

Introdução à Física Computacional - 7600017

Introdução ao Matplotlib

Um método moderno de gerar gráficos faz uso conjunto de duas bibliotecas Python: [NumPy](#), para ler, gerar ou manipular dados; e [Matplotlib](#), para gerar gráficos e animações. A documentação de ambos se encontra em seus respectivos sites. O Matplotlib em especial possui uma “infinidade” de exemplos em [Examples - Matplotlib](#). Para utilizar ambas as bibliotecas, é preciso primeiro instalá-las em um ambiente virtual (Virtual Environment ou *venv*). Isto é necessário pois não é recomendado instalar bibliotecas junto à versão Python do seu sistema, então criamos uma versão isolada do Python, um ambiente virtual, no qual podemos instalar o que quisermos.

I. ORGANIZAÇÃO DO PROJETO E AMBIENTE VIRTUAL

A. Diretório do Projeto

Antes de seguir as instruções abaixo, certifique-se de que você se está no diretório do seu projeto. Se o diretório ainda não existe, abra um terminal e crie-o da seguinte forma:

```
mkdir projeto1
```

seguido de

```
cd projeto1
```

Sempre execute este último comando e certifique-se que você se encontra no diretório correto.

B. Criando e Ativando o *venv*

Abra um terminal no diretório do seu projeto e utilize o comando:

```
python3 -m venv .venv
```

Isto cria um ambiente virtual chamado *.venv*. Este é geralmente o padrão, mas o nome pode ser qualquer um. Uma vez que criamos o ambiente virtual, **não precisamos criá-lo de novo. Contudo, sempre que formos instalar bibliotecas ou utilizar a versão do Python instalada no nosso ambiente virtual, precisamos ativá-lo.** Para isso, use o comando:

```
source .venv/bin/activate
```

C. Instalando Bibliotecas

Agora, com o ambiente virtual ativado, podemos instalar nossas bibliotecas:

```
pip install numpy matplotlib
```

II. LENDO DADOS DE ARQUIVOS E ESCRIVENDO UM SCRIPT

Crie um script no diretório do seu projeto chamado *grafico.py*. Nas primeiras linhas do seu script, coloque:

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

Assim, podemos usar as bibliotecas instaladas na seção anterior. Para importar dados de um arquivo *.dat* utilize:

```
x, y = np.loadtxt("dados.dat", unpack=True)
```

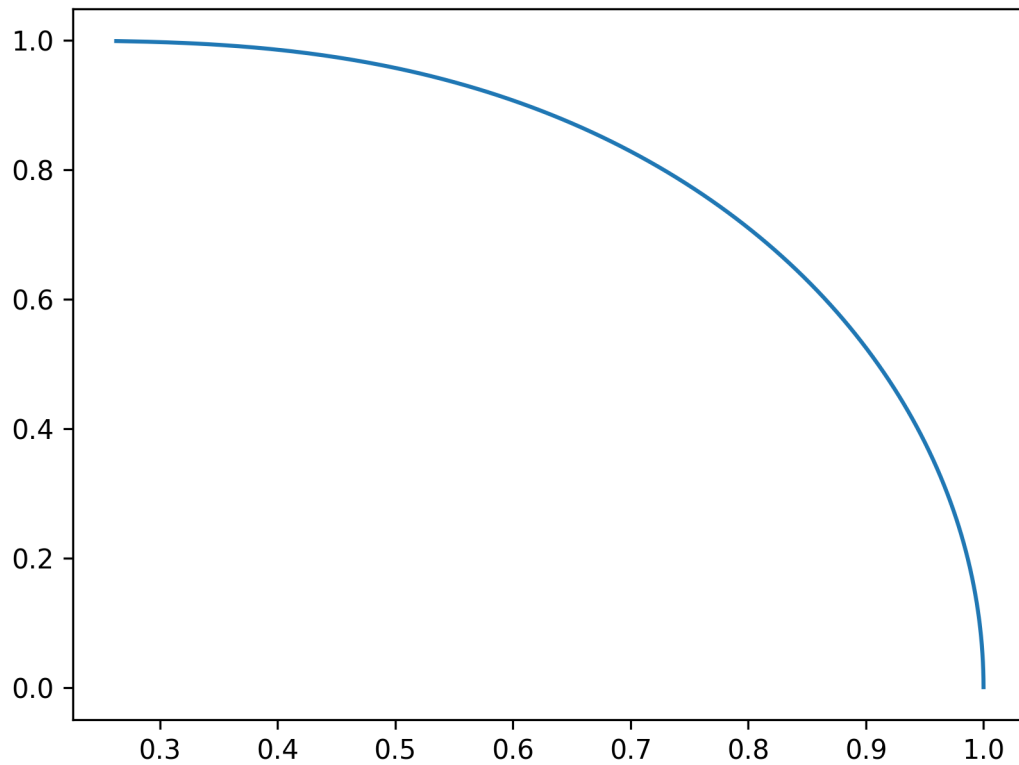
Nesta linha, *unpack=True* é necessário para interpretar o arquivo de entrada corretamente. Para gerar o gráfico, precisamos de no mínimo duas linhas:

```
plt.plot(x, y)
plt.show()
```

Então, no terminal e **com o ambiente virtual ativado**, executamos

```
python grafico.py
```

Para terminar o script, feche a janela com o gráfico. O resultado se encontra na figura abaixo.



