



Corso Pratico Python

[Modulo 1- 2026]

- 3. Matplotlib -
- 3.1. Plots
- 3.2. Esempi e casi d'uso

Richiedente: Dott. Roberto Felici [CNR-ISM] • Insegnante: Dott. Scigè J. Liu [INAF-IAPS]

Fondamenti di Matplotlib: La Gerarchia degli Oggetti

Matplotlib è la libreria di Python per visualizzazione dati. È fondamentale comprendere che il grafico è costruito da una **gerarchia di oggetti**.

1. L'Architettura "Artist"

Ogni elemento visibile sul grafico è un "Artist". La struttura si divide in due componenti principali:

- **Figure:** Il contenitore di primo livello (la "tela" o la finestra intera). Può contenere più grafici (Axes).
- **Axes:** Il singolo grafico o "sotto-grafico". È qui che risiedono i dati, le etichette e le scale.

2. Anatomia di un Grafico (Elementi Chiave)

Per comunicare i dati in modo efficace, ogni grafico deve includere : _____ ↗

Elemento	Funzione Matplotlib	Descrizione
Title	<code>plt.title()</code>	Titolo del grafico.
Axis Labels	<code>plt.xlabel()</code> , <code>plt.ylabel()</code>	grandezze e unità (es. $\text{\$s\$}$, $\text{\$V\$}$, $\text{\$kg\$}$).
Legend	<code>plt.legend()</code>	Identifica le serie dati
Grid	<code>plt.grid()</code>	Facilitazione alla lettura secondo gli assi.
Ticks	<code>plt.xticks()</code> , <code>plt.yticks()</code>	segni di graduazione della scala numerica.

I Due Approcci alla Programmazione

Esistono due modi per interagire con Matplotlib:

1. Interfaccia Pyplot (Stato implicito): Ideale per script rapidi e analisi interattive (es. `plt.plot()`).

2. Interfaccia Object-Oriented (Stato esplicito): Consigliata per progetti complessi e personalizzazioni avanzate (es. `fig, ax = plt.subplots()`).

💡 **Best Practice:** Utilizza sempre nomi chiari per le etichette e, dove possibile, la formattazione $\text{\$LaTeX\$}$ per le formule matematiche e le unità di misura, migliorando la leggibilità scientifica.

3 Plots



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

# 1. Generazione dei dati (Seno e Coseno)
x = np.linspace(0, 4, 100)
y_seno = np.sin(x * np.pi / 2)
y_coseno = np.cos(x * np.pi / 2)

# 2. Creazione della figura e del grafico
plt.figure(figsize=(10, 6))

# Linea Dati A (Seno) - Blu con marcatori circolari
plt.plot(x, y_seno, label='Dati Sensore A (Seno)', color='blue', marker='o', markevery=5)

