



**INSTITUTO
FEDERAL**

Santa Catarina

Câmpus
São José

Calculo de sobreposição de espectro

Redes de Transmissão

Arthur Cadore Matuella Barcella

25 de Agosto de 2024

Engenharia de Telecomunicações - IFSC-SJ

Sumário

1. Introdução:	3
2. Frequências utilizadas:	3
2.1. Comprimentos de entrada:	3
2.2. Conversão para frequência:	3
3. Alterando o script python:	3
3.1. Script original:	3
3.2. Adição do calculo de sobreposição:	4
3.3. Código final:	4

1. Introdução:

2. Frequências utilizadas:

2.1. Comprimentos de entrada:

$$\lambda_1 - 1528,77 \text{ nm} \quad (1)$$

$$\lambda_2 - 1529,55 \text{ nm} \quad (2)$$

$$\lambda_3 - 1530,33 \text{ nm} \quad (3)$$

$$\lambda_4 - 1531,12 \text{ nm} \quad (4)$$

2.2. Conversão para frequência:

Considerando a velocidade da luz como 299.792.458 m/s, temos como que a frequência de cada uma das ondas é dada por:

$$F_1 = \frac{299792458}{1528.77 * 10^{-9}} = 196100432373738,3 \text{ ou } 196,1004323737383 \text{ THz} \quad (5)$$

$$F_2 = \frac{299792458}{1529.55 * 10^{-9}} = 196000430191886,5 \text{ ou } 196,0004301918865 \text{ THz} \quad (6)$$

$$F_3 = \frac{299792458}{1530.33 * 10^{-9}} = 195900529951056,3 \text{ ou } 195,9005299510563 \text{ THz} \quad (7)$$

$$F_4 = \frac{299792458}{1531.12 * 10^{-9}} = 195799452688228,2 \text{ ou } 195,7994526882282 \text{ THz} \quad (8)$$

3. Alterando o script python:

3.1. Script original:

```
1 # IFSC Câmpus São José
2 # Engenharia de Telecomunicações
3 # RTR029007 Redes de Transmissão
4 # Professor: Fábio Alexandre de Souza
5 # Four Wave Mixing
6
7 import os
8 import itertools
9
10 print('=====Inicio=====')
11 print('Ff = fi + fj - fk, onde i dif k e j dif k')
12
13 n = int(input('Digite n: '))
14
15 canais = list()
```

```

16
17 for cont in range(0,n):
18     canais.append(float(input('Digite f em THz: ')))
19 c = 0;
20
21 print('=====Arranjos=====')
22
23 for f in itertools.product(range(0, n), repeat=3):
24     if f[0] != f[2] and f[1] != f[2]:
25         c = c+1
26         print('\nArranjo ',c, '=', '['+canais[f[0]], canais[f[1]],
27             canais[f[2]],']')
28         Ff = canais[f[0]] + canais[f[1]] - canais[f[2]]
29         print('Ff = ', round(Ff,2))
30         if Ff in canais:
31             print("Sobreposição!")
32
33 numar = c
34 print('número de conjuntos = ',numar)
35
36 print('=====')
37 print('=====FIM=====')

```

3.2. Adição do calculo de sobreposição:

```

1 # Função para verificar sobreposição com tolerância
2 def verificar_sobreposicao(canais, resultados, tolerancia=0.000435):
3     sobreposicoes = []
4     canais_set = set(canais)
5     for f, arranjo in resultados:
6         for canal in canais:
7             if abs(f - canal) <= tolerancia:
8                 sobreposicoes.append((arranjo, f, canal))
9                 break
10    return sobreposicoes

```

3.3. Código final:

```

1 # IFSC Câmpus São José
2 # Engenharia de Telecomunicações
3 # RTR029007 Redes de Transmissão
4 # Professor: Fábio Alexandre de Souza
5 # Four Wave Mixing
6
7 import os
8 import itertools
9
10 # Entradas
11 n = 4
12
13 canais = [
14     196.1004323737383,
15     196.0004301918865,
16     195.9005299510563,
17     195.7994526882282

```

```

18 ]
19
20 # Autor: Arthur Cadore M. Barcella
21 # Função para verificar sobreposição com tolerância
22 def verificar_sobreposicao(canais, resultados, tolerancia=0.000435):
23     sobreposicoes = []
24     canais_set = set(canais)
25     for f, arranjo in resultados:
26         for canal in canais:
27             if abs(f - canal) <= tolerancia:
28                 sobreposicoes.append((arranjo, f, canal))
29                 break
30     return sobreposicoes
31
32 # Arranjos
33 resultados = []
34 c = 0
35 print('=====Arranjos=====')
36
37 for f in itertools.product(range(0, n), repeat=3):
38     if f[0] != f[2] and f[1] != f[2]:
39         c += 1
40         arranjo = [canais[f[0]], canais[f[1]], canais[f[2]]]
41         Ff = canais[f[0]] + canais[f[1]] - canais[f[2]]
42         resultados.append((round(Ff, 2), arranjo))
43         print('\nArranjo ', c, '=', '[' + arranjo[0], arranjo[1], arranjo[2],
44             ']')
45         print('Ff = ', round(Ff, 2))
46
47 # Verificar sobreposição
48 sobreposicoes = verificar_sobreposicao(canais, resultados)
49 num_sobreposicoes = len(sobreposicoes)
50 print('=====Sobreposições=====')
51
52 if sobreposicoes:
53     print('\nCanais com Sobreposição:')
54     for arranjo, freq_calculada, freq_original in sobreposicoes:
55         print(f'Arranjo {arranjo} tem sobreposição com Ff = {freq_calculada}
56 THz (próximo de {freq_original} THz)')
57
58 print('=====Resultados=====')
59
60 print('Número de conjuntos = ', c)
61 print('Número de sobreposições = ', num_sobreposicoes)

```