

Лабораторна робота № 14

Використання бібліотек Python в наукових розрахунках. Елементи аналізу даних.

Мета роботи: ознайомлення з засобами супроводу науково-технічної діяльності з використанням програмування на Python.

Короткі теоретичні відомості

Jupyter Notebook - це веб-додаток з відкритим вихідним кодом, що надає функціонал для створення та поширення документів (блокнотів), що містять фрагменти коду, рівняння, графічну візуалізацію та текст. Jupyter Notebook супроводиться спільнотою розробників під назвою Project Jupyter.

Jupyter Notebooks - це проєкт, що виокремився з проєкту IPython, який раніше мав власний проєкт IPython Notebook. Назва Jupyter походить від основних підтримуваних мов програмування, які він підтримує: Julia, Python та R. Jupyter постачається з ядром IPython, яке дозволяє писати програми на Python, але наразі існує понад 100 інших ядер, які ви також можете використовувати.

Щоб використовувати Jupyter Notebook його можна встановити на комп'ютер. Процес встановлення аналогічний до встановлення пакетів Python (див. <https://jupyter.org/install>). Можна також використовувати одну з онлайн-версій, поданих на сторінці <https://jupyter.org/try>.

Jupyter Notebook входить також до дистрибутиву програми для аналізу даних Anaconda, який також можна встановити на власний ПК. Anaconda - це дистрибутив мов програмування Python та R для наукових обчислень (наука про дані, програми машинного навчання, обробка великих масивів даних, предиктивна аналітика тощо), який має на меті спростити керування пакетами та їх розгортанням. Дистрибутив містить пакети для науки про дані, придатні для Windows, Linux та macOS. Він розроблений і підтримується компанією Anaconda, Inc., яку заснували Пітер Ванг і Тревіс Оліфант у 2012 р.

Anaconda має також онлайн-версію, використання якої вимагає простої реєстрації за допомогою e-mail адреси. Доступна вона за посиланням з головної сторінки проєкту, або за адресою <https://www.anaconda.com/code-in-the-cloud>.

Завдання. Написати Python-код для розв’язування поданих нижче задач відповідно до варіанта. Розв’язування задач оформити у вигляді блокнота Jupiter з блоками, які формулюють умову задачі, містять код її розв’язування та відображають розв’язок а також блоки з поясненням до отриманого результату, де це потрібно.

1. Побудувати графік заданої функції $f(x)$ на заданому відрізку $[a; b]$. Словесно описати поведінку функції на цьому відрізку, а також вказати кількість та приблизні значення коренів рівняння $f(x) = 0$.

№	Функція	Інтервал для x
1	$y = -x^2 + 4x - 3$	$[-5, 5]$
2	$y = \frac{x^3}{3}$	$[-30, 30]$
3	$y = 2^x$	$[-3, 3]$
4	$y = e^{\frac{x}{2}}$	$[-3, 3]$
5	$y = \frac{8}{x}$	$[1, 80]$
6	$y = e^{-x^2}$	$[-30, 30]$
7	$y = \lg(x - 4)$	$[5, 10]$
8	$y = 4ch \frac{x}{4}$	$[-40, 40]$
9	$y = \frac{2x}{x^2 + 1}$	$[-60, 60]$
10	$y = \frac{10}{x^2 + 1}$	$[-15, 15]$
11	$y = \frac{1}{x^2}$	$[1, 20]$
12	$y = \operatorname{tg} x$	$[0, 2\pi]$
13	$y = \frac{e^x + e^{-x}}{2}$	$[-10, 10]$
14	$y = \frac{e^x - e^{-x}}{2}$	$[-10, 10]$
15	$y = x^4 - 3x^2 + 1$	$[-2, 2]$

2. Побудувати графік функції двох змінних. Змінна x належить інтервалу $[-\pi; \pi]$, змінна y належить інтервалу $[-2\pi; 2\pi]$.

