Nome: Victor Yaegashi Setti

RA: 206362

Importando as bibliotecas que serão usadas e definindo pi = π

In [328…
```
import numpy as np
import matplotlib.pyplot as plt
import scipy as sp
import pandas as pd
from IPython.display import Image, display

pi = np.pi
```

1) Filtro de Chebyshev

Copiando a função que calcula os coeficientes do polinômio de Chebyshev

In [329…
```
"""
Parâmetros: w - vetor de frequências (sugestão: usar um vetor com amostra
            wc - freq. de corte do filtro (em rad/s)
            n - ordem do filtro de Chebyshev
Saída:      Tn - vetor com os coeficientes calculados do polinômio de Che
"""

def coeficientes_chebyshev(w, wc, n):
    Tn = np.zeros((w.size,))
    Tn[abs(w) < wc] = np.cos(n*np.arccos(w[abs(w) < wc] / wc))
    Tn[abs(w) >= wc] = np.cosh(n*np.arccosh(w[abs(w) >= wc] / wc))
    return Tn
```

Implementando uma função para calcular o módulo da resposta em frequência para o filtro de Chebyshev

In [330…
```
def resposta_chebyshev(w, wc, n, e):
    Tn = coeficientes_chebyshev(w, wc, n)

    resposta = 1 / np.sqrt(1 + (e**2) * (Tn**2))

    return resposta
```

-> Item a)

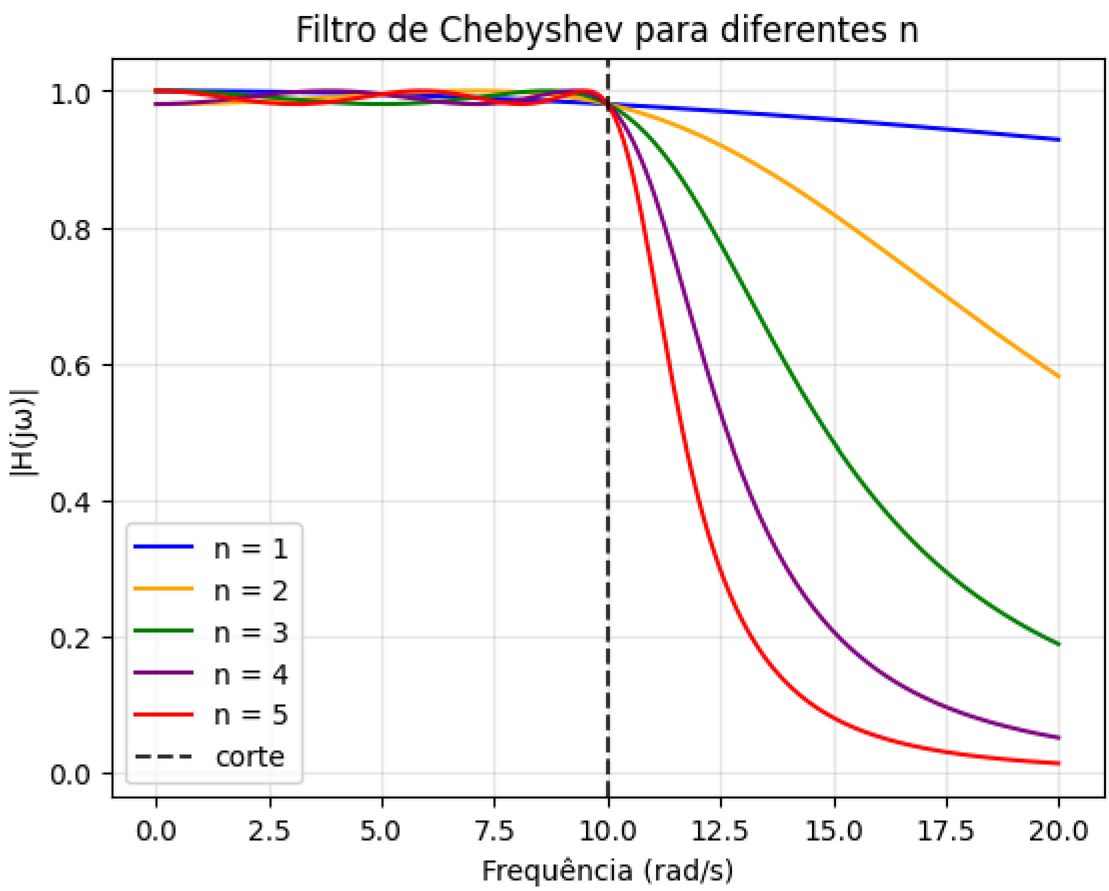
Fixando ω_c = 10 rad/s e ε = 0,2 e variando n de 1 a 5, obtemos as seguintes respostas em frequência

In [331…
```
w = np.linspace(0, 20, 200)

resposta1 = resposta_chebyshev(w, 10, 1, 0.2)
```

```
resposta2 = resposta_chebyshev(w, 10, 2, 0.2)
resposta3 = resposta_chebyshev(w, 10, 3, 0.2)
resposta4 = resposta_chebyshev(w, 10, 4, 0.2)
resposta5 = resposta_chebyshev(w, 10, 5, 0.2)

plt.plot(w, resposta1, label="n = 1", color="blue")
plt.plot(w, resposta2, label="n = 2", color="orange")
plt.plot(w, resposta3, label="n = 3", color="green")
plt.plot(w, resposta4, label="n = 4", color="purple")
plt.plot(w, resposta5, label="n = 5", color="red")
plt.xlabel("Frequência (rad/s)")
plt.ylabel("|H(jω)|")
plt.title(f"Filtro de Chebyshev para diferentes n")
plt.axvline(x=10, color="black", linestyle="--", alpha=0.8, label="corte"
plt.legend()
plt.grid(True, alpha=0.3)
plt.show()
```



Comentários:

Nota-se que a transição entre a banda passante e a banda de rejeição fica mais marcante conforme n aumenta, uma vez que as ondas de maior n caem para zero mais rapidamente após passarem pela frequência de corte ω_c = 10 rad/s. Isso representa uma maior seletividade do filtro.

-> Item b)

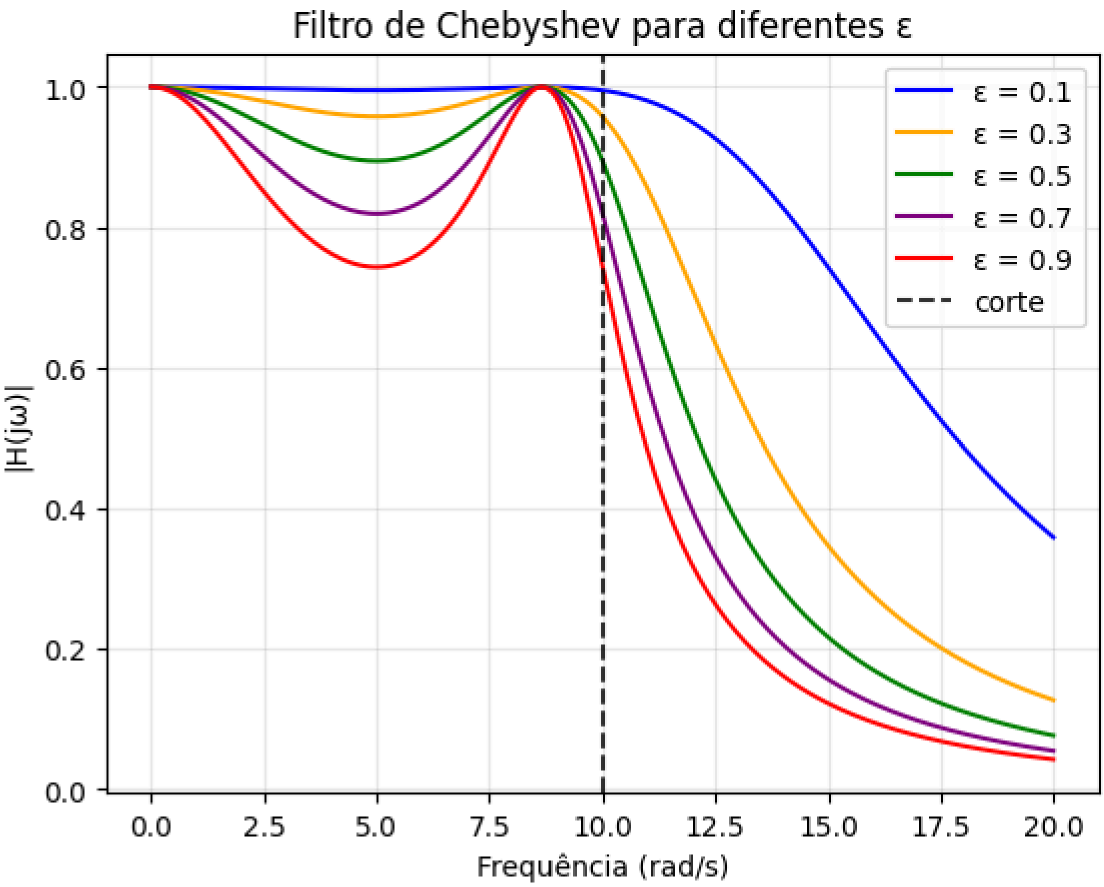
Fixando ω_c = 10 rad/s e n = 3, mas variando ε para 0.1, 0.3, 0.5, 0.7 e 0.9, obtemos as seguintes respostas em frequência

In [332…

```
w = np.linspace(0, 20, 200)

resposta1 = resposta_chebyshev(w, 10, 3, 0.1)
resposta2 = resposta_chebyshev(w, 10, 3, 0.3)
resposta3 = resposta_chebyshev(w, 10, 3, 0.5)
resposta4 = resposta_chebyshev(w, 10, 3, 0.7)
resposta5 = resposta_chebyshev(w, 10, 3, 0.9)

plt.plot(w, resposta1, label="ε = 0.1", color="blue")
plt.plot(w, resposta2, label="ε = 0.3", color="orange")
plt.plot(w, resposta3, label="ε = 0.5", color="green")
plt.plot(w, resposta4, label="ε = 0.7", color="purple")
plt.plot(w, resposta5, label="ε = 0.9", color="red")
plt.xlabel("Frequência (rad/s)")
plt.ylabel("|H(jω)|")
plt.title(f"Filtro de Chebyshev para diferentes ε")
plt.axvline(x=10, color="black", linestyle="--", alpha=0.8, label="corte"
plt.legend()
plt.grid(True, alpha=0.3)
plt.show()
```



Comentários:

Nota-se que, quanto maior o valor de ε, maior é a ondulação (ripple) na banda de passagem do filtro. Esse comportamento é característico dos filtros de Chebyshev, pois o parâmetro ε controla diretamente a amplitude das oscilações nessa região.

Além disso, observa-se que, à medida que ε aumenta, a transição entre a banda passante e a banda de rejeição torna-se mais abrupta, resultando em uma queda mais rápida após a frequência de corte. Esse efeito indica que valores maiores de ε proporcionam maior seletividade, permitindo uma atenuação mais eficiente das frequências acima da frequência de corte — ainda que à custa de maior distorção na banda passante.

2) Filtro de Butterworth

Implementando uma função para calcular o módulo da resposta em frequência para o filtro de Butterworth

In [333…

```
def resposta_butterworth(w, wc, n):
    resposta = 1 / np.sqrt(1 + (w/wc)**(2*n))

    return resposta
```

-> Item c)

