

Aula 4 – Introdução a matplotlib

EMC410235 - Programação Científica para Engenharia e Ciência Térmicas

Prof. Rafael F. L. de Cerqueira

2025.2

Objetivos da Aula

- Apresentar os principais recursos de visualização gráfica da biblioteca `matplotlib`.
- Explorar a criação de gráficos cartesianos: `plt.plot()`, `plt.scatter()` e `plt.hist()`.
- Customizar e salvar gráficos para uso em relatórios técnicos e científicos.

Por que usar Matplotlib?

```
conda install matplotlib  
...  
import matplotlib.pyplot as plt
```

- Biblioteca padrão para visualização científica em Python
- Permite criação de gráficos estáticos, animados e interativos
- Sintaxe inspirada no MATLAB, com grande flexibilidade
- Ideal para geração de gráficos científicos



<https://matplotlib.org/>

Galeria de Exemplos

[https://matplotlib.org/stable/
plot_types/index.html](https://matplotlib.org/stable/plot_types/index.html)

Gráfico de Linha com `plt.plot()`

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

x = np.linspace(-3, 3, 100)
y = x**2

plt.plot(x, y)
plt.xlabel('x')
plt.ylabel('y = x^2')
plt.title('Gráfico de y = x^2')
plt.grid()
plt.show()
```

Customização explícita de estilo com plt.plot()

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

x = np.linspace(-3, 3, 100)
y1 = x**2
y2 = x**3

plt.plot(x, y1, color='blue',
         linestyle='--', marker='o',
         linewidth=2, label='y = x^2
')
```

```
plt.plot(x, y2, color='green',
         linestyle=':', marker='s',
         linewidth=3, label='y =
x^3')

plt.xlabel('x')
plt.ylabel('y')
plt.title('Curvas com estilo
customizado')

plt.grid()
plt.legend()
plt.show()
```

Organização de múltiplos gráficos com `plt.subplot()`

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

```
x = np.linspace(-3, 3, 100)
y1 = x**2
y2 = x**3
```

```
plt.figure(figsize=(10, 4))
plt.subplot(1, 2, 1)
plt.plot(x, y1)
plt.title('y = x^2')
plt.grid()
```

```
plt.subplot(1, 2, 2)
plt.plot(x, y2, color='green')
plt.title('y = x^3')
plt.grid()
```

```
plt.tight_layout()
plt.show()
```

Visualização de dados com dispersão usando `plt.scatter()`

Quando usar `scatter`?

Use `plt.scatter()` para visualizar conjuntos de dados com:

- Variabilidade ponto a ponto (ex: dados experimentais com ruído)
- Distribuição espacial de eventos (ex: posições, partículas)
- Comparação com uma curva de referência

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

np.random.seed(42)

x = np.linspace(0, 10, 100)
y_ideal = 2 * x + 1

ruído = np.random.normal(loc=0.0,
                          scale=2.0, size=x.shape)

y_medido = y_ideal + ruído
```

```
plt.scatter(x, y_medido,
            color='blue', alpha=0.5,
            label='dados com ruído')

plt.plot(x, y_ideal, color='red',
         label='y = 2x + 1')

plt.xlabel('x')
plt.ylabel('y')
plt.title('Dispersão de dados com scatter')
plt.grid()
plt.legend()
plt.show()
```

Criação de múltiplas figuras nomeadas com plt.figure()

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

```
x = np.linspace(-3, 3, 100)
y1 = x**2
y2 = x**3
```

```
fig1 = plt.figure()
plt.plot(x, y1, color='blue')
plt.title('Figura 1:  $y = x^2$ ')
plt.xlabel('x')
plt.ylabel('y')
plt.grid()
```

```
fig2 = plt.figure()
plt.plot(x, y2, color='green')
plt.title('Figura 2:  $y = x^3$ ')
plt.xlabel('x')
plt.ylabel('y')
plt.grid()

plt.show()
```


Salvando uma figura com savefig()

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

```
x = np.linspace(-2, 2, 100)
y = x**2
```

```
plt.figure()
plt.plot(x, y)
plt.title('y = x^2')
plt.xlabel('x')
plt.ylabel('y')
plt.grid()
```

```
# Salva em alta resolucao
plt.savefig('grafico_x2.png', dpi
           =300)
```

```
# Exibe a figura na tela
plt.show()
```

Salvando múltiplas figuras com savefig()

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

