Пальонка Анастасія 2-8 ,Варіант 20



import numpy as np

import matplotlib.pyplot as plt

from scipy.optimize import least\_squares

def func(x):

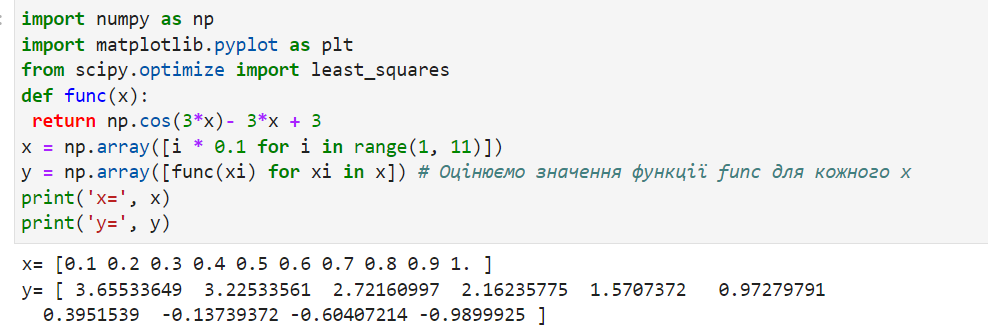
return np.cos(3\*x)- 3\*x + 3

x = np.array([i \* 0.1 for i in range(1, 11)])

y = np.array([func(xi) for xi in x]) # Оцінюємо значення функції func для кожного x

print('x=', x)

print('y=', y)



# МНК. Наближення параболою

def fun\_parabola(a, x, y):

return a[0] + a[1] \* x + a[2] \* x\*\*2 - y

a0\_parabola = np.array([1, 1, 1])

res\_lsq\_parabola = least\_squares(fun\_parabola, x0=a0\_parabola, args=(x, y))

print("a0 = %.2f, a1 = %.2f, a2 = %.2f" % tuple(res\_lsq\_parabola.x))

f\_parabola = lambda x: sum([u \* v for u, v in zip(res\_lsq\_parabola.x, [1, x, x\*\*2])])

x\_p\_parabola = np.linspace(min(x), max(x), 20)

y\_p\_parabola = f\_parabola(x\_p\_parabola)

# МНК. Наближення прямою

def fun\_line(a, x, y):

return a[0] + a[1] \* x - y

a0\_line = np.array([1, 1])

res\_lsq\_line = least\_squares(fun\_line, x0=a0\_line, args=(x, y))

print("a0 = %.2f, a1 = %.2f" % tuple(res\_lsq\_line.x))

f\_line = lambda x: res\_lsq\_line.x[0] + res\_lsq\_line.x[1] \* x

x\_p\_line = np.linspace(min(x), max(x), 20000)

y\_p\_line = f\_line(x\_p\_line)

# Графіки

plt.plot(x, y, 'o', label='Дані')

plt.plot(x\_p\_parabola, y\_p\_parabola, 'b', label='МНК (парабола)')

plt.plot(x\_p\_line, y\_p\_line, 'r', label='МНК (пряма)')

plt.title("МНК\_наближення параболою та прямою")

plt.legend()

plt.grid(True)

plt.show()