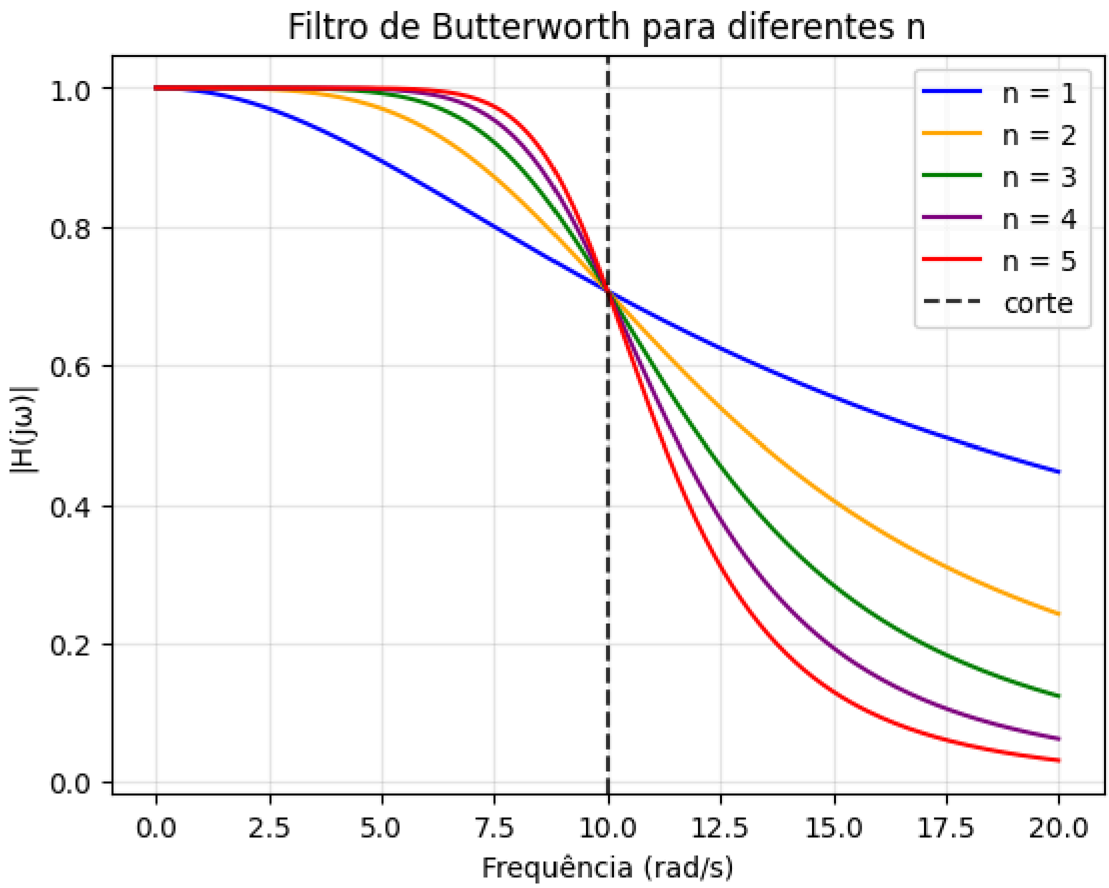
Fixando ω_c = 10 rad/s e variando n de 1 a 5, obtemos as seguintes respostas em frequência

In [334…

```
w = np.linspace(0, 20, 200)

resposta1 = resposta_butterworth(w, 10, 1)
resposta2 = resposta_butterworth(w, 10, 2)
resposta3 = resposta_butterworth(w, 10, 3)
resposta4 = resposta_butterworth(w, 10, 4)
resposta5 = resposta_butterworth(w, 10, 5)

plt.plot(w, resposta1, label="n = 1", color="blue")
plt.plot(w, resposta2, label="n = 2", color="orange")
plt.plot(w, resposta3, label="n = 3", color="green")
plt.plot(w, resposta4, label="n = 4", color="purple")
plt.plot(w, resposta5, label="n = 5", color="red")
plt.xlabel("Frequência (rad/s)")
plt.ylabel("|H(jω)|")
plt.title(f"Filtro de Butterworth para diferentes n")
plt.axvline(x=10, color="black", linestyle="--", alpha=0.8, label="corte"
plt.legend()
plt.grid(True, alpha=0.3)
plt.show()
```



Comentários:

Nota-se que quanto maior o valor de n, assim como no filtro de Chebyshev, as ondas caem mais rapidamente para zero após a frequência de corte, indicando uma maior seletividade do filtro.