№	Функція	№	Функція
1	$z = \frac{x^2}{25} + \frac{y^2}{25}$	9	$z = 8\cos(x + y) + 2\sin(x + y)$
2	$z = \frac{\sin x}{x} \times \frac{\sin y}{y}$	10	$z = \cos x / \sin y + 5(\cos 6x + \sin 2y)$
3	$z = 5\cos x \times \sin y$	11	$z = x \times \sin(x + y)$
4	$z = \sqrt{x^2 + y^2}$	12	$z = \frac{x^2}{16} + \frac{y^2}{25}$
5	$z = 8\cos(x + y)$	13	$z = 50\cos(x + y) \times \sin(x + y)$
6	$z = e^{xy}$	14	$z = x^5 + y^4 + xy$
7	$z = 16\cos(x + y) + \sin xy$	15	$z = \frac{x^2}{25} - \frac{y^2}{25}$
8	$z = 8\cos(x + y) + xy$		

3. Дослідити систему лінійних алгебраїчних рівнянь на сумісність. Якщо система сумісна та має єдиний розв'язок, знайти його. Якщо ні – вивести відповідне повідомлення:

$$\begin{cases} 3,81x_1 + 0,25x_2 + 1,28x_3 + (0,75 + \alpha)x_4 = 4,21 \\ 2,25x_1 + 1,32x_2 + (4,48 + \alpha)x_3 + 0,49x_4 = 6,47 + \alpha \\ 5,31x_1 + (6,28 + \alpha)x_2 + 0,98x_3 + 1,04x_4 = 2,38 \\ (9,39 + \alpha)x_1 + 2,45x_2 + 3,35x_3 + 2,28x_4 = 10,48 + \alpha \end{cases},$$

$\alpha = 0,5k$, де k – номер варіанта.

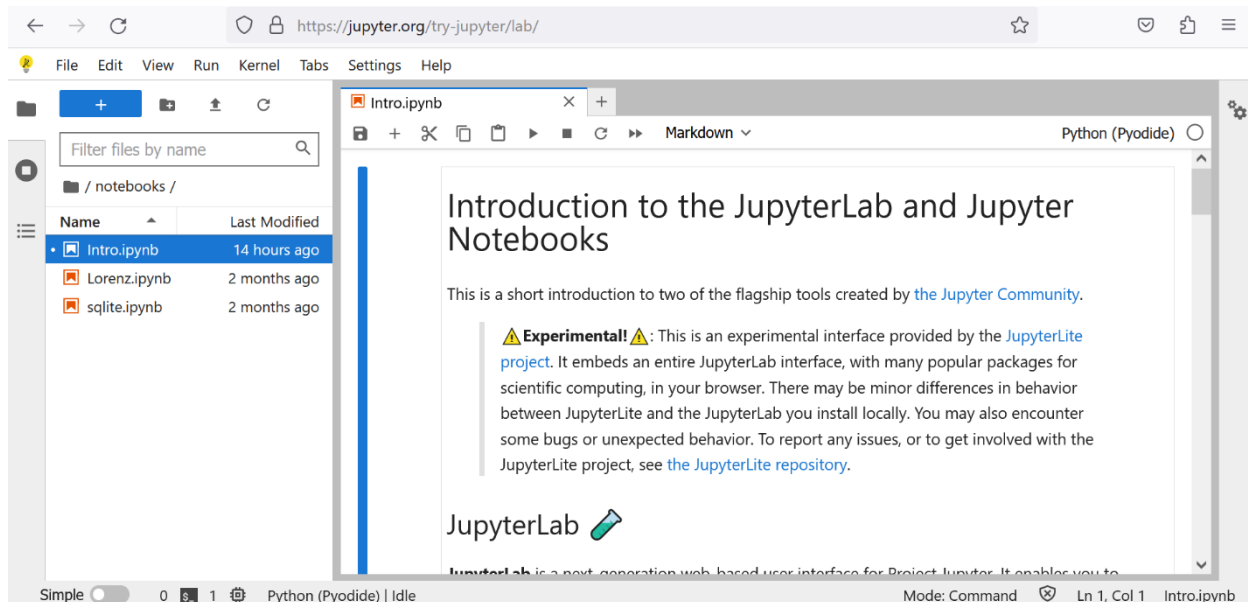
4. Побудувати діаграму розсіювання з лінією регресії для випадкових вибірок x , та y , згенерованих за правилом $x = F()$, $y = G() + n * x$. Варіанти розподілів подано в таблиці. Числові параметри розподілів задати самостійно.