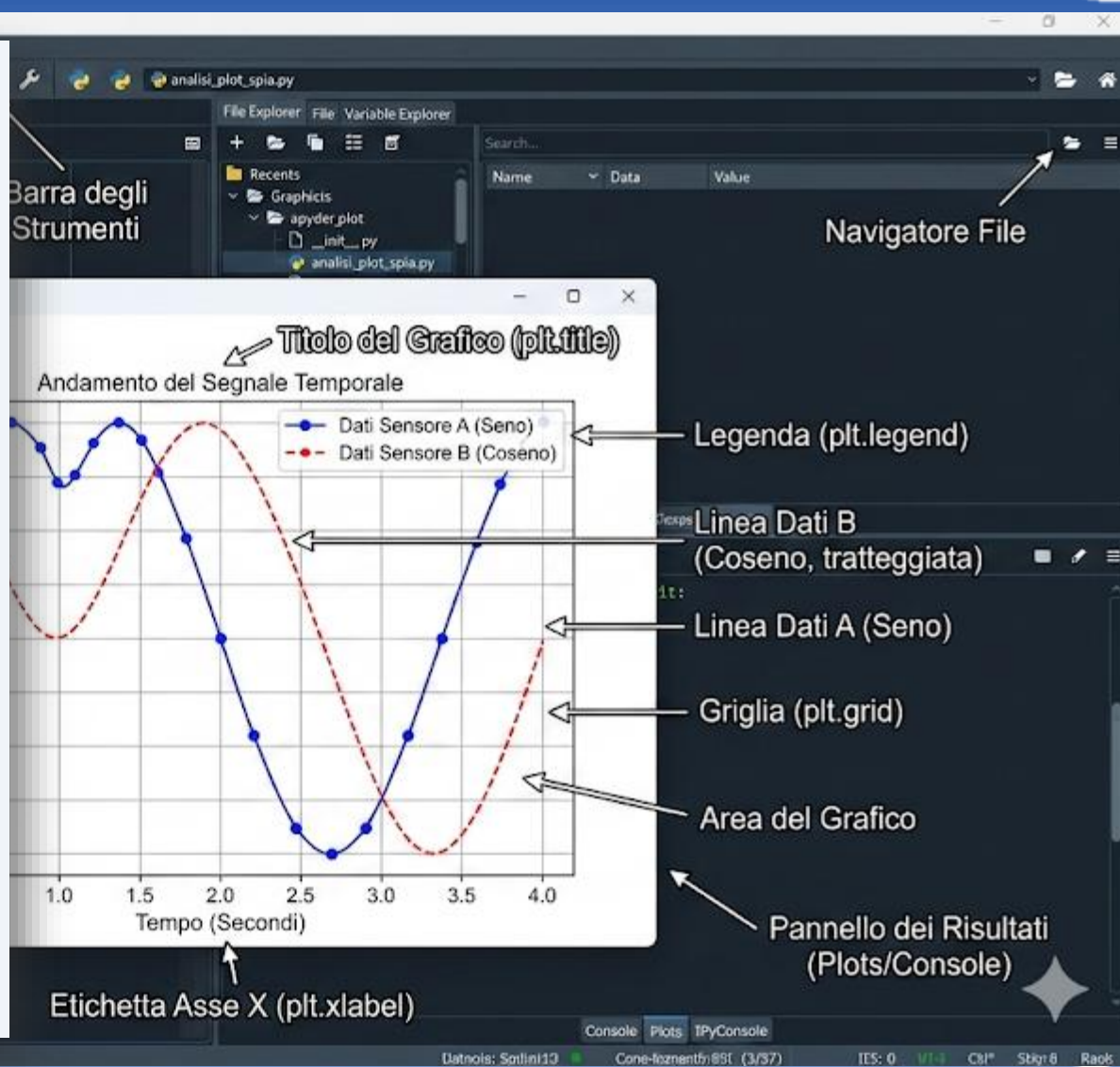
# Linea Dati B (Coseno) - Rossa tratteggiata
plt.plot(x, y_coseno, label='Dati Sensore B (Coseno)', color='red', linestyle='--')

# 3. Etichettatura degli elementi grafici
plt.title('Andamento del Segnale Temporale') # Titolo del Grafico
plt.xlabel('Tempo (Secondi)')                # Etichetta Asse X
plt.ylabel('Ampiezza (Volt)')                 # Etichetta Asse Y

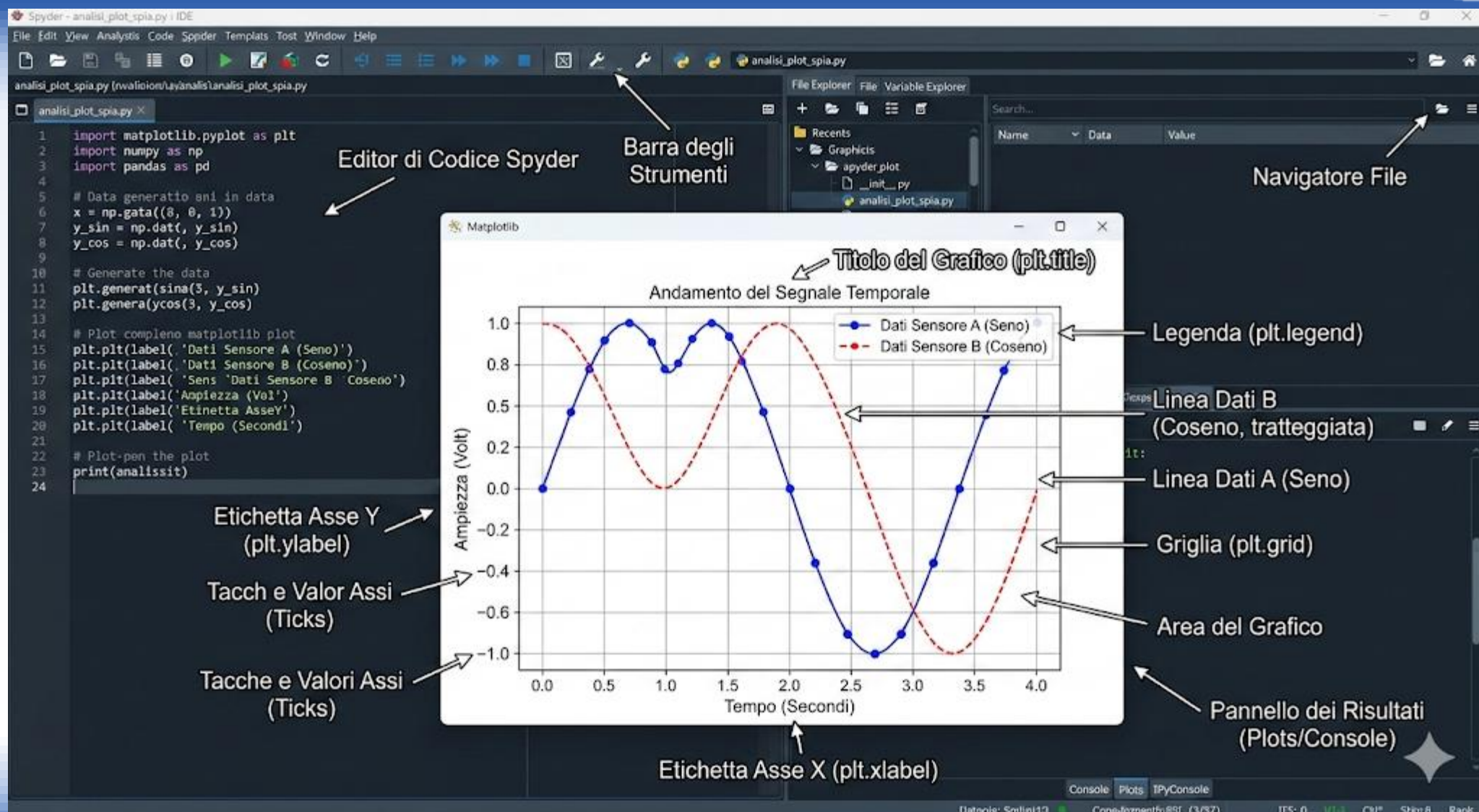
# 4. Personalizzazione degli assi e della griglia
plt.grid(True, linestyle='-', alpha=0.7)      # Griglia (plt.grid)
plt.legend(loc='upper right')                 # Legenda (plt.legend)

# Definizione manuale dei Ticks (Tacche e Valori Assi) per precisione
plt.xticks(np.arange(0, 4.5, 0.5))
plt.yticks(np.arange(-1.0, 1.2, 0.2))

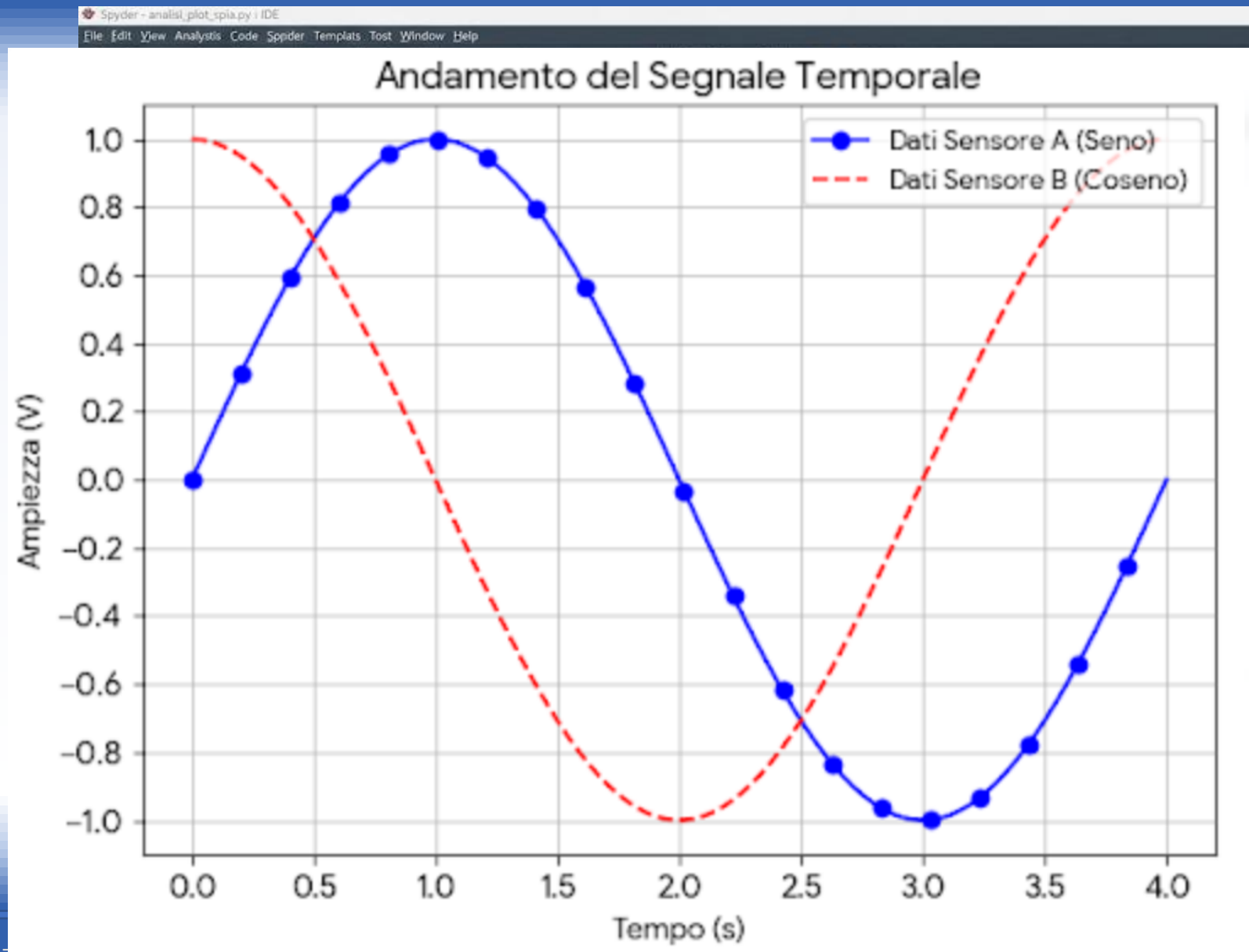
# 5. Visualizzazione
plt.tight_layout()
plt.show()
```



3 Approfondimenti



3 Plots



3 Plots

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

# 1. Generazione dei dati (Seno e Coseno)
x = np.linspace(0, 4, 100)
y_seno = np.sin(x * np.pi / 2)
y_coseno = np.cos(x * np.pi / 2)

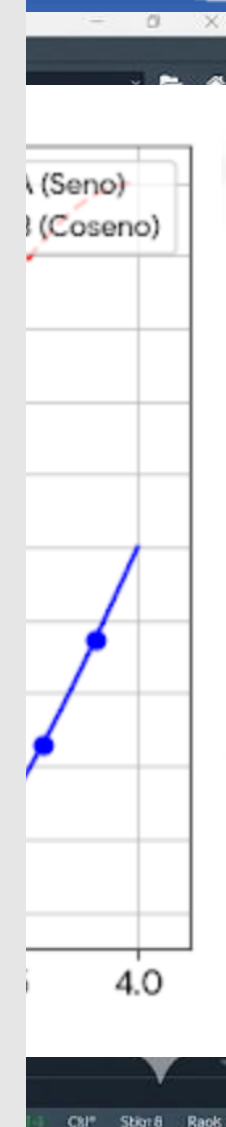
# 2. Definizione del grafico (seno e coseno)
plt.plot(x, y_seno, label='Dati Sensore A (Seno)', color='blue', marker='o', markevery=5)
plt.plot(x, y_coseno, label='Dati Sensore B (Coseno)', color='red', linestyle='--')

# 3. Etichettatura degli elementi grafici con notazione LaTeX
plt.title('Andamento del Segnale Temporale')
plt.xlabel('Tempo ( $s$ )')          # Tempo in secondi
plt.ylabel('Ampiezza ( $V$ )')        # Ampiezza in Volt

# 4. Personalizzazione degli assi e della griglia
plt.grid(True, linestyle='-', alpha=0.7)
plt.legend(loc='upper right')

# Definizione delle tacche (ticks) per gli assi
plt.xticks(np.arange(0, 4.5, 0.5))
plt.yticks(np.arange(-1.0, 1.2, 0.2))

# 5. Salvataggio e visualizzazione
plt.tight_layout()
plt.savefig('spyder_plot.png')
```