Vale ressaltar que a suavidade da banda de passagem característica do filtro de Butterworth foi mantida para todos os valores de n.

3) Filtragem de um pulso retangular

Definindo uma função que calcula a Transformada de Fourier de um retângulo

In [335…

```
def fourier_retangulo(w, tau):
    transformada = np.sinc((tau*w) / (2*pi)) * tau

    return transformada
```

-> Item d)

Calculando X(jω) considerando $\tau = 2\pi/\omega_m$ e $\omega_m = 7.5$ rad/s (sabemos que a transformada de um retângulo é o sampling)

In [336…

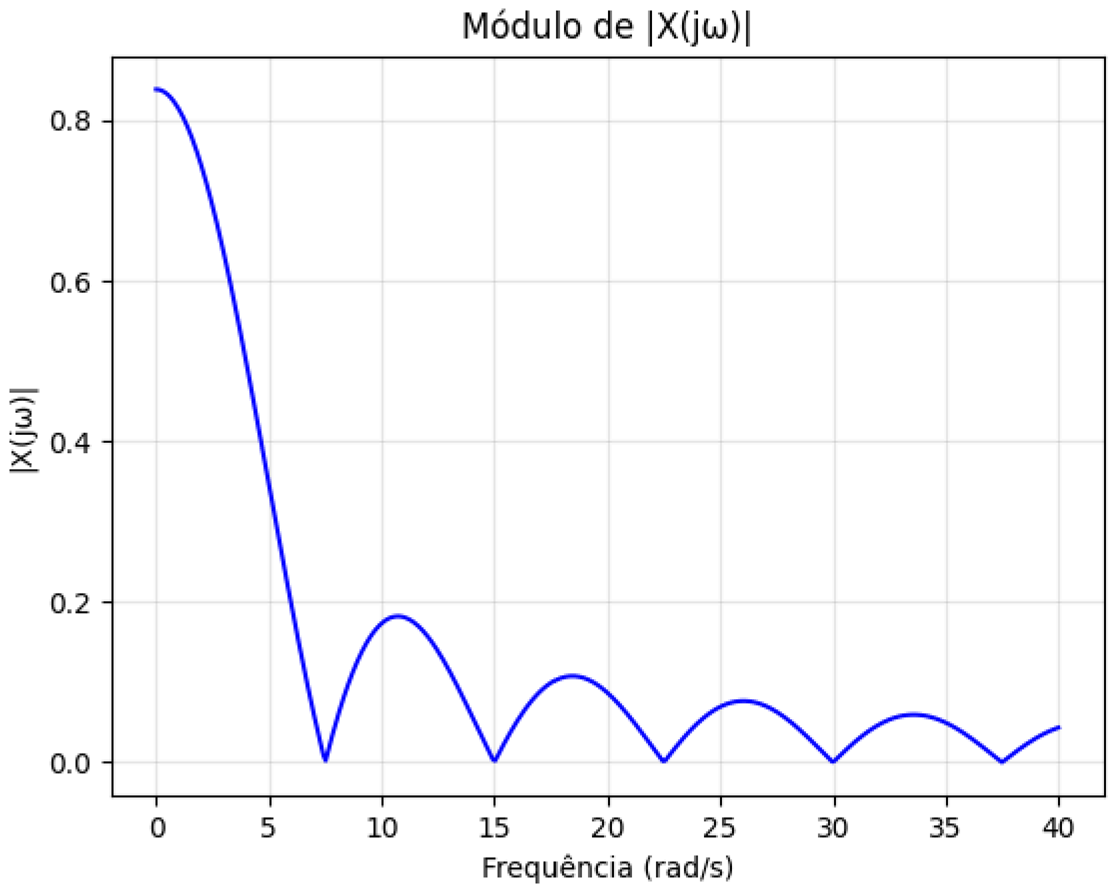
```
w = np.linspace(0, 40, 400)
wm = 7.5
```

```
tau = 2*pi/wm

transformada = fourier_retangulo(w, tau)
modulo_transformada = np.abs(transformada)
```

