

ЛАБОРАТОРНА РОБОТА №10

Тема: «Метод найменших квадратів»

Трубчанінов Андрій Сергійович

ФІТ 2-8

В-29

Код:

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.optimize import least_squares

# Задана функція
def func(x):
    return np.sin(x + 1) + x

# Задані точки x_i
x = np.array([i * 0.1 for i in range(10)])

# Знаходимо значення y_i = f(x_i)
y = np.array([func(xi) for xi in x])

# Виводимо значення x_i та y_i
print('x_i =', x)
print('y_i =', y)

# Функція, яку будемо наближати за МНК (параболою)
def fun_parabola(a, x, y):
    return a[0] + a[1] * x + a[2] * x**2 - y

# Початкове наближення для параметрів a (парабола)
a0_parabola = np.array([1, 1, 1])

# Застосовуємо МНК для параболі
res_lsq_parabola = least_squares(fun_parabola, x0=a0_parabola, args=(x, y))

# Виводимо отримані параметри для параболі
print("Парабола: a0 = %.2f, a1 = %.2f, a2 = %.2f" % tuple(res_lsq_parabola.x))
```

```
# Виводимо отримані параметри для параболі
print("Парабола: a0 = %.2f, a1 = %.2f, a2 = %.2f" % tuple(res_lsq_parabola.x))

# Побудова графіку для параболі
f_parabola = lambda x: sum([u * v for u, v in zip(res_lsq_parabola.x, [1, x, x**2])])
x_p = np.linspace(min(x), max(x), 100)
y_parabola = f_parabola(x_p)

plt.plot(x, y, 'o', label='Дані')
plt.plot(x_p, y_parabola, 'b', label='МНК (Парабола)')
plt.title("МНК наближення функції (Парабола)")
plt.xlabel("x")
plt.ylabel("y")
plt.legend()
plt.grid(True)
plt.show()

# Функція, яку будемо наближати за МНК (прямою)
def fun_line(a, x, y):
    return a[0] + a[1] * x - y

# Початкове наближення для параметрів a (пряма)
a0_line = np.array([1, 1])

# Застосовуємо МНК для прямої
res_lsq_line = least_squares(fun_line, x0=a0_line, args=(x, y))

# Виводимо отримані параметри для прямої
print("Пряма: a0 = %.2f, a1 = %.2f" % tuple(res_lsq_line.x))
```

```
# Функція, яку будемо наближати за МНК (прямою)
def fun_line(a, x, y):
    return a[0] + a[1] * x - y

# Початкове наближення для параметрів a (пряма)
a0_line = np.array([1, 1])

# Застосовуємо МНК для прямої
res_lsq_line = least_squares(fun_line, x0=a0_line, args=(x, y))

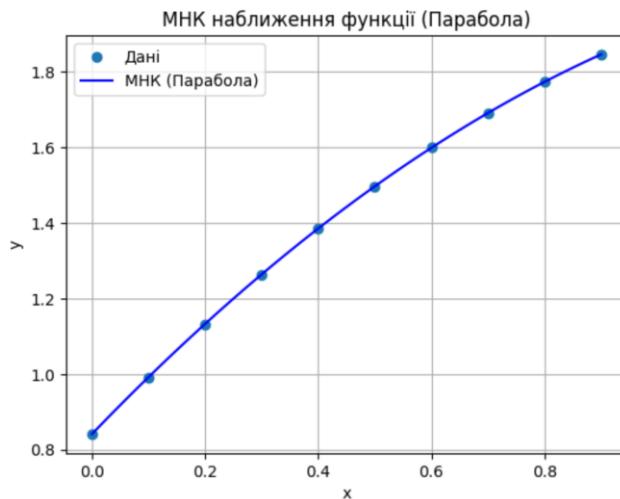
# Виводимо отримані параметри для прямої
print("Пряма: a0 = %.2f, a1 = %.2f" % tuple(res_lsq_line.x))

