Министерство науки и высшего образования РФ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Курский государственный университет»

Кафедра программного обеспечения и администрирования информационных систем

Направление подготовки математическое обеспечение и администрирование информационных систем

Форма обучения очная

**Отчет**

**по лабораторной работе №2**

«Решение нелинейного уравнения методами хорд, касательных и комбинированным методом»

дисциплина «Методы вычислений»

Выполнил:

студент группы 313.1 Козявин М.С.

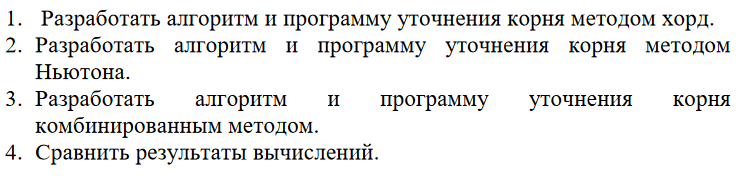
Проверил:

доцент кафедры АГиТОМ Селиванова И. В.

Курск, 2023

Цель: Изучение особенностей применения методов хорд, касательных и комбинированного метода к решению нелинейного уравнения.

Задание:



|  |  |
| --- | --- |
| **Номер варианта** | **Уравнение** |
| 9 |  |

Код программы уточнения корня методом хорд:

import math

import matplotlib.pyplot as plt

def f(x):

return math.acos(x\*x)-x

xs = [x / 100 for x in range(-100, 101)]

ys = [f(x) for x in xs]

plt.plot(xs, ys)

plt.grid()

a, b = -1, 1

plt.scatter(a, f(a), color="black")

plt.scatter(b, f(b), color="black")

eps = 0.0001

new = a - ((b - a)\*f(a))/(f(b)-f(a))

count = 0

while abs(f(new)) > eps:

if f(a) \* f(new) < 0:

b = new

plt.scatter(b, f(b))

else:

a = new

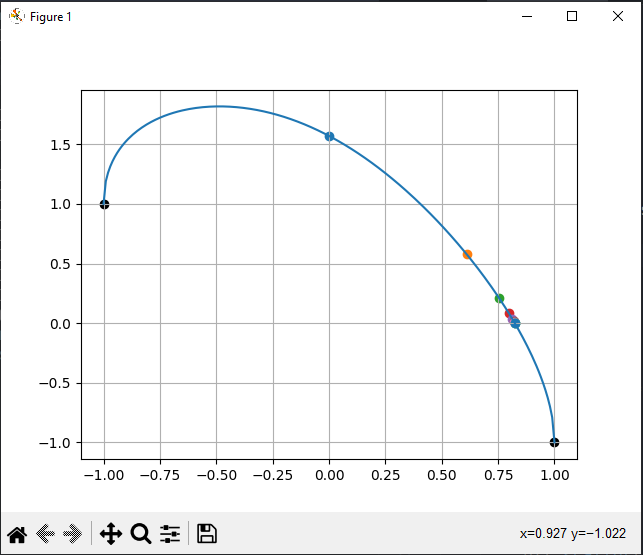
plt.scatter(a, f(a))

new = a - ((b - a)\*f(a))/(f(b)-f(a))

count += 1

print(count)

plt.show()



Код программы уточнения корня методом Ньютона:

import math

import matplotlib.pyplot as plt

def f(x):

return math.acos(x\*x)-x

# return (x+3)\*(x+3) #-5 5

# return 2 - x - math.log(x) # 1.5 1.6

def diff(x):

return (f(x+0.00001) - f(x-0.00001))/0.00002

xs = [x / 100 for x in range(-100, 101)]

ys = [f(x) for x in xs]

plt.grid()

a, b = -1, 1

plt.scatter(a, f(a), color="black")

plt.scatter(b, f(b), color="black")

eps = 0.0001

prev = 0

a = a + eps # если не определено

b = b - eps

left = a

right = b

new = b - f(b)/diff(b)

plt.scatter(new, f(new))

plt.axline((b, f(b)), (new, 0), color="red")

count = 1

while abs(f(new)) > eps:

prev = new

new = prev - f(prev)/diff(prev)

plt.scatter(new, f(new))

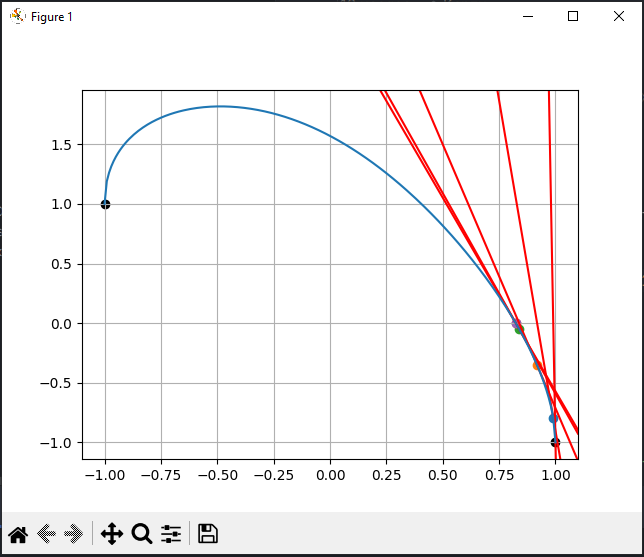
plt.axline((prev, f(prev)), (new, 0), color="red")

count += 1

print(count)

plt.plot(xs, ys)

plt.show()



Код программы уточнения корня комбинированным методом:

import math

import matplotlib.pyplot as plt

def f(x):

return math.acos(x\*x)-x

# return x\*x #-5 5

# return 2 - x - math.log(x) # 1.5 1.6

def diff(x):

return (f(x+0.00001) - f(x-0.00001))/0.00002

def diff2(x):

return (diff(x+0.00001) - diff(x-0.00001))/0.00002

xs = [x / 100 for x in range(-100, 101)]

ys = [f(x) for x in xs]

plt.grid()

a, b = -1, 1

plt.scatter(a, f(a), color="black")

plt.scatter(b, f(b), color="black")

eps = 0.0001

a = a + eps # если не определено

b = b - eps

count = 0

if diff(b)\*diff2(b) > 0: # недостаток избыток

while abs(a - b) > eps:

a = a - ((b - a) \* f(a)) / (f(b) - f(a))

b = b - f(b) / diff(b)

plt.scatter(a, f(a), color="red")

plt.scatter(b, f(b), color="green")

count += 1

else: # избыток недостаток

while abs(a - b) > eps:

b = b - ((b - a) \* f(b)) / (f(b) - f(a))

a = a - f(a) / diff(a)

plt.scatter(a, f(a), color="green")

plt.scatter(b, f(b), color="red")

count += 1

print(count)

plt.plot(xs, ys)

plt.show()

