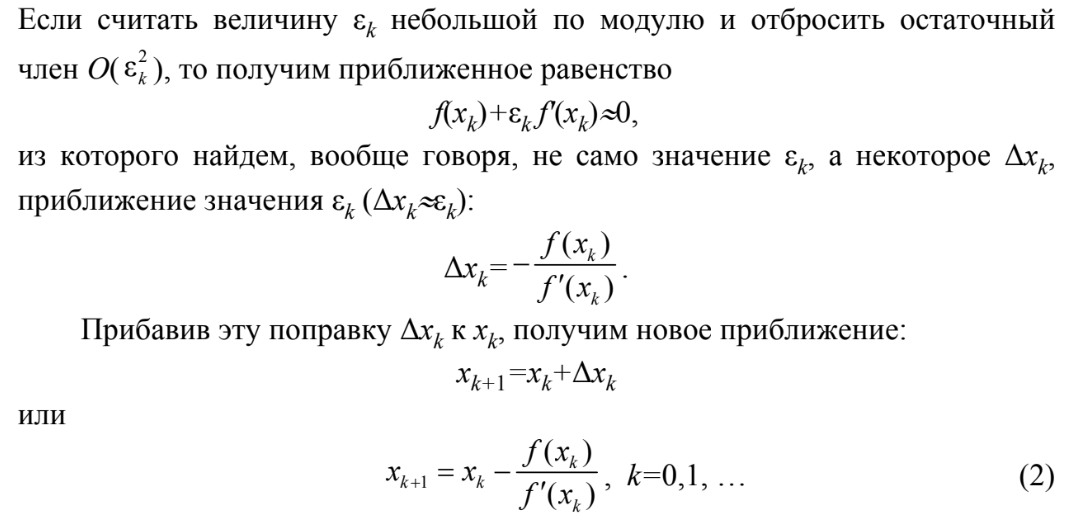
Решить уравнение f (x) 0  согласно своему варианту с точностью   10-7 методом простой итерации и методом Ньютона. Корень отделяем сначала графически, затем с помощью метода половинного деления до   10-2 . Если корней несколько, то необходимо найти ближайший к началу координат. Провести сравнительный анализ полученных результатов.