Plotando a transformada num gráfico

In [337…
```
plt.plot(w, modulo_transformada, label="", color="blue")
plt.xlabel("Frequência (rad/s)")
plt.ylabel("|X(jω)|")
plt.title(f"Módulo de |X(jω)|")
plt.grid(True, alpha=0.3)
plt.show()
```



Comentários:

Nota-se que o módulo apresenta valor zero quando há o fim de um período e o começo de outro. A amplitude não é uniforme e está cada vez diminuindo mais, uma vez que é amortecido (o que pode ser visível pelo fato da onda ser expressa pela função Sampling)

-> Item e)

Implementando uma função que aplica o FPB ideal

In [338…
```
def resposta_fpb_ideal(w, wc):
    resposta = np.copy(w)
```

```
    for i in range(0, len(w)):
        if np.abs(w[i]) <= wc:
            resposta[i] = 1
        else:
            resposta[i] = 0

    return resposta
```

Fixando ω_c = 10 rad/s e calculando $|H_{ideal}(j\omega)|$

In [339…

```
w = np.linspace(0, 20, 200)

h_ideal = resposta_fpb_ideal(w, 10)
```

Fixando ε = 0.6, n = 4 e ω_c = 10 rad/s e calculando $|H_C(j\omega)|$

In [340…

```
h_chebyshev = resposta_chebyshev(w, 10, 4, 0.6)
```

Fixando n = 2 e ω_c = 10 rad/s e calculando $|H_B(j\omega)|$

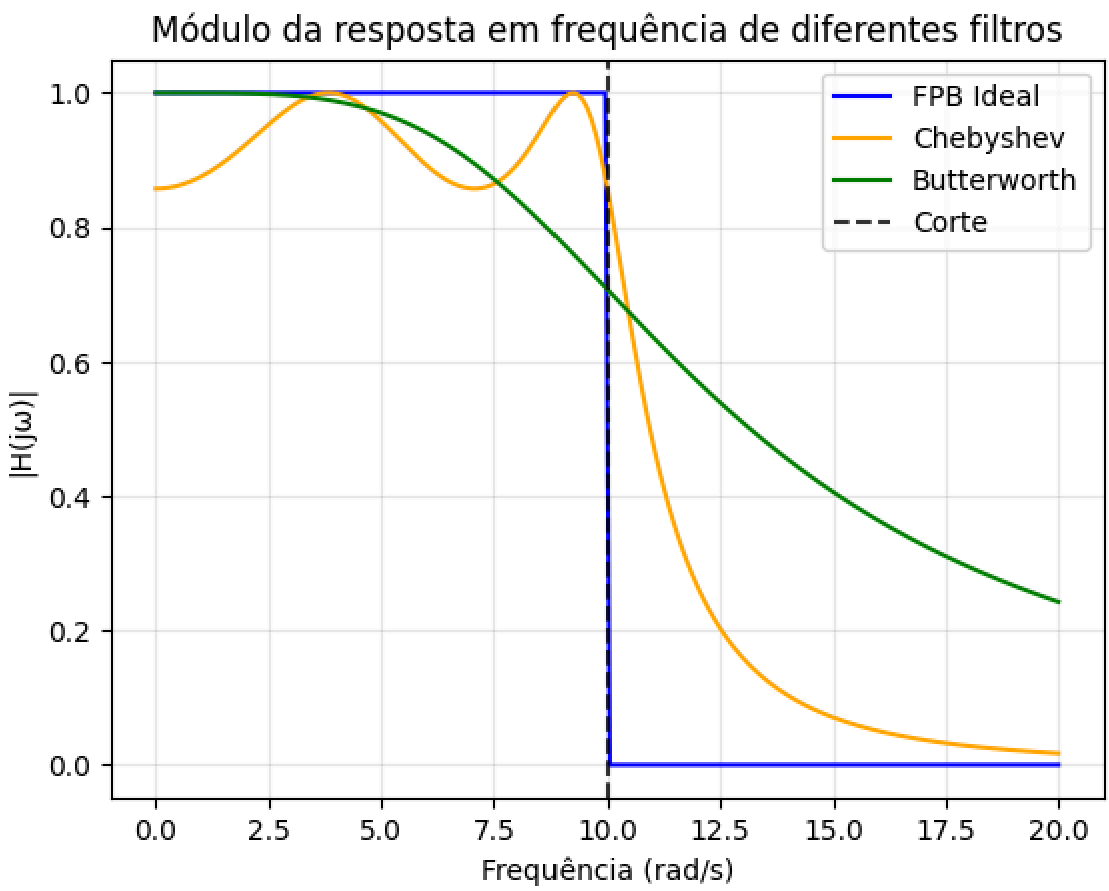
In [341…

```
h_butterworth = resposta_butterworth(w, 10, 2)
```

