

	Universidade de São Paulo Escola de Engenharia de São Carlos Laboratório de Análise Computacional em Sistemas Elétricos de Potência	LACOSEP
---	---	----------------

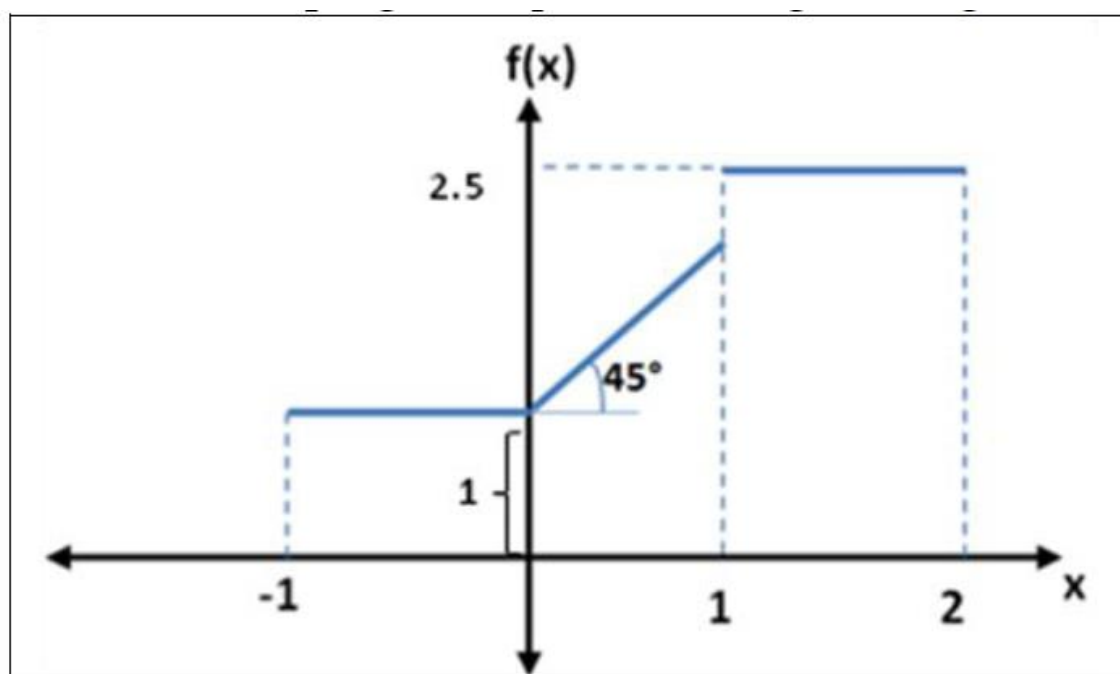
Exercícios Propostos Python

Professor Elmer Pablo Tito Cari

Vitor Augusto Corrêa

NUSP: 8957217

Procedimento geral



Será necessário a criação 2 vetores. Um para o eixo x e outro para o eixo y.

O vetor do eixo x recebe valores de -1 a 2, com passo de 0.001, resultando em 3000 elementos.

O vetor y, inicialmente, terá o mesmo tamanho do vetor x, mas com seus elementos possuindo valores 0 e que serão adequados até o fim do algoritmo.

Utiliza-se uma variável de controle da função de laço (while, no caso), que será responsável por percorrer o número de elementos existentes entre 2 e -

1, com passo de 0.001 (i.e.: $2 - (-1)/0.001 = 3000$). Ou seja, ela percorrerá o vetor x .

E outra variável de controle, para indexar os elementos do vetor y .

Dentro do laço, utiliza-se o IF, considerando a posição atual no eixo x , para poder inserir o valor adequado no vetor y , para dada posição do vetor x , são 3 casos:

1. y recebe 1, para x maior que -1 e menor do que 0
2. y recebe o mesmo valor atual de x , para x maior que 0 e menor que 1
3. y recebe 2.5, para x entre 1 e 2

Para os itens 3 a 5, o que ocorre é uma generalização do 1 e 2, onde:

1. -1 passa a ser a
2. 2 passa a ser c
3. 1 passa a ser b
4. 45 graus passam a ser teta

Essa é a idéia geral. Mais detalhes e o que cada vetor, elemento, comando etc .Faz, serão descritos no próprio código.