3 Plots



```
import matplotlib.pyplot as plt
import datetime as dt

# Dati temporali
date = [dt.datetime(2024, 1, i) for i in range(1, 11)]
valori = [10, 15, 7, 25, 20, 28, 12, 18, 22, 30]

plt.plot(date, valori, marker='o', linestyle='-')
plt.gcf().autofmt_xdate() # Ruota le date per evitare sovrapposizioni
plt.title("Analisi Serie Temporale")
plt.ylabel("Valore ($u.a.$)")
plt.show()
```

```
x = np.linspace(1, 100, 100)
y = np.exp(x / 20)

plt.semilogy(x, y, color='green') # Scala logaritmica solo su Y
plt.grid(True, which="both", ls="--", alpha=0.5)
plt.title("Risposta in Scala Logaritmica")
plt.ylabel("Scala  $\log_{10}$ ")
plt.show()
```

```
x = np.linspace(0, 10, 100)
y_mean = np.sin(x)
error = 0.2

plt.plot(x, y_mean, 'b-', label='Media')
plt.fill_between(x, y_mean - error, y_mean + error, color='b', alpha=0.2, label='Incertezza  $\pm 0.2$ ')
plt.legend()
plt.title("Visualizzazione dei Range")
plt.show()
```

INAF
Osservatorio
di Astrofisica





3 Plots



```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Creazione griglia dati
x = np.linspace(-5, 5, 50)
y = np.linspace(-5, 5, 50)
X, Y = np.meshgrid(x, y)
Z = np.sin(np.sqrt(X**2 + Y**2))

fig = plt.figure(figsize=(10, 7))
ax = fig.add_subplot(111, projection='3d')

# Generazione superficie
surf = ax.plot_surface(X, Y, Z, cmap='viridis', edgecolor='none')

ax.set_title("Superficie 3D:  $z = \sin(\sqrt{x^2 + y^2})$ ")
fig.colorbar(surf, ax=ax, shrink=0.5, aspect=5) # Barra del colore
plt.show()

plt.show()
```

```
z = np.linspace(1, 100, 100)
x = x / 20
```

```
ax.plot(x, y, color='green') # Scala logaritmica solo su Y
ax.set_ylabel('y', which="both", ls="--", alpha=0.5)
ax.set_xlabel('Risposta in Scala Logaritmica')
ax.set_zlabel('Scala  $\log_{10}$ ')

```

50

label='Incertezza ± 0.2 ')