Plotando o módulo das respostas em frequência num gráfico

In [342…

```
plt.plot(w, h_ideal, label="FPB Ideal", color="blue")
plt.plot(w, h_chebyshev, label="Chebyshev", color="orange")
plt.plot(w, h_butterworth, label="Butterworth", color="green")
plt.xlabel("Frequência (rad/s)")
plt.ylabel("|H(jω)|")
plt.title(f"Módulo da resposta em frequência de diferentes filtros")
plt.axvline(x=10, color="black", linestyle="--", alpha=0.8, label="Corte"
plt.legend()
plt.grid(True, alpha=0.3)
plt.show()
```



Comentários sobre as respostas em frequência:

Nota-se que antes da frequência de corte, o filtro de Butterworth parece mais próximo do ideal, uma vez que apresenta menos ondulação, enquanto o de Chebyshev apresenta bastante.

Já após a frequência de corte, o filtro de Chebyshev passa a ser mais próximo do ideal, uma vez que desce para 0 mais rapidamente, enquanto o de Butterworth apresenta uma suavidade muito maior, demorando mais para chegar a 0.

Calculando X(jω) e as respostas novamente, agora num linspace de 0 a 40, para melhor vizualização

In [343…

```
w = np.linspace(0, 40, 400)

transformada = fourier_retangulo(w, 2*pi/7.5)

h_ideal = resposta_fpb_ideal(w, 10)
h_chebyshev = resposta_chebyshev(w, 10, 4, 0.6)
h_butterworth = resposta_butterworth(w, 10, 2)
```

Aplicando os filtros em X(jω) de modo a calcular o módulo das saídas $|Y_{ideal}(j\omega)|$, $|Y_C(j\omega)|$ e $|Y_B(j\omega)|$