Podemos customizar nosso gráfico colocando as seguintes linhas entre `plt.plot` e `plt.show`:

```
plt.xlabel('T')
plt.ylabel('m')
plt.xlim(0, 1)
plt.ylim(0, 1)
```

Podemos ainda, modificar as propriedades da linha gerada e adicionar uma legenda. Em `plt.plot`:

```
plt.plot(x, y, linewidth=3, linestyle='--', color='purple', label='Sem legenda')
```

Para salvar sua figura, adicione a seguinte linha logo antes de `plt.show()`

```
plt.savefig('grafico.png')
```

No Matplotlib existem vários estilos de gráficos pré-prontos. Para usá-los, use a seguinte linha logo após os imports:

```
plt.style.use('ggplot')
```

Após isso, temos um script `grafico.py` para gerar e salvar gráficos. Ele se parece com:

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

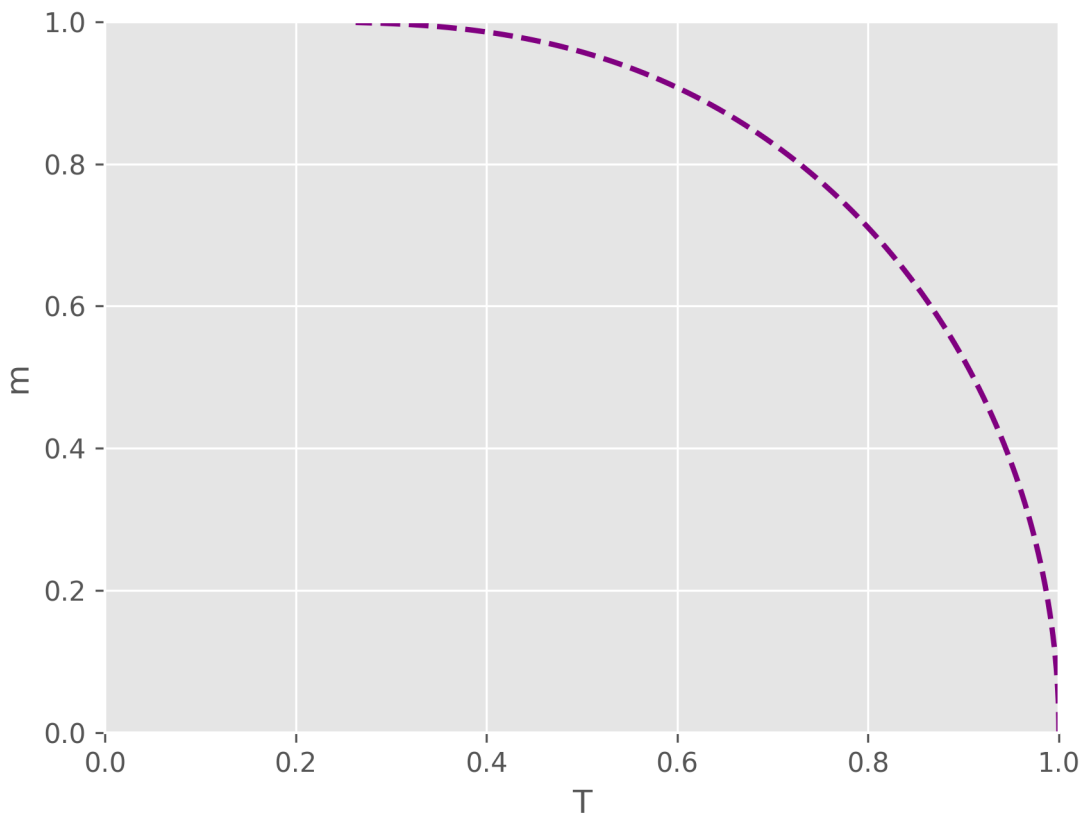
plt.style.use('ggplot')

x, y = np.loadtxt("dados.dat", unpack=True)

plt.plot(x, y, linewidth=2, linestyle='--', color='purple', label='Sem legenda')
plt.xlabel('T')
```

```
plt.ylabel('m')
plt.ylim(0, 1)
plt.xlim(0, 1)
plt.savefig("grafico.png")
plt.show()
```

Que nos gera o gráfico:



III. EXEMPLO: GERANDO DADOS COM NUMPY (FUNÇÃO SENO)

Com o NumPy, podemos utilizar suas funções matemáticas diretamente no script para gerar dados, em vez de ler de um arquivo:

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

plt.style.use('ggplot')

x = np.linspace(0, 2 * np.pi, 100)
y = np.sin(x)

plt.plot(x, y, linewidth=2, linestyle='-', color='orange', label='sen(x)')
plt.xlabel('x [radianos]')
plt.ylabel('y')
plt.legend()
plt.savefig("grafico_seno.png")
plt.show()
```

O resultado é o gráfico a seguir:

