

# Gráficos com NumPy

Python para Sala de Aula de Matemática

Luis Alberto D'Afonseca

CEFET-MG

# Conteúdo

Gerando Gráficos

# Matplotlib

- ▶ Biblioteca para gerar gráficos
- ▶ `pyplot` – estilo Matlab



<https://matplotlib.org>

# Bibliotecas

```
import numpy as np
```

```
from matplotlib import pyplot as plt
```

# Desenhar Linha Poligonal

Função `plot` desenha uma linha poligonal

- ▶ vetor com coordenadas  $x$
- ▶ vetor com coordenadas  $y$

```
x = [ 1, 2, 5, 6, 7 ]
```

```
y = [ 0, 1, 0, 2, 4 ]
```

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

# Desenhar o Gráfico de Uma Função

1. Avaliar os valores  $x$  desejados e criar o vetor  $x$
2. Avaliar a função nesses pontos e criar o vetor  $y$
3. Desenhar a linha poligonal ligando os pontos

# Desenhar o Gráfico de $f(x) = x^3$

```
x = np.linspace( -5, 5, 40 )
```

```
y = x**3
```

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

# Desenhar o Gráfico de $f(x) = \sin(x)$

```
x = np.linspace(0, 10, 100)
```

```
y = np.sin(x)
```

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



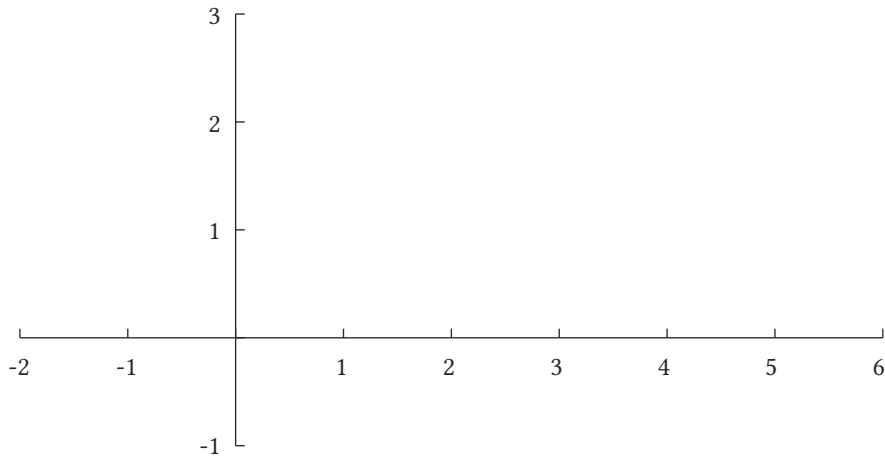
# Desenhar o Gráfico de $f(x) = e^{-x^2} \cos(x)$

```
x = np.linspace(-5, 5, 400)
y = np.exp( -x**2 ) * np.cos(9*x)
plt.plot( x, y )
```

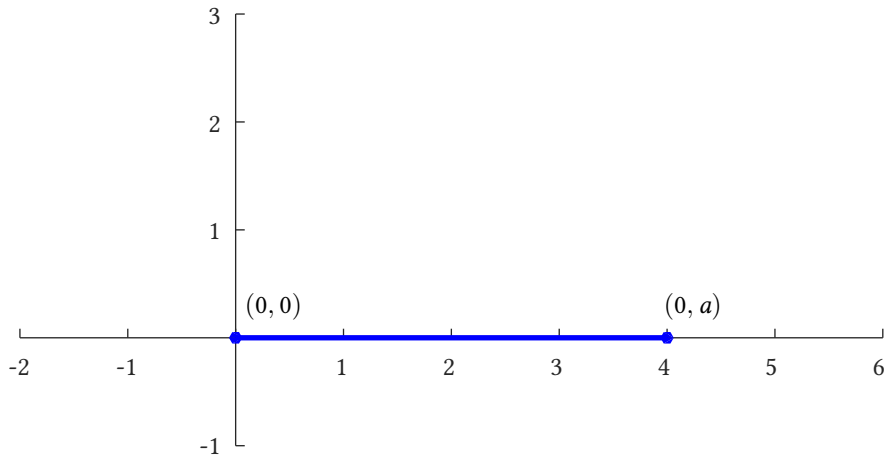
# Desenhar Um Triângulo Retângulo

1. Dadas as medidas dos catetos  $a$  e  $b$
2. Desenhar um triângulo retângulo no plano cartesiano

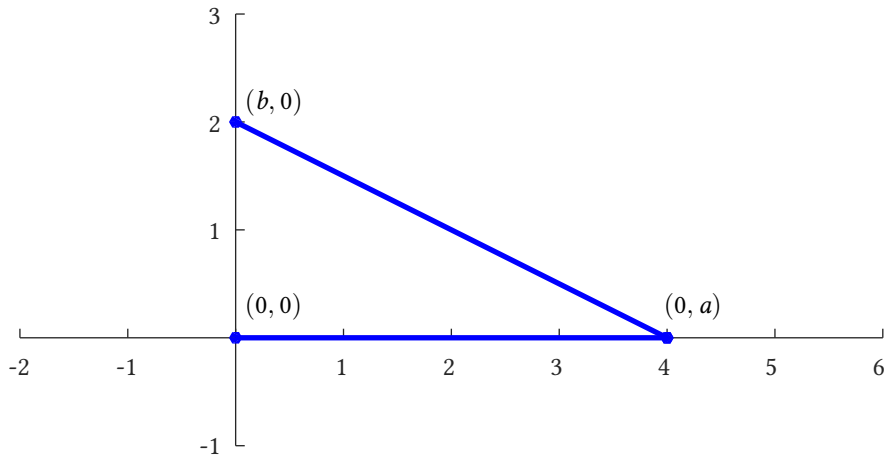
# Triângulo Retângulo



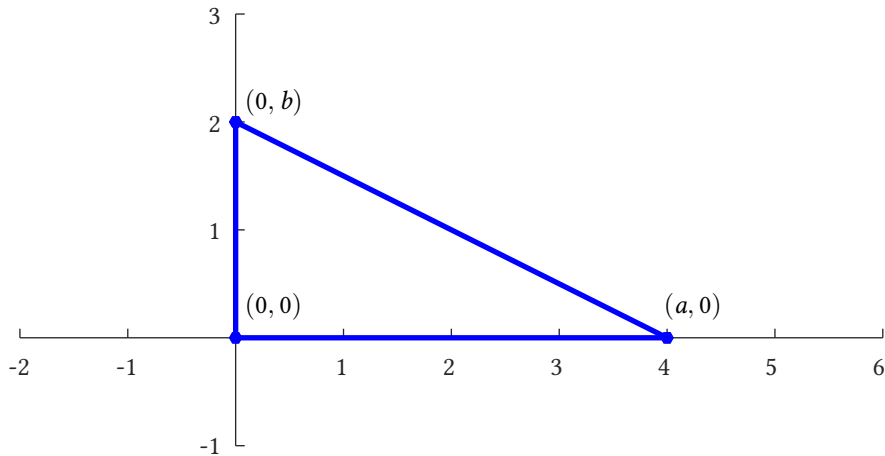
# Triângulo Retângulo



# Triângulo Retângulo



# Triângulo Retângulo



# Desenhar um Triângulo Retângulo

Dados  $a$ ,  $b$

Desenhar linha poligonal

$(0, 0)$

$(a, 0)$

$(0, b)$

$(0, 0)$

# Desenhar um Triângulo Retângulo

```
def triangulo(a, b):  
    x = [ 0, a, 0, 0]  
    y = [ 0, 0, b, 0]  
    plt.plot(x, y)
```



# Desenhar um Triângulo Retângulo

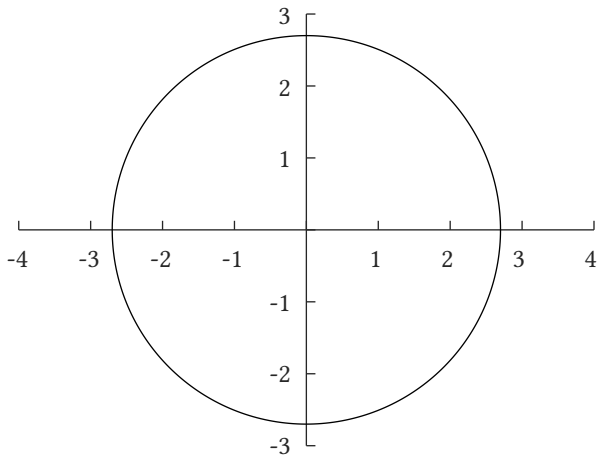
```
def triangulo(a, b):  
  
    x = [ 0, a, 0, 0]  
    y = [ 0, 0, b, 0]  
  
    plt.plot(x, y)  
    plt.gca().set_aspect('equal', adjustable='box')  
    plt.axis('off')  
    plt.draw()
```

