

oom_3_1_1_graphic

3 січня 2023 р.

0.1 # 3. Розв'язування нелінійних рівнянь: Лабораторний практикум

Нехай маємо рівняння

$$(1) \quad f(x) = 0, \quad x \in [a, b],$$

де $f : [a, b] \rightarrow \mathbb{R}$ – неперервна на $[a, b]$ функція.

Чисельне розв'язування такого рівняння складається з двох етапів:

1. локалізація коренів, тобто пошук числових проміжків, на яких є тільки один корінь цього рівняння;
2. обчислення коренів з наперед заданою точністю на кожному з виділених проміжків.

0.2 ### 3.1.1.2 Графічна локалізація коренів рівняння

Для побудови графіка функції **f** скористаємося бібліотекою **matplotlib**. Найпростіше це виконати в інтерактивному режимі, який встановлюють командою **%matplotlib widget**. У цьому випадку після виконання функції **plot_graphics**, яка побудує потрібний графік, треба виконати такі дії: 1. активізувати режим **Zoom to rectangle** за допомогою іконки меню на графічній панелі (яка містить зображення прямокутника) 2. на графічній панелі визначити мишкою якомога меншу прямокутну область, яка містить точку перетину графіка функції з віссю абсцис.

Виконання пункту 2 автоматично приведе до відображення визначеної прямокутної області на всю графічну панель. Виконуючи цей пункт кілька разів, можна локалізувати нуль функції на відрізку **[a, b]** з потрібною точністю на тій самій панелі.

Якщо режим **widget** не встановлено, то функцію **plot_graphics** виконують кілька разів, поспідовно уточнюючи координати відрізка **[a, b]** поки не локализують корінь з потрібною точністю.

0.3 #### Пояснення до використання програмного коду

- Підготувати потрібні функції :
 1. виконати комірку, де виконується підготовка середовища (в JupyterLab розкоментувати рядок з **%matplotlib widget**, щоб мати змогу масштабувати графік при локалізації у п.4)
 2. виконати комірку з визначенням функції **plot_graphics**
 3. виконати комірку з визначенням функції **f** (цю комірку виконувати кожного разу після внесення змін у визначення функції **f**)
- Локалізувати потрібний корінь рівняння :
 1. виконати комірку, в якій задається відрізок

2. виконати комірку з викликом функції `plot_graphics` для побудови графіка
3. уточнити (звужити) відрізок `[a, b]`, щоб на ньому знаходився лише один корінь рівняння
4. якщо в середовищі доступний інтерактивний режим (наприклад, в JupyterLab), то активізувати `Zoom to rectangle` (вибрати в меню на графічній панелі іконку з прямокутником) і виділити мишкою прямокутну область з потрібним фрагментом графіка
5. пункти 1 - 3 можна повторити для точнішої локалізації потрібного кореня

```
[1]: # Підготовка середовища
# при виконанні в JupyterLab наступний рядок розкоментувати
#%%matplotlib widget

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

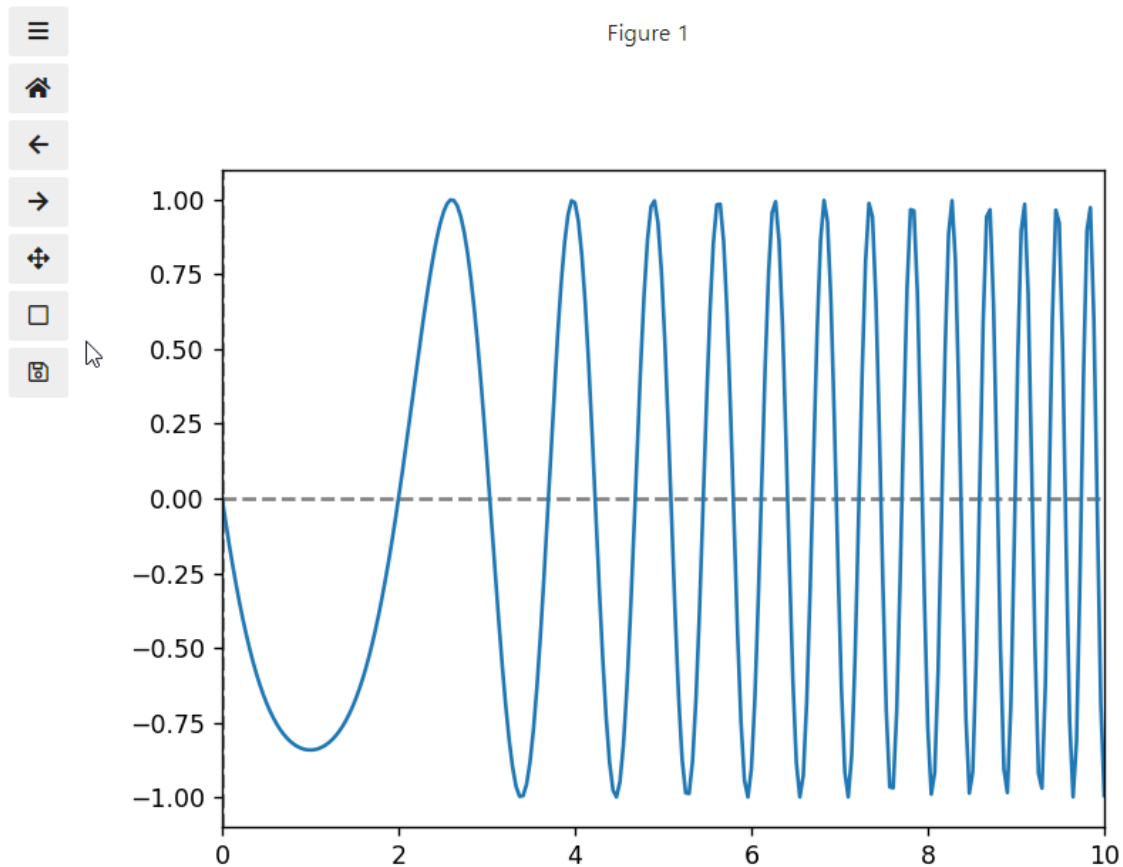
```
[2]: def plot_graphics(f, a, b, n):
    """функція для побудови графіка функції f
    на відрізку [a,b] за значеннями в n точках
    """
    xarr = np.linspace(a, b, n)
    y=f(xarr)
    fig = plt.figure()
    ax = fig.gca()
    ax.plot(xarr,y)
    ax.axhline(color="grey", ls="--", zorder=-1)
    ax.axvline(color="grey", ls="--", zorder=-1)
    ax.set_xlim(a,b)
    plt.show()
```

Приклад 1. Локалізувати другий додатний корінь рівняння (1), якщо $f(x) = \sin(x^2 - 2x)$

```
[3]: def f(x):
    """функція лівої частини рівняння (1)"""
    return np.sin(x*x-2*x)
```