Problema 1:

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

#Variaveis
x = np.arange(-1.0,2.0, 0.001)
y = 0*x

cont = -1.0 #controle comparativo
posy = 0 #indexador do vetor y

espacos = len(x) #numero de elementos para controle do while
```

```
#Funcoes
```

```
while cont<(2-0.001): #para dar o numero de elementos corretos
```

```
    if cont <= 0:
```

```
        y[posy] = 1
```

```
    elif 0 < cont < 1:
```

```
        y[posy] = cont + 1 #reta
```

```
    else:
```

```
        y[posy] = 2.5
```

```
    posy = posy + 1 #incremento no indice
```

```
    cont = cont + 0.001 #incremento no controle do if
```

```
#Plotagem
```

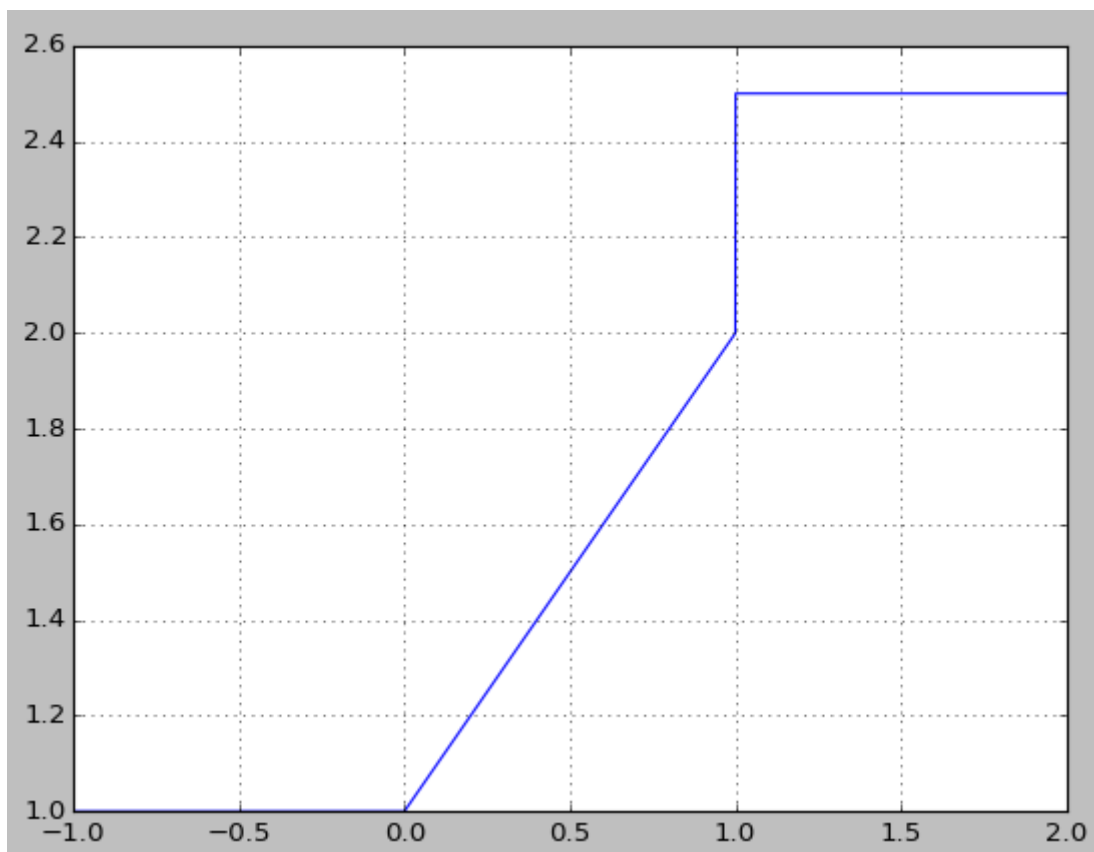
```
tabela = np.array([x,y])
```

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

```
# plt.plot(tabela[:,0], tabela[:,1])
```

```
plt.grid(True)
```

```
plt.show()
```



Note: y começa em 1(apesar de não estar mostrando o zero !)

Problema 2:

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

```
#Variaveis
```

```
x = np.arange(-1.0,2.0, 0.001)
```

```
y = 0*x
```

```
cont = 0
```

```
#Função
```

```
#Assumir que o -1 comeca no zero. O proprio 0 em 1000, 1 em 2000 e 2 em 3000
```

```
#existe 3000 espacos entre 2 e -1, com passo de 0.001 (fazer:  $(2 - (-1))/0.001 = 3000$  )
```