```
x = np.linspace(-3, 3, 100)
y1 = x**2
y2 = x**3
```

```
fig1 = plt.figure()
plt.plot(x, y1, color='blue')
plt.title('Figura 1:  $y = x^2$ ')
plt.grid()
fig1.savefig('figura1.png', dpi
            =300)
```

```
fig2 = plt.figure()
plt.plot(x, y2, color='green')
plt.title('Figura 2:  $y = x^3$ ')
plt.grid()
fig2.savefig('figura2.png', dpi
            =300)

plt.show()
```

Formatos de saída para salvar gráficos

- **PNG**: formato raster (imagem pixelada), ideal para uso na web ou apresentações.
 - Aceita argumento `dpi=` para definir resolução (ex: `dpi=300` para alta qualidade).
 - **Não ideal** para documentos científicos impressos ou zoom em detalhes.
- **PDF e SVG**: formatos **vetoriais**.
 - Escalam sem perda de qualidade.
 - Indicados para artigos, relatórios técnicos e publicações.
 - Arquivos editáveis (ex: no Inkscape ou Illustrator).
- **EPS**: outro formato vetorial, comum em submissões para periódicos em LaTeX.
 - Mais antigo, mas ainda amplamente aceito.
 - Suporte limitado em ferramentas modernas.
- Use: `plt.savefig('figura.pdf')` ou `plt.savefig('grafico.png', dpi=300)`

Histograma (plt.hist()) com linha de media usando plt.axvline()

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

# Gera dados com media 2.0 e desvio
    0.5
dados = np.random.normal(loc=2.0,
    scale=0.5,
    size=1000)

media = np.mean(dados)

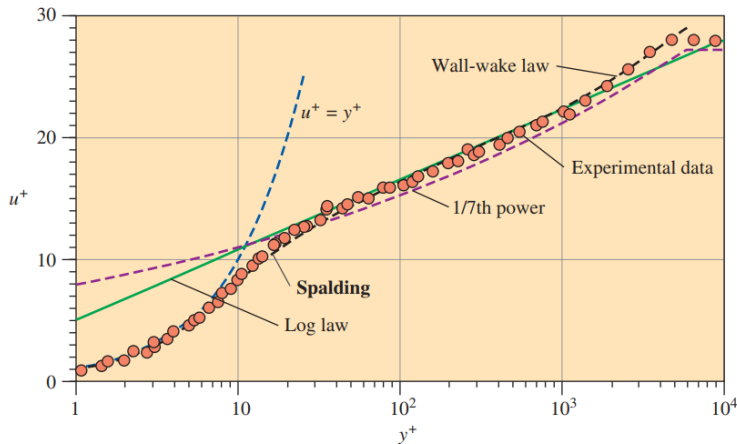
plt.hist(dados,
    bins=30,
```

```
    color='lightgray',
    edgecolor='black',
    density=True)
```

```
plt.axvline(media,
    color='red',
    linestyle='--',
    label='media')
```

```
plt.xlabel('valor')
plt.ylabel('densidade')
plt.title('Distribuicao normal com
    media destacada')
plt.legend()
plt.grid()
plt.show()
```

Exemplo de aplicação: Perfil de velocidade em uma C.L. turbulenta



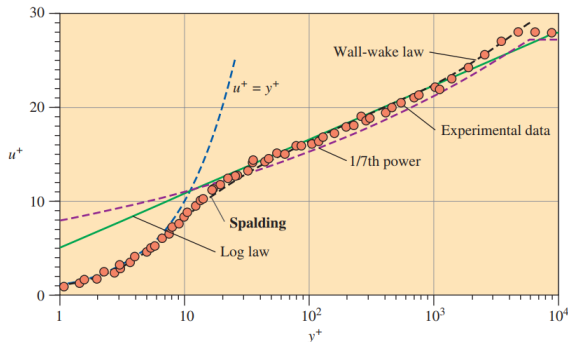
Definições:

- $u^+ = \frac{u}{u_*}$
- $y^+ = \frac{u_* y}{\nu}$
- $u_* = \sqrt{\frac{\tau_w}{\rho}}$

Onde:

- τ_w : tensão cisalhante na parede
- ρ : densidade do fluido
- ν : viscosidade cinemática