```
[4]: # задання відрізка
a=0
b=10
```

```
[5]: plot_graphics(f, a, b, 256)
```

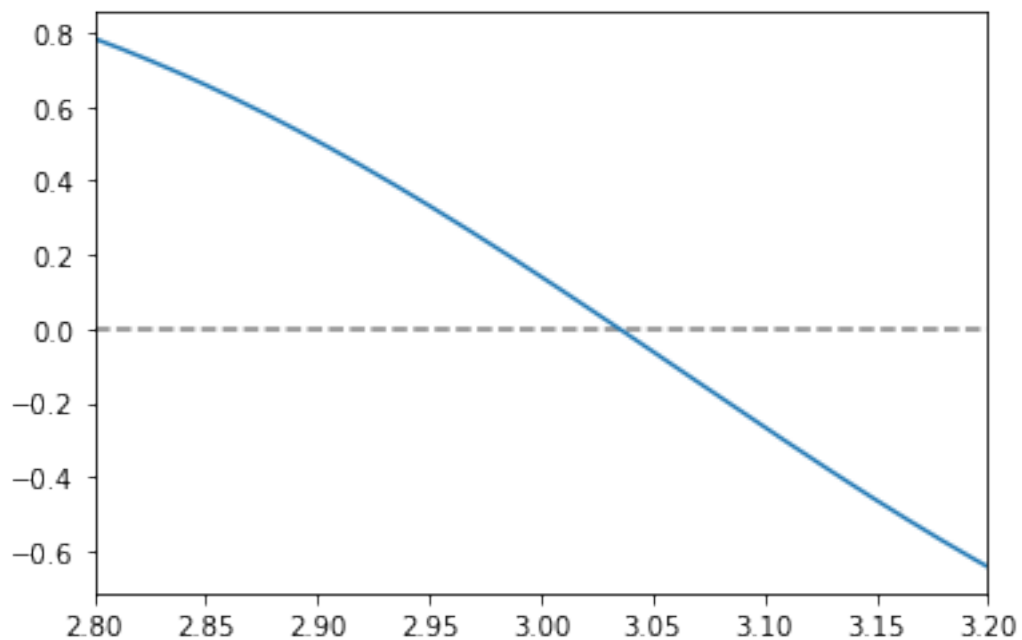


В інтерактивному режимі на графічній панелі використовуємо в меню іконку **Zoom to rectangle** і кілька разів звужуємо відрізок, на якому перемальовуватиметься графік.

Без використання масштабування можна кілька разів явно задати нові значення кінців відрізка і повторно побудувати графік:

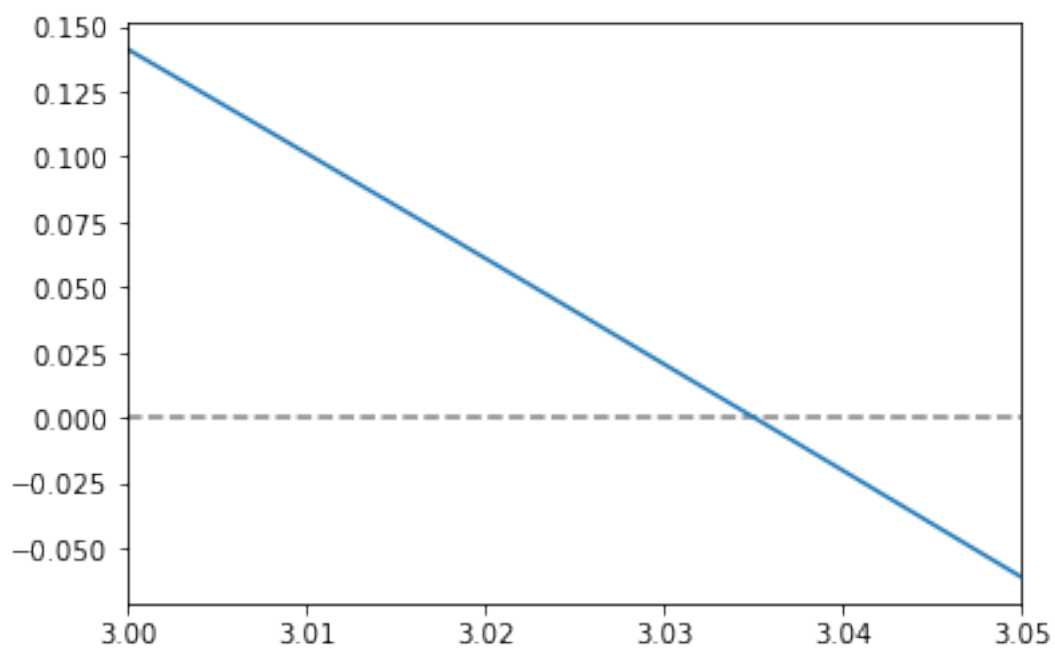
```
[6]: # задання оновленого відрізка  
a=2.8  
b=3.2
```