3 Plots



```
import matplotlib.pyplot as plt
import datetime as dt
from mpl_toolkits.mplot3d import Axes3D

# Dati temporali
date = [dt.datetime(2010, 1, 1), dt.datetime(2010, 1, 2), dt.datetime(2010, 1, 3)]
valori = [10, 15, 7, 2]

plt.plot(date, valori)
plt.gcf().autofmt_xdate()
plt.title("Analisi Serie Temporale")
plt.ylabel("Valore ($u$)")
plt.show()

x = np.linspace(0, 10, 100)
y_mean = np.sin(x)
error = 0.2

plt.plot(x, y_mean)
plt.fill_between(x, y_mean - error, y_mean + error)
plt.legend()
plt.title("Visualizzazione con errore")
plt.show()

x = np.linspace(-5, 5, 50)
y = np.linspace(-5, 5, 50)
X, Y = np.meshgrid(x, y)
Z = np.sin(np.sqrt(X**2 + Y**2))

fig = plt.figure(figsize=(10, 7))
ax = fig.add_subplot(111, projection='3d')

# Generazione superficie
surf = ax.plot_surface(X, Y, Z, cmap='viridis', edgecolor='none')

ax.set_title("Superficie 3D: $z = \sin(\sqrt{x^2 + y^2})$")
fig.colorbar(surf, ax=ax, shrink=0.5, aspect=5) # Barra del colore
plt.show()
```

```
x = np.linspace(1, 100, 100)
```

```
ax.set_yscale('log') # Scala logaritmica solo su Y
ax.plot(x, y, ls="--", alpha=0.5)
ax.set_xlabel("logaritmica")
plt.show()
```

```
ax.set_xlabel("$\pm 0.2$")
```



3 Plots

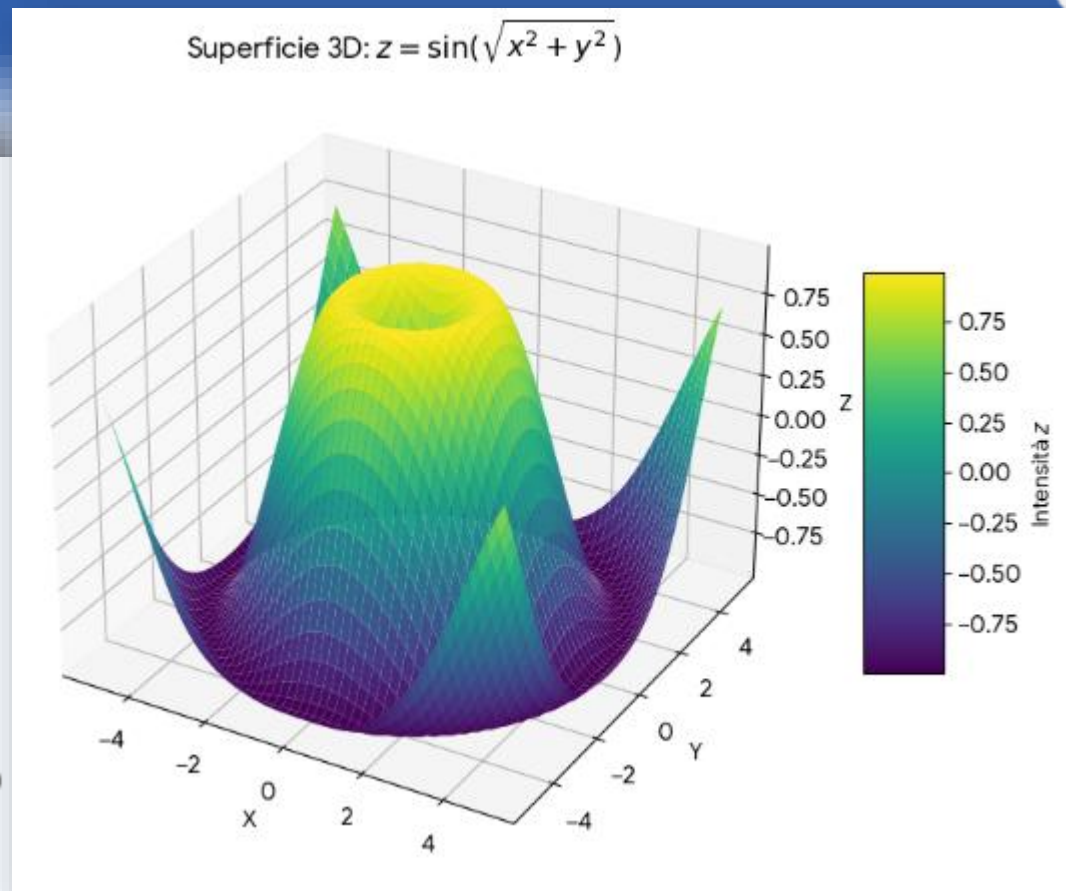
```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

# Creazione griglia dati
x = np.linspace(-5, 5, 50)
y = np.linspace(-5, 5, 50)
X, Y = np.meshgrid(x, y)
Z = np.sin(np.sqrt(X**2 + Y**2))

fig = plt.figure(figsize=(10, 7))
ax = fig.add_subplot(111, projection='3d')

# Generazione superficie
surf = ax.plot_surface(X, Y, Z, cmap='viridis', edgecolor='none')

ax.set_title("Superficie 3D:  $z = \sin(\sqrt{x^2 + y^2})$ ")
fig.colorbar(surf, ax=ax, shrink=0.5, aspect=5) # Barra del colore
plt.show()
```