# Побудова графіку для прямої
f_line = lambda x: res_lsq_line.x[0] + res_lsq_line.x[1] * x
y_line = f_line(x_p)

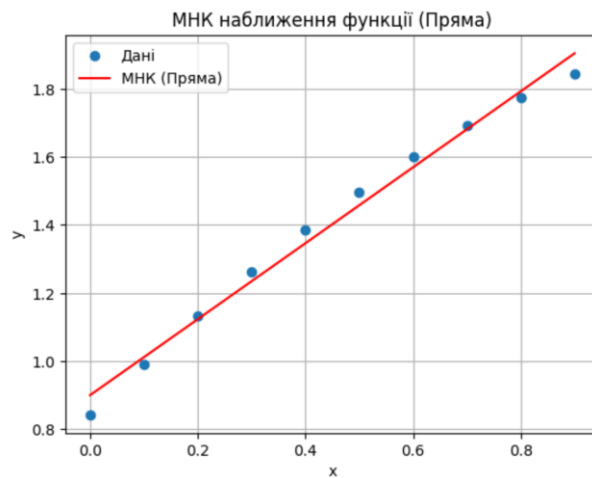
plt.plot(x, y, 'o', label='Дані')
plt.plot(x_p, y_line, 'r', label='МНК (Пряма)')
plt.title("МНК наближення функції (Пряма)")
plt.xlabel("x")
plt.ylabel("y")
plt.legend()
plt.grid(True)
plt.show()
```

```
xi = [0. 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9]
yi = [0.84147098 0.99120736 1.13203909 1.26355819 1.38544973 1.49749499
1.5995736 1.69166481 1.77384763 1.84630009]
Парабола: a0 = 0.84, a1 = 1.56, a2 = -0.49
```

```
xi = [0. 0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8 0.9]
yi = [0.84147098 0.99120736 1.13203909 1.26355819 1.38544973 1.49749499
1.5995736 1.69166481 1.77384763 1.84630009]
Парабола: a0 = 0.84, a1 = 1.56, a2 = -0.49
```



Пряма: a0 = 0.90, a1 = 1.12



Код зі скрінів:

```
import numpy as np
```

```
import matplotlib.pyplot as plt
from scipy.optimize import least_squares

# Задана функція
def func(x):
    return np.sin(x + 1) + x

# Задані точки  $x_i$ 
x = np.array([i * 0.1 for i in range(10)])

# Знаходимо значення  $y_i = f(x_i)$ 
y = np.array([func(x_i) for x_i in x])

# Виводимо значення  $x_i$  та  $y_i$ 
print('x_i =', x)
print('y_i =', y)

# Функція, яку будемо наближати за МНК (параболою)
def fun_parabola(a, x, y):
    return a[0] + a[1] * x + a[2] * x**2 - y

# Початкове наближення для параметрів a (парабола)
a0_parabola = np.array([1, 1, 1])
```

```
# Застосовуємо МНК для параболі
```

```
res_lsq_parabola = least_squares(fun_parabola, x0=a0_parabola,  
args=(x, y))
```

```
# Виводимо отримані параметри для параболі
```

```
print("Парабола: a0 = %.2f, a1 = %.2f, a2 = %.2f" %  
tuple(res_lsq_parabola.x))
```

```
# Побудова графіку для параболі
```

```
f_parabola = lambda x: sum([u * v for u, v in  
zip(res_lsq_parabola.x, [1, x, x**2])])
```

```
x_p = np.linspace(min(x), max(x), 100)
```

```
y_parabola = f_parabola(x_p)
```

```
plt.plot(x, y, 'o', label='Дані')
```

```
plt.plot(x_p, y_parabola, 'b', label='МНК (Парабола)')
```

```
plt.title("МНК наближення функції (Парабола)")
```

```
plt.xlabel("x")
```

```
plt.ylabel("y")
```

```
plt.legend()
```

```
plt.grid(True)
```

```
plt.show()
```

```
# Функція, яку будемо наближати за МНК (прямою)
```

```
def fun_line(a, x, y):  
    return a[0] + a[1] * x - y  
  
# Початкове наближення для параметрів а (пряма)  
a0_line = np.array([1, 1])  
  
# Застосовуємо МНК для прямої  
res_lsq_line = least_squares(fun_line, x0=a0_line, args=(x, y))  
  
# Виводимо отримані параметри для прямої  
print("Пряма: a0 = %.2f, a1 = %.2f" % tuple(res_lsq_line.x))  
  
# Побудова графіку для прямої  
f_line = lambda x: res_lsq_line.x[0] + res_lsq_line.x[1] * x  
y_line = f_line(x_p)  
  
plt.plot(x, y, 'o', label='Дані')  
plt.plot(x_p, y_line, 'r', label='МНК (Пряма)')  
plt.title("МНК наближення функції (Пряма)")  
plt.xlabel("x")  
plt.ylabel("y")  
plt.legend()  
plt.grid(True)  
plt.show()
```

GitHub: <https://github.com/Rey-ui/chiselni-metody>