```
[7]: plot_graphics(f, a, b, 256)
```



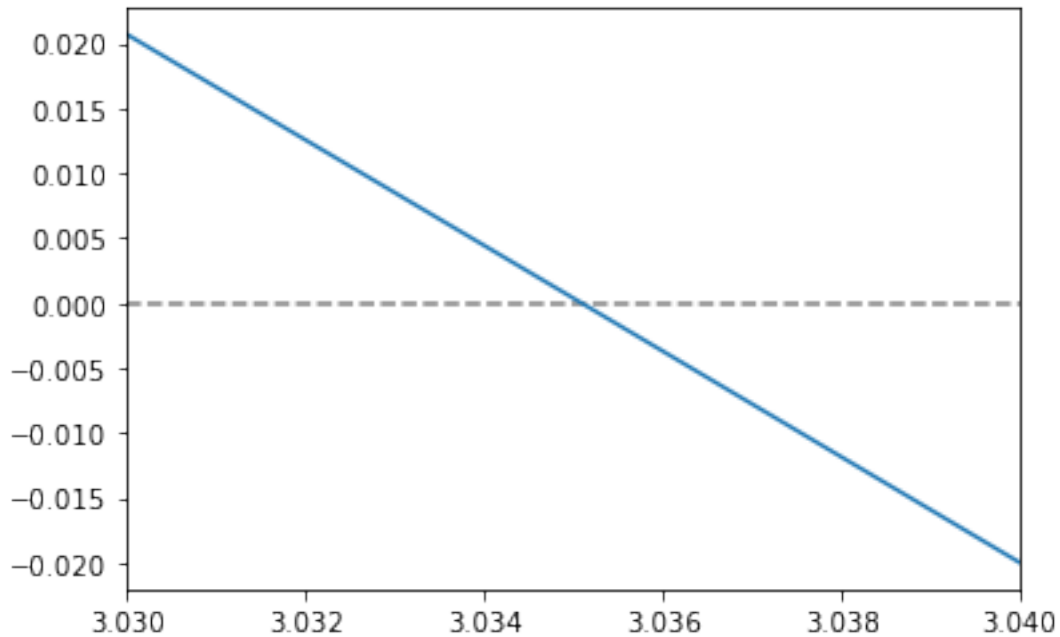
```
[8]: # задання оновленого відрізка
a=3.0
b=3.05
```

```
[9]: plot_graphics(f, a, b, 256)
```



```
[10]: # задання оновленого відрізка
a=3.03
b=3.04
```

```
[11]: plot_graphics(f, a, b, 256)
```



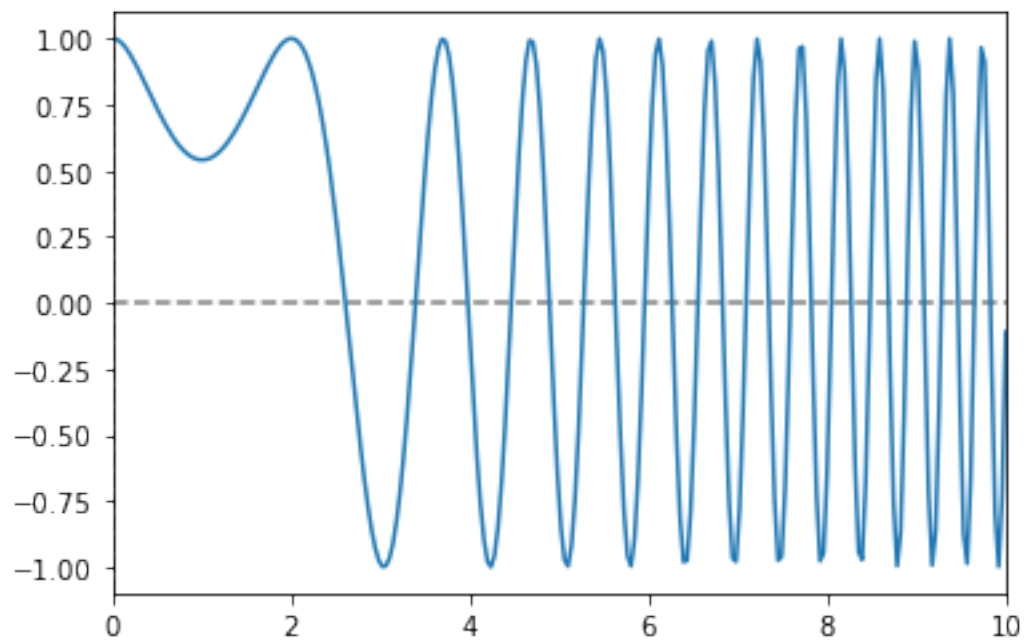
Висновок. Другий додатний корінь рівняння знаходиться на відріжку $[3.034, 3.036]$.

Приклад 2. Локалізувати другий додатний корінь рівняння (1), якщо $f(x) = \cos(x^2 - 2x)$

```
[12]: def fcs(x):
        """функція лівої частини рівняння (1)"""
        return np.cos(x*x-2*x)
```

```
[13]: # задання відрізка
a=0
b=10
```

```
[14]: plot_graphics(fcs, a, b, 256)
```



Висновок. Другий додатний корінь рівняння знаходиться на відрізку $[3.389, 3.390]$.

```
[15]: plt.close('all')
```