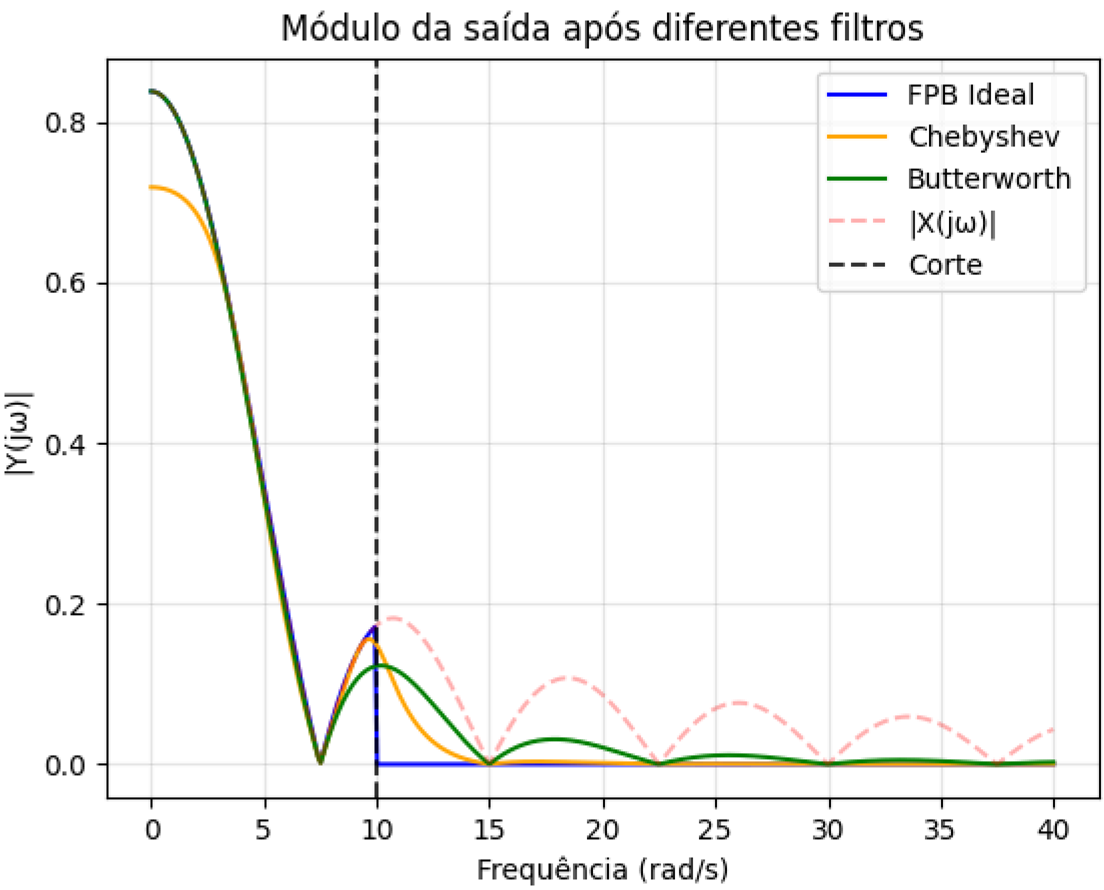
In [344…

```
saida_ideal = np.abs(h_ideal * transformada)
saida_chebyshev = np.abs(h_chebyshev * transformada)
saida_butterworth = np.abs(h_butterworth * transformada)
```

Plotando as saídas num gráfico

In [345…

```
plt.plot(w, saida_ideal, label="FPB Ideal", color="blue")
plt.plot(w, saida_chebyshev, label="Chebyshev", color="orange")
plt.plot(w, saida_butterworth, label="Butterworth", color="green")
plt.xlabel("Frequência (rad/s)")
plt.ylabel("|Y(jω)|")
plt.title(f"Módulo da saída após diferentes filtros")
plt.plot(w, np.abs(transformada), label="|X(jω)|", linestyle="--", alpha=
plt.axvline(x=10, color="black", linestyle="--", alpha=0.8, label="Corte"
plt.legend()
plt.grid(True, alpha=0.3)
plt.show()
```



Comentários sobre as saídas e conclusão:

Nota-se que, antes da frequência de corte, a saída referente ao filtro de Chebyshev apresenta uma leve variação na amplitude (ripple) devido às ondulações características na banda passante. No entanto, próximo da frequência de corte, sua resposta em magnitude se aproxima mais da do filtro ideal, já que o Chebyshev possui uma transição mais abrupta entre a banda passante e a banda de rejeição. Por outro lado, a saída referente ao filtro de Butterworth mantém a amplitude praticamente constante antes da frequência de corte — característica de sua resposta maximamente plana —, mas apresenta uma transição mais gradual, afastando-se mais do comportamento ideal nessa região.

Após a frequência de corte, observa-se que o filtro de Chebyshev tende a zero mais rapidamente que o de Butterworth, mostrando uma maior seletividade e melhor rejeição das altas frequências logo após o corte.

Dessa forma, conclui-se que o filtro de Chebyshev oferece maior seletividade após a frequência de corte, mas perde um pouco em constância na banda passante devido ao ripple. Já o filtro de Butterworth mantém maior uniformidade antes da frequência de corte, porém atenua mais lentamente as componentes acima dela.