```
for ctrl in range(0,3000,1):
```

```
    if cont <= 1000:
```

```
        y[cont] = 1 # primeira reta
```

```
    else:
```

```
        if cont <= 2000:
```

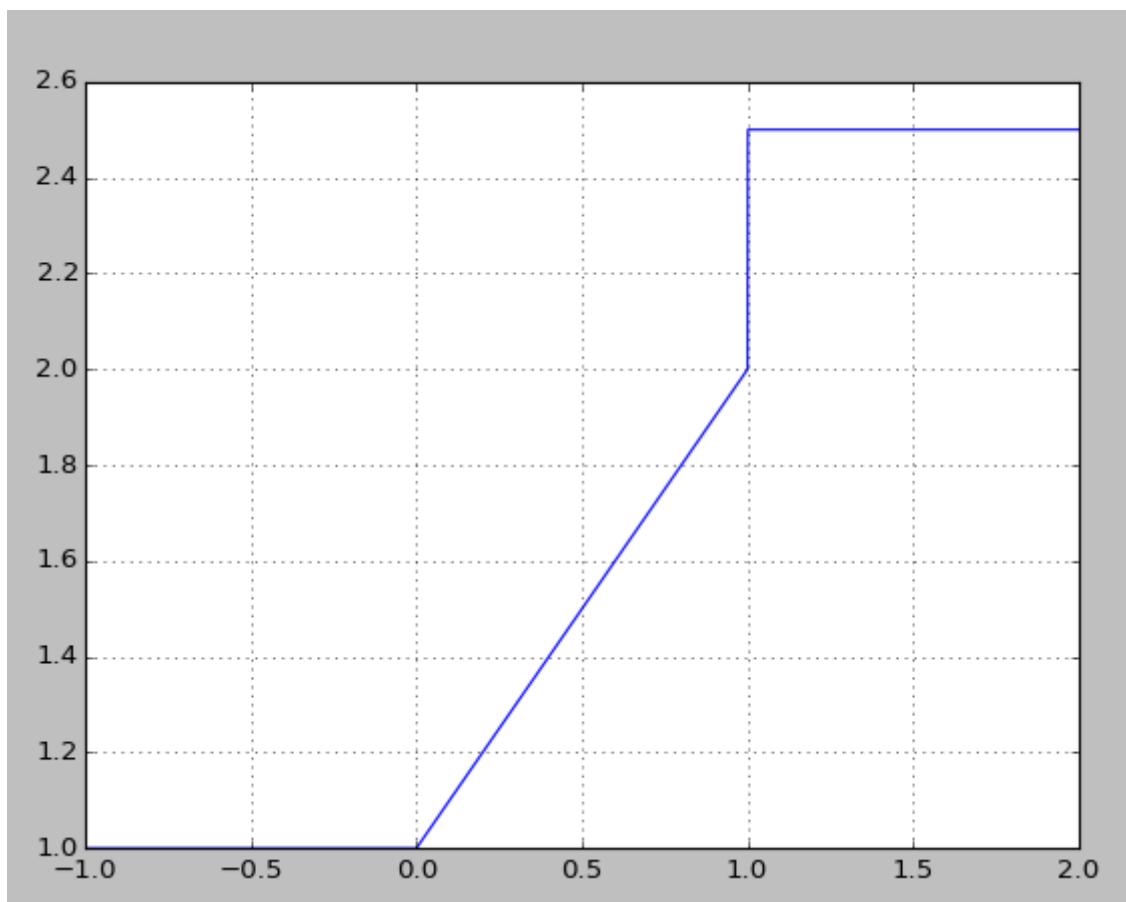
```
            y[cont] = cont*0.001 #reta inclinada
```

```
        else:
```

```
            y[cont] = 2.5 #segnda reta
```

```
    cont = cont + 1
```

```
#Plotagem  
tabela = np.array([x,y])  
plt.plot(x,y)  
# plt.plot(tabela[:,0], tabela[:,1])  
plt.grid(True)  
plt.show()
```



Note: y começa em 1(apesar de não estar mostrando o zero !) IDEM!

Problema 3:

```
# -*- coding: utf-8 -*-
```

```
''''''
```

Created on Mon Apr 03 20:05:00 2017

```
@author: User
```

```
''''''
```

```
def funcao_Vitor(a,b,c,teta):
```

```
    import matplotlib.pyplot as plt
```

```
    import numpy as np
```

```
    import math
```

```
    #Variaveis
```

```
    x = np.arange(a,c,1.0)
```

```
    y = 0*x
```

```
    cont = a;  #controle do eixo x do grafico
```

```
    posy = 0; #indexador do vetor y
```

```
    angg = (0.01745)*teta #conversor de grau para rad
```

```
    #Funcao
```

```
    while cont< c:
```

```
    if cont < 0:
        y[posy] = 1
    elif 0 <= cont < b:
        y[posy] = math.tan(angg)*cont + 1 #reta inclinada
    elif b <= cont < c:
        y[posy] = 2.5
    cont = cont + 1
    posy = posy + 1
return x, y
```

#Programa

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