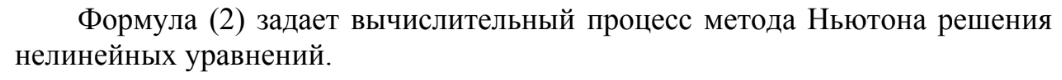
*f(x) = 4x – 5x – 2*

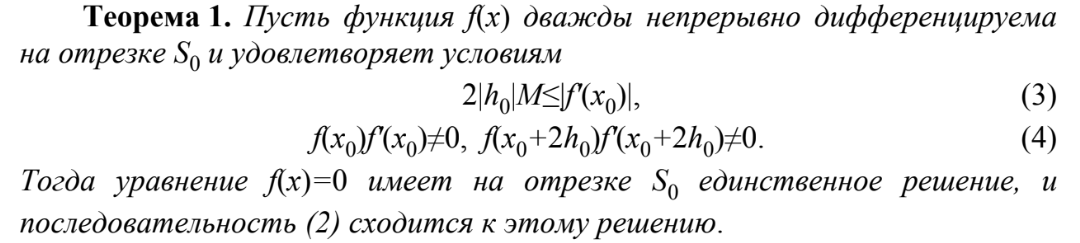
*2) Краткие теоретические сведения*

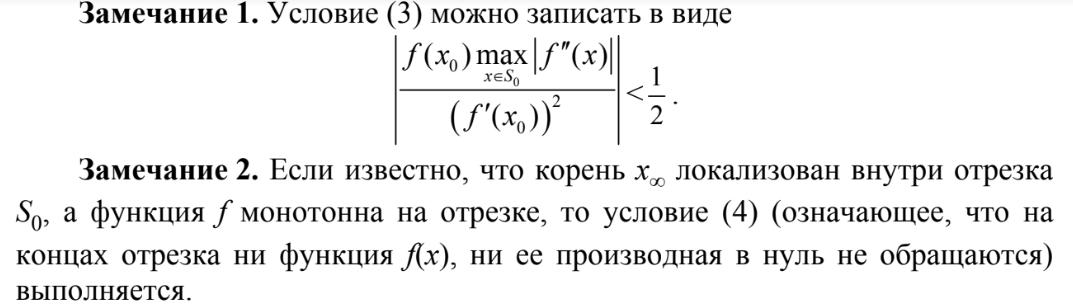
**Метод Ньютона**



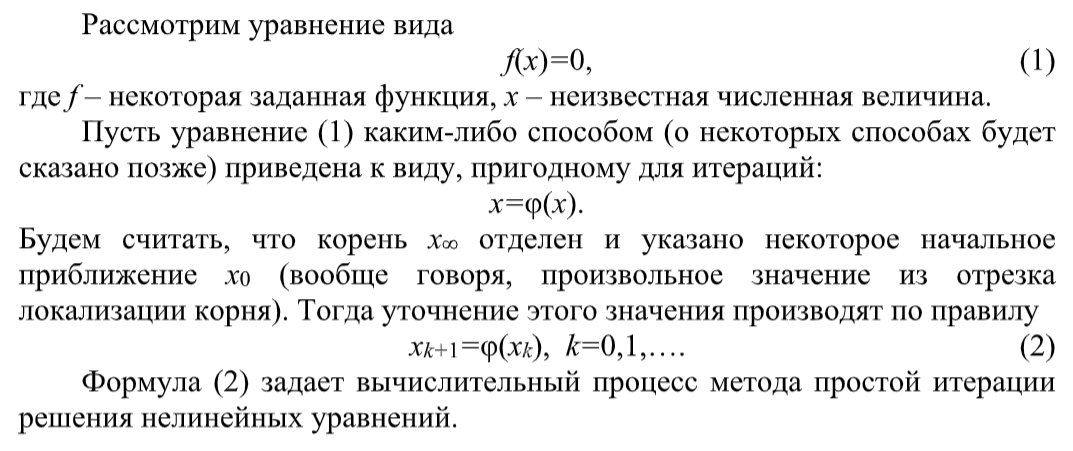


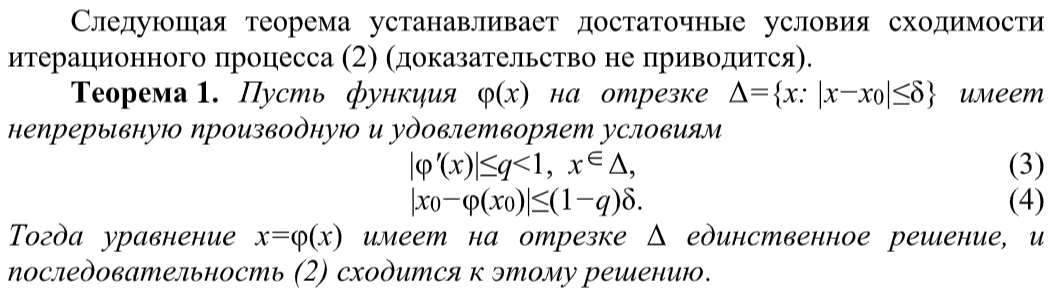






**МПИ**





*3) Листинг программы*

Дихотомия:

**from** prettytable **import** PrettyTable

**def** get\_y(x):

**return** 4\*\*x - 5\*x — 2

**def** get\_borders():

table = PrettyTable()

table.field\_names = [**'k'**, **'ak'**, **'bk'**, **'mid'**, **'f(ak)'**, **'f(bk)'**, **'f(mid)'**, **'bk - ak'**]

eps = 0.01

a = -1.

b = 0.

k = 0

**while** abs(a - b) > eps:

mid = (a + b) / 2.

**if** get\_y(mid) > 0.:

a = mid

**else**:

b = mid

table.add\_row([k, a, b, mid, get\_y(a), get\_y(b), get\_y(mid), b - a])

k += 1

print(table)

**return** a, b

l, r = get\_borders()

МПИ:

**from** prettytable **import** PrettyTable

**import** script0

**def** get\_next\_x(x):

**return** (4\*\*x - 2) / 5

eps = 0.0000001

a = script0.l

table = PrettyTable()

table.field\_names = [**'k'**, **'xk'**, **'ek'**]

curr\_x = a

next\_x = get\_next\_x(curr\_x)

k = 0

**while** abs(curr\_x - next\_x) > eps:

table.add\_row([k, curr\_x, abs(curr\_x - next\_x)])

k += 1

curr\_x = next\_x

next\_x = get\_next\_x(curr\_x)

print(**'Простые итерации:'**)

print(curr\_x)

print(table)

Ньютон:

**from** prettytable **import** PrettyTable

**import** script0

**from** math **import** log

**def** get\_derivative\_func(x):

**return** (4\*\*x) \* log(4) — 5

**def** get\_next\_x(x):

**return** x - script0.get\_y(x) / get\_derivative\_func(x)

eps = 0.0000001

a = script0.l

table = PrettyTable()

table.field\_names = [**'k'**, **'xk'**, **'ek'**]

curr\_x = a

next\_x = get\_next\_x(curr\_x)

k = 0

**while** abs(curr\_x - next\_x) > eps:

table.add\_row([k, curr\_x, abs(curr\_x - next\_x)])

k += 1

curr\_x = next\_x

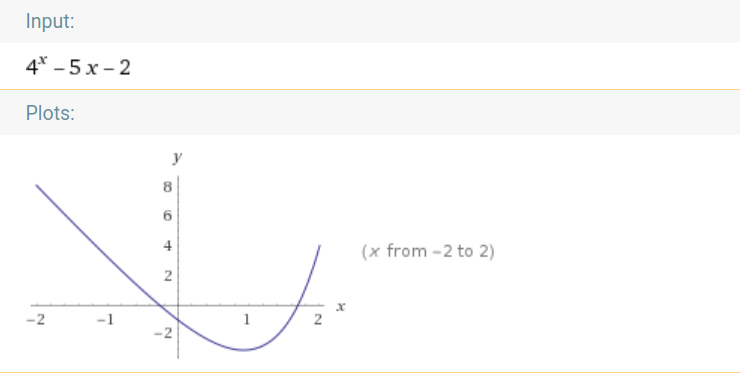
next\_x = get\_next\_x(curr\_x)

print(**'Метод Ньютона:'**)

print(curr\_x)

print(table)

*4) Результаты*





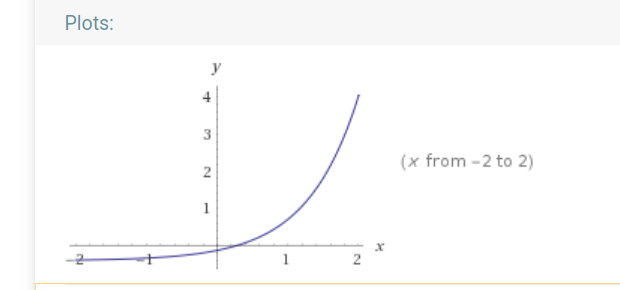
Проверка условий :

1. Имеем 4x – 5x - 2 = 0

x = (4x – 2) / 5 = φ(x)

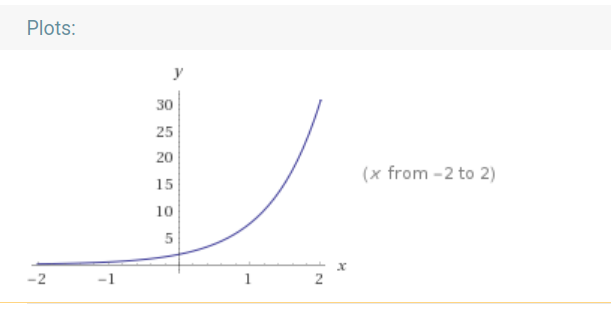
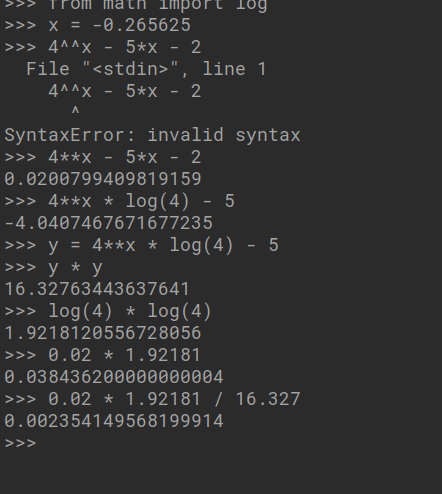
φ’(x) = (4x \* ln4 — 2) / 5;

График :

Видно, что в промежутке от -1 до 0 φ’(x) <1;

Значит МПИ сходится.

Наша функция f(x) = 4x – 5x – 2 дважды непрерывно дифференцируема. Также f’(x) ≠ 0 на нашем промежутке.



Метод Ньютона сходится.

*5) Вывод*

**Метод дихотомии**

*Плюсы* : не требует приведения функции к специальному вида, а также устойчив к ошибкам округления.

*Минусы* : требует отделения корня, а также низкая скорость схидимости.

**Метод Ньютона :**

*Плюсы :* очень высокая скорость сходимости

*Минусы* : большая зависимость метода от начального приближения.

**МПИ :**

*Минусы*: требуется приводить к специальному виду . Меньшая скорость сходимости по сравнению с методом Ньютона