# Desenhar Polígono Regular

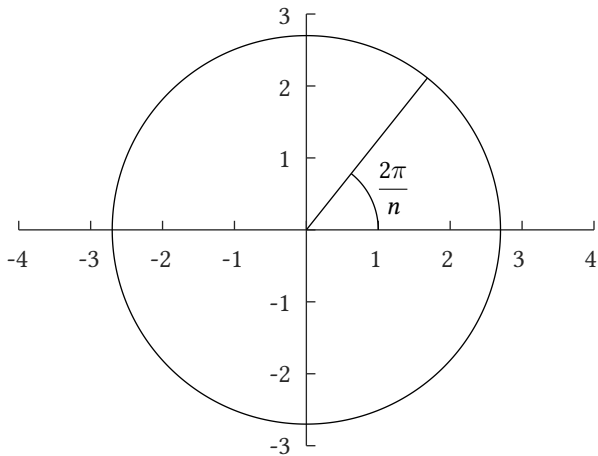
Desenhar o polígono regular de  $n$  lados circunscrito na circunferência de raio  $r$

1. Dividir o intervalo  $[0, 2\pi]$
2. Calcular as coordenadas  $(x, y)$  de cada vértice
3. Desenhar a linha poligonal ligando os vértices

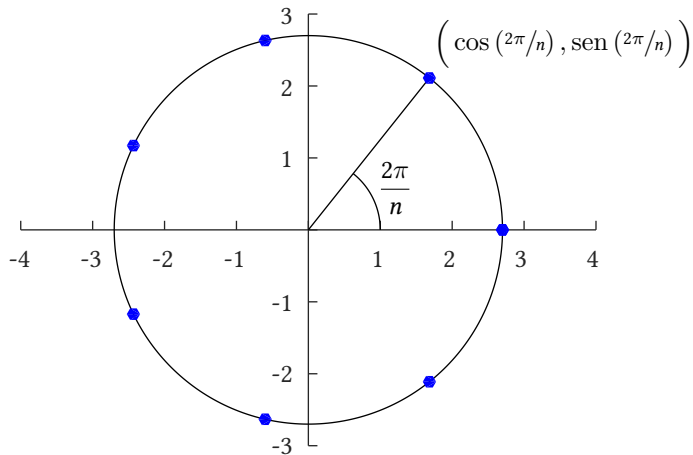
# Polígono – Circunferência



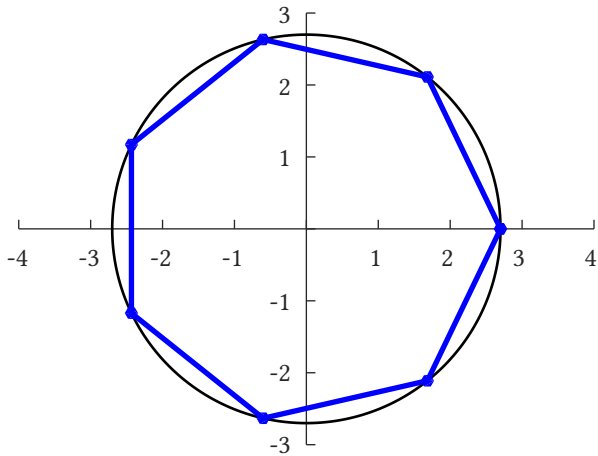
# Polígono – Divisão dos Ângulos



# Polígono – Coordenadas dos Vértices



# Polígono



# Desenhar Polígono Regular

```
def desenhar_poligono( n, r ):  
  
    t = np.linspace( 0, 2*np.pi, n+1 );  
    x = r * np.cos( t )  
    y = r * np.sin( t )  
  
    plt.plot( x, y )  
    plt.gca().set_aspect('equal', adjustable='box')  
    plt.axis('off')  
    plt.draw()
```

# Desenhar Polígono Regular

```
for k in range(4,20):  
    desenhar_poligono( k, k )
```