3 Plots

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D

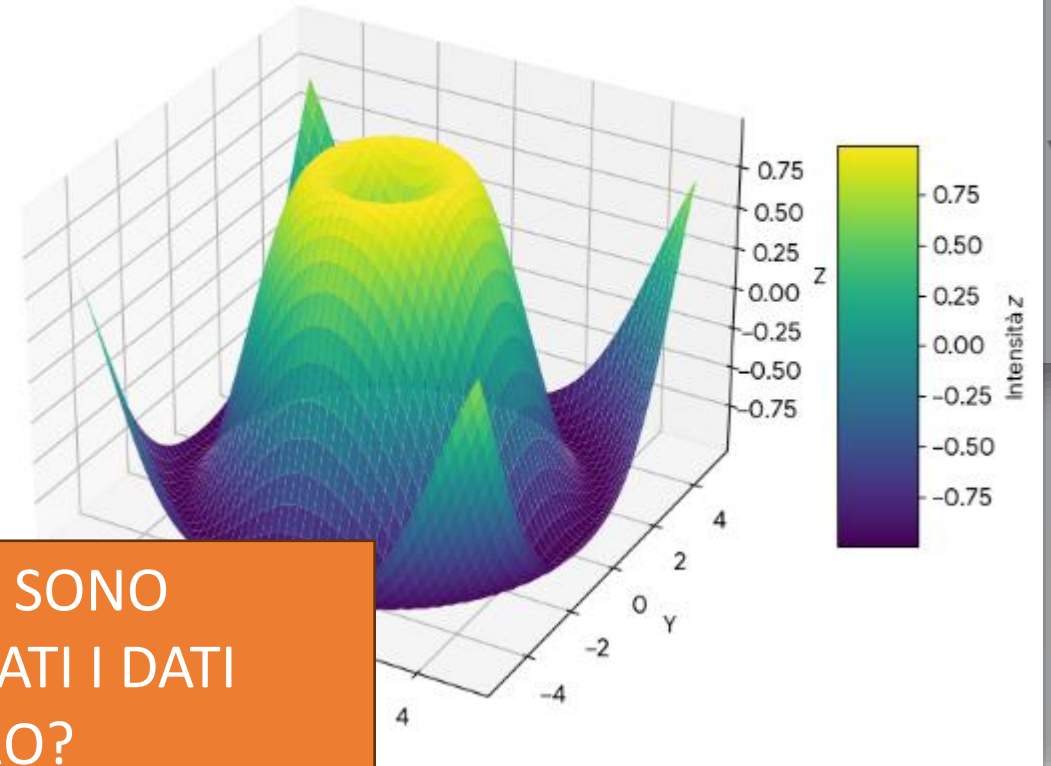
# Creazione griglia dati
x = np.linspace(-5, 5, 50)
y = np.linspace(-5, 5, 50)
X, Y = np.meshgrid(x, y)
Z = np.sin(np.sqrt(X**2 + Y**2))

fig = plt.figure(figsize=(10, 7))
ax = fig.add_subplot(111, projection='3d')

# Generazione superficie
surf = ax.plot_surface(X, Y, Z, cmap='viridis')

ax.set_title("Superficie 3D:  $z = \sin(\sqrt{x^2 + y^2})$ ")
fig.colorbar(surf, ax=ax, shrink=0.5, aspect=1)
plt.show()
```

Superficie 3D: $z = \sin(\sqrt{x^2 + y^2})$



MA COME SONO
RAPPRESENTATI I DATI
DENTRO?

ESERCIZIO :
VISUALIZZARE I VALORI