№ вар.	F()	G()	№ вар.	F()	G()
1	Нормальний розподіл	Біноміальний розподіл	9	Нормальний розподіл	Рівномірний розподіл
2	Рівномірний розподіл	Нормальний розподіл	10	Експоненціальний розподіл	Нормальний розподіл
3	Експоненціальний розподіл	Біноміальний розподіл	11	Нормальний розподіл	Нормальний розподіл
4	Нормальний розподіл	Рівномірний розподіл	12	Рівномірний розподіл	Експоненціальний розподіл

5	Експоненціальний розподіл	Біноміальний розподіл	13	Нормальний розподіл	Експоненціальний розподіл
6	Рівномірний розподіл	Експоненціальний розподіл	14	Біноміальний розподіл	Рівномірний розподіл
7	Експоненціальний розподіл	Рівномірний розподіл	15	Рівномірний розподіл	Нормальний розподіл
8	Рівномірний розподіл	Рівномірний розподіл			

Приклад виконання завдання 1. Варіант № 15.

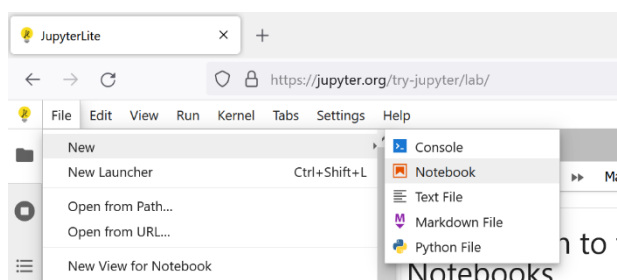
Використовувати Jupiter Notebooks можна online в одному з сервісів, рекомендованих за адресою <https://jupyter.org/try>



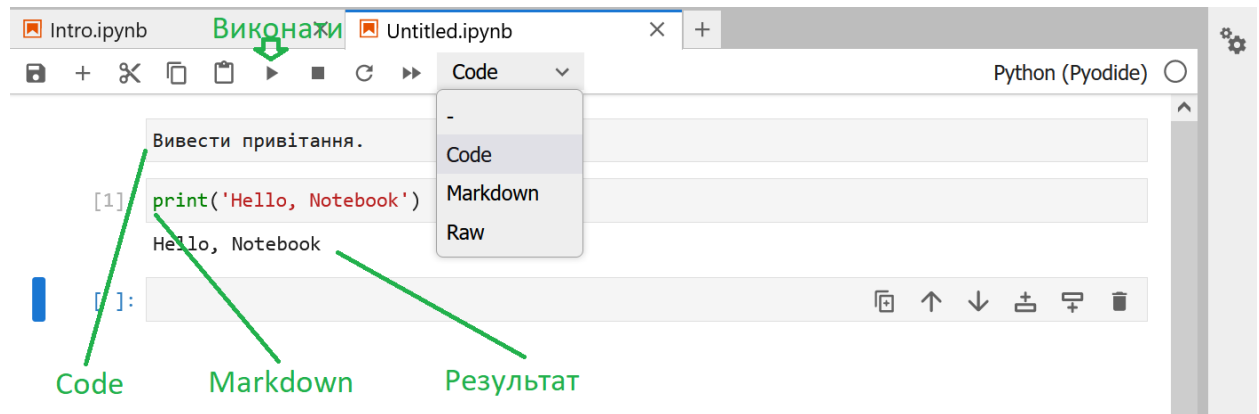
В прикладі далі для створення блокнота використано онлайн-версію пакету Anaconda (<https://www.anaconda.com/code-in-the-cloud>), після чого був кнвертований в html-документ локально встановленою версією цього ж пакету.