Exemplo de aplicação: Perfil de velocidade em uma C.L. turbulenta



Subcamada viscosa:

$$u^+ = y^+$$

Lei logarítmica (von Kármán):

$$u^+ = \frac{1}{\kappa} \ln y^+ + B \quad \text{com } \kappa = 0,40 \quad \text{e} \quad B = 5,0$$

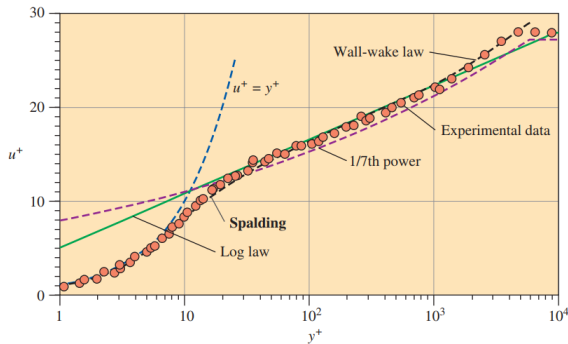
Lei da potencia (1/7):

$$\frac{u}{U_\infty} = \left(\frac{y}{\delta}\right)^{1/7}$$

Equação de Spalding:

$$y^+ = u^+ + e^{-\kappa B} \left(e^{\kappa u^+} - 1 - \kappa u^+ - \frac{(\kappa u^+)^2}{2!} - \frac{(\kappa u^+)^3}{3!} \right)$$

Exemplo de aplicação: Perfil de velocidade em uma C.L. turbulenta



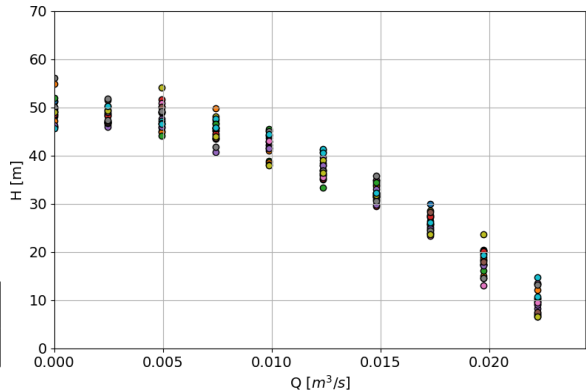
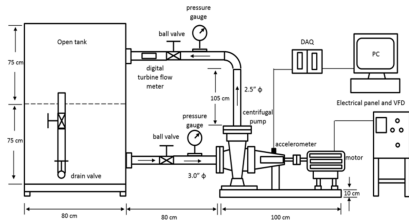
Como ficam os perfis de velocidade?

Exemplo:

- Ar a 25° ;
- $U_\infty = 30,8$ m/s;
- Para cálculo de τ_w :
 $Cf(x) = 0,059 / (Re_x)^{1/5}$;
- Perfis de velocidade na posição $x = 5$ m.

Exercício para Casa - Aplicação de Leis de Semelhança para Bombas

Realizados um total de 20 “experimentos” temos a seguinte relação entre carga líquida H e vazão Q .



Exercício para Casa - Aplicação de Leis de Semelhança para Bombas

Leitura de dados e aplicação das Leis de Semelhança para Bombas

Tarefas:

- 1 Ler os arquivos de cada experimento;
- 2 Criar a curva de carga média da bomba, relação Q vs. H – média das realizações;
- 3 Incluir também o desvio padrão em cada ponto;
- 4 Admitindo que os experimentos foram realizados a uma rotação de 1200 rpm, plotar a nova curva de Q vs. H para rotação de 2400 rpm;

$$\frac{Q_B}{Q_A} = \frac{\omega_B}{\omega_A} \left(\frac{D_B}{D_A} \right)^3 \quad \text{e} \quad \frac{H_B}{H_A} = \left(\frac{\omega_B}{\omega_A} \right)^2 \left(\frac{D_B}{D_A} \right)^2$$

- 5 Salvar em um arquivo de texto os dados da curva média Q , vs., H p/ situação de 1200 rpm e 2400 rpm.

Exercício para Casa - Aplicação de Leis de Semelhança para Bombas

Dicas:

- Para trabalhar com arquivos, pastas e afins, procure utilizar a `pathlib`
 - `from pathlib import Path`
- Para listar arquivos em uma pasta, utilize o pacote `os`
 - `os.listdir(pasta)`
- Para lidar com dados faltantes, utilize as funções do tipo `np.nanxxxx`

