Міністерство освіти і науки України  
НТУУ «КПІ ім. Ігоря Сікорського»  
Кафедра диференціальних рівнянь

Лабораторна робота №8  
Тема: “Числове розв’язування нелінійних рівнянь”  
Варіант № 6

Виконав студент 2-го курсу  
ТЕФ, групи ТР-71  
Зуєв Михайло Олександрович

Київ – 2018

# Код програми:

""" Лабораторна работа номер 8

з курсу Чисельні методи, варіант 6

Завдання: Знайти всі корені рівняння f(x) = 0 на відрізку [-10, 10] за допомогою

другої модифікації методу дотичних (Ньютона).

f(x) = 0.5 + sqrt(abs(x)) - exp(-x)

Виконав студент 2 курсу: Зуєв Михайло Олександрович

"""

import matplotlib.pyplot as plt

import numpy as np

import math

def function(arg):

return 0.5 + math.sqrt(math.fabs(arg)) - math.exp(-arg)

def function\_derivative(x):

return x / (2 \* math.sqrt(math.fabs(x \* x \* x))) + math.exp(-x)

def solve(left, right):

"""

Находит корень x0 на промежутке x0 є [left; right], такой что f(x0) = 0

:type left: float

:type right: float

:param left: левая граница

:param right: правая граница

:return: корень x0: f(x0) = 0

"""

# точность с которой ищем корни

eps = 1e-3

# счетчик

k = 0

# текущее приближение (x[k])

cur = 0

# предыдущее приближение (x[k-1])

prev = function(left) if function(left) >= 0 else function(right)

# значение производной

der = function\_derivative(prev)

# усливие выхода

flag = True

# печатаем таблицу

print("For x in [{0}; {1}]".format(left, right))

print("-----------------------------")

print("| k | xk | f(xk) |")

print("-----------------------------")

while flag:

if k % 4:

# считаем значени производной раз в 4 итерации

der = function\_derivative(prev)

k += 1

# считаем следующее приближение

cur = prev - function(prev) / der

# проверяем условие выхода

flag = abs(cur - prev) > eps if cur < 1 else abs((cur - prev) / cur) > eps

# печатаем таблицу

print("|{0:3} | {1:8.3} |{2:10.3} |".format(k, cur, function(cur)))

prev = cur

print("-----------------------------")

return cur

# Границы поиска корней

a = -10

b = 10

# Шаг

h = 0.5

# Фиксируем интервалы на которых функция меняет знак

intervals = []

left = right = a

left\_value = right\_value = function(left)

while right <= b:

right += h

right\_value = function(right)

if (left\_value < 0 < right\_value) or (right\_value < 0 < left\_value):

interval = (left, right)

intervals.append(interval)

left = right

left\_value = right\_value

# Рисуем график

xData = np.arange(-1, 1, 0.01)

yData = [function(x) for x in xData]

plt.plot(xData, yData, label='f(x)')

plt.plot([-1, 1], [0, 0], 'k--')

plt.plot([0, 0], [min(yData), max(yData)], 'k--')

plt.xlabel(r'$x$')

plt.ylabel(r'$y$')

plt.title(r'$y = f(x)$')

plt.legend()

plt.show()

# Корни уравнения

roots = [solve(i[0], i[1]) for i in intervals]

print("Root(s) of the equation: ")

for x in roots:

print("x = {0:6.4f}, f(x) = {1:4.2f}".format(x, function(x)))

# Результати роботи:

For x in [-10; 0.5]

-----------------------------

| k | xk | f(xk) |

-----------------------------

| 1 | -0.00802 | -0.418 |

| 2 | -0.0995 | -0.289 |

| 3 | -0.701 | -0.679 |

| 4 | -0.223 | -0.278 |

| 5 | -0.0275 | -0.362 |

| 6 | -0.21 | -0.275 |

| 7 | 1.75 | 1.65 |

| 8 | -1.23 | -1.82 |

| 9 | 2.06 | 1.81 |

| 10 | -1.74 | -3.88 |

| 11 | -1.01 | -1.24 |

| 12 | -0.459 | -0.405 |

| 13 | -0.279 | -0.294 |

| 14 | 0.502 | 0.604 |

| 15 | 0.0417 | -0.255 |

| 16 | 0.117 | -0.0487 |

| 17 | 0.131 | -0.0157 |

| 18 | 0.138 | -0.000146 |

| 19 | 0.138 | -1.25e-08 |

-----------------------------

Root(s) of the equation:

x = 0.1378, f(x) = -0.00