Для початку роботи створюємо новий блокнот командою File => New => Notebook з меню програми.

Далі обираємо ядро (інтерпретатор для виконання коду).



Для запису умов задач та коду їх розв'язування до блокнота додаємо комірки відповідного типу: для коду – code, для умов задач – markdown або Raw. Щоб відобразити результат роботи коду потрібно виконати одну або декілька комірок з кодом, натиснувши відповідну кнопку на панелі інструментів.



Орієнтовний вміст блокнота для розв'язування задач (варіант 15):

Завдання 1. Побудувати графік заданої функції $f(x)$ на заданому відрізку $[a; b]$. Словесно описати поведінку функції на цьому відрізку, а також вказати кількість та приблизні значення коренів рівняння $f(x) = 0$.

In [7]:

```
from numpy import exp, arange
from pylab import
meshgrid, cm, imshow, contour, clabel, colorbar, axis, title, show

# 100 linearly spaced numbers
x = np.linspace(-2, 2, 100)

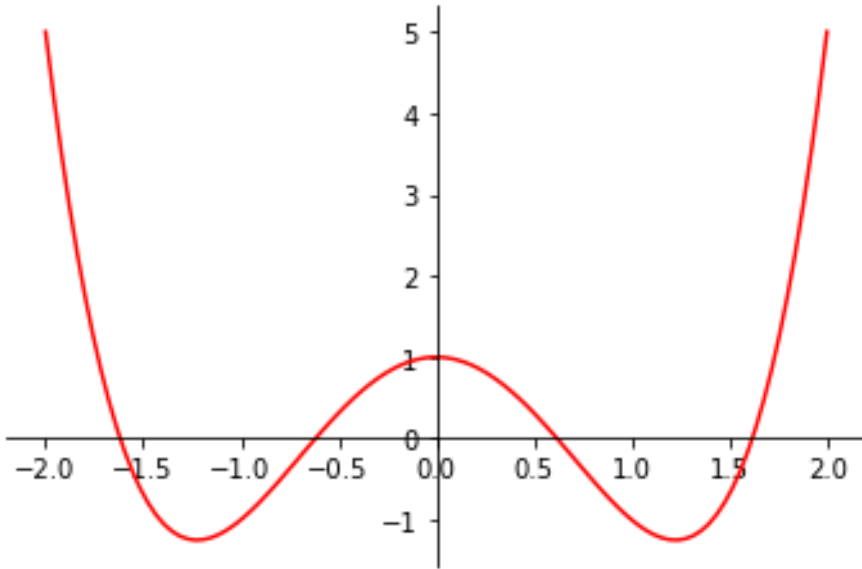
# the function, which is  $y = x^2$  here
y = x**4 - 3*x**2 + 1

# setting the axes at the centre
fig = plt.figure()
ax = fig.add_subplot(1, 1, 1)
ax.spines['left'].set_position('center')
ax.spines['bottom'].set_position('zero')
ax.spines['right'].set_color('none')
ax.spines['top'].set_color('none')
ax.xaxis.set_ticks_position('bottom')
ax.yaxis.set_ticks_position('left')

# plot the function
```

Лабораторний практикум з програмування на Python

```
plt.plot(x, y, 'r')  
  
# show the plot  
plt.show()
```



На заданому відрізку функція є парною та не є монотонною, - вона має два проміжки спадання та два - зростання. Графік функції чотири рази перетинає вісь, тобто рівняння $f(x) = 0$ на заданому відрізку має 4 корені, які приблизно дорівнюють -1,65, -0,65, 0,65 та 1,65

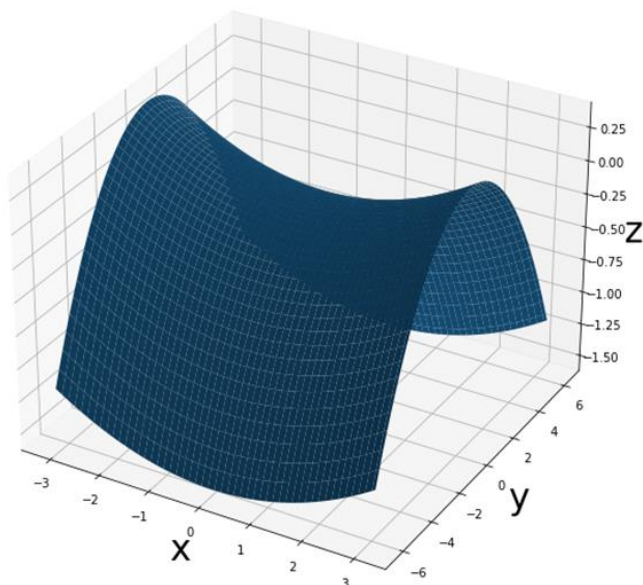
Завдання 2. Побудувати графік функції двох змінних $z = x^2/25 - y^2/25$. Інтервал зміни змінної x $[-\pi; \pi]$, змінна y належить інтервалу $[-2\pi; 2\pi]$.

In [7]:

```
# Import libraries  
from mpl_toolkits import mplot3d  
import numpy as np  
import matplotlib.pyplot as plt  
  
# Creating dataset  
x = np.outer(np.linspace(-np.pi, np.pi, 100),  
np.ones(100))  
y = np.outer(np.linspace(-2 * np.pi, 2 * np.pi, 100),  
np.ones(100)).T # transpose  
z = (x **2 /25 - y **2/25 )  
  
# Creating figure  
fig = plt.figure(figsize =(10, 10))  
ax = plt.axes(projection ='3d')  
ax.set_xlabel('x', fontsize=30)  
ax.set_ylabel('y', fontsize=30)  
ax.set_zlabel('z', fontsize=30)
```

```
# Creating plot
ax.plot_surface(x, y, z)

# show plot
plt.show()
```



In [30]:

```
import numpy as np
import numpy.linalg as la

A = np.array([[3.81, 0.25, 1.28, 8.25],
              [2.25, 1.32, 11.98, 0.49],
              [5.31, 13.78, 0.98, 1.04],
              [16.89, 2.45, 3.35, 2.28]])
b = np.array([4.21, 13.97, 2.38, 17.98])
r0 = la.matrix_rank(A)
bm = np.matrix(b).T.copy()
AB = np.append(A.copy(), bm, axis=1)
r1 = la.matrix_rank(AB)
print(f'Ранг основної матриці {r0}, ранг розширеної {r1}')
if r0 == 4 & r1 == 4:
    x = np.linalg.solve(A, b)
    print(f"Розв'язок системи {x}")
elif r0 == r1:
    print("Система має безліч розв'язків.")
else:
    print('Система несумісна.')
Ранг основної матриці 4, ранг розширеної 4
```

Лабораторний практикум з програмування на Python

Розв'язок системи [0.9045467 -0.24431655 1.02556343
-0.05914727]

In [2]:

```
from matplotlib import pyplot as plt
import numpy as np

# Створюємо генератор випадкових чисел
rng = np.random.default_rng()

# Генеруємо дані з відповідним розподілом
x = rng.uniform(0,20,size=100)
y = rng.normal(0,10,size=100) + 15*x

# отримуємо об'єкт для побудови графіка
fig, ax = plt.subplots(figsize = (9, 9))

# додаємо бульбашкову діаграму
ax.scatter(x, y, s=60, alpha=0.7, color="y",
edgecolors="b")

# Fit linear regression via least squares with numpy.polyfit
# It returns an slope (b) and intercept (a)
# deg=1 means linear fit (i.e. polynomial of degree 1)
b, a = np.polyfit(x, y, deg=1)

# Create sequence of 100
numbers from 0 to 100
xseq =
np.linspace(0, 20,
num=100)

# Plot regression line
ax.plot(xseq, a + b
* xseq, color="k",
lw=2.5)

plt.show()
```