```
x, y = funcao_Vitor(-5,5,12,45)
```

```
tabela = np.array([x,y])
```

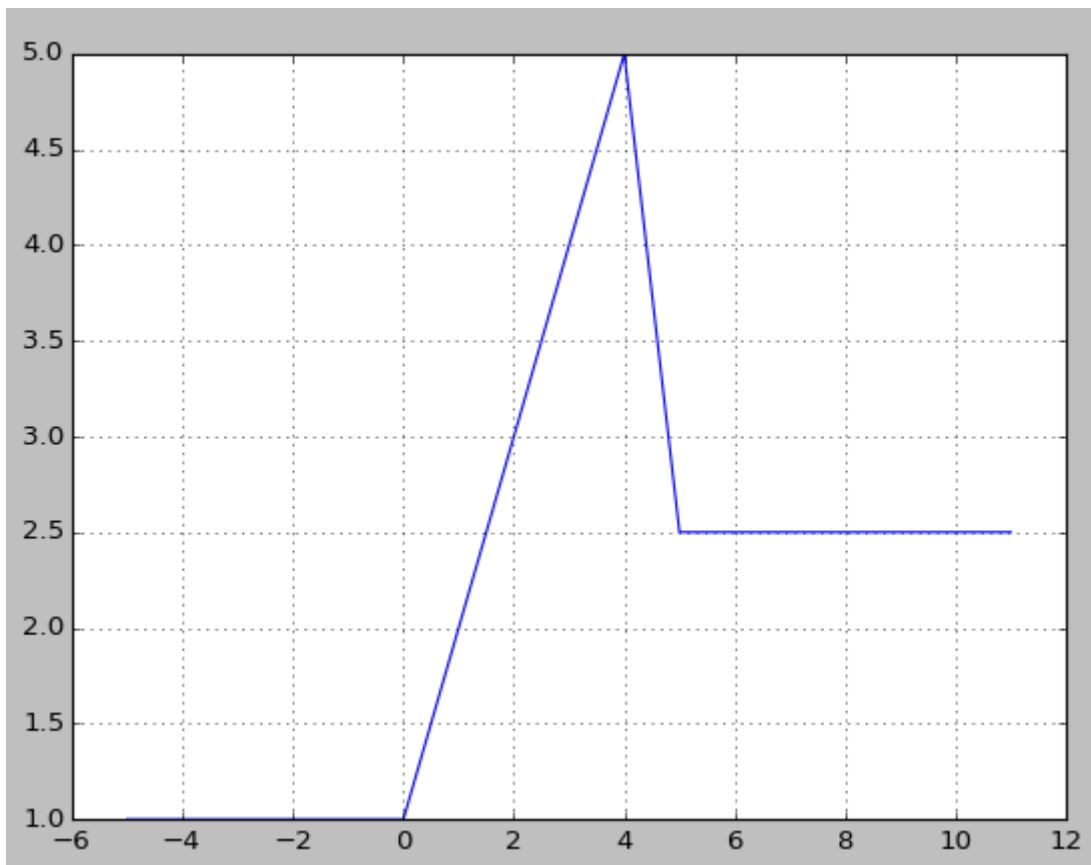
#Plotagem

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

```
#plt.plot(tabela[:,0], tabela[:,1])
```

```
plt.grid(True)
```

```
plt.show()
```



Para: $a = -5.0$; $b = 5.0$; $c = 12.0$; $teta = 45$ graus

Problema 4:

```
def funcao_Vitor(a,b,c,teta):  
    import matplotlib.pyplot as plt  
    import numpy as np  
    import math  
    #Variaveis  
    x = np.arange(a,c,1.0)  
    y = 0*x  
    cont = a; #controle do eixo x do grafico  
    posy = 0; #indexador do vetor y
```

```
    angg = (0.01745)*teta #conversor de grau para rad

#Funcao
while cont< c:

    if cont < 0:

        y[posy] = 1

    elif 0 <= cont < b:

        y[posy] = math.tan(angg)*cont + 1 #reta inclinada

    elif b <= cont < c:

        y[posy] = 2.5

    cont = cont + 1

    posy = posy + 1

return x, y
```

```
#Programa
```

```
import matplotlib.pyplot as plt
```

```
import numpy as np
```

```
x, y = funcao_Vitor(-5,5,12,45)
```

```
tabela = np.array([x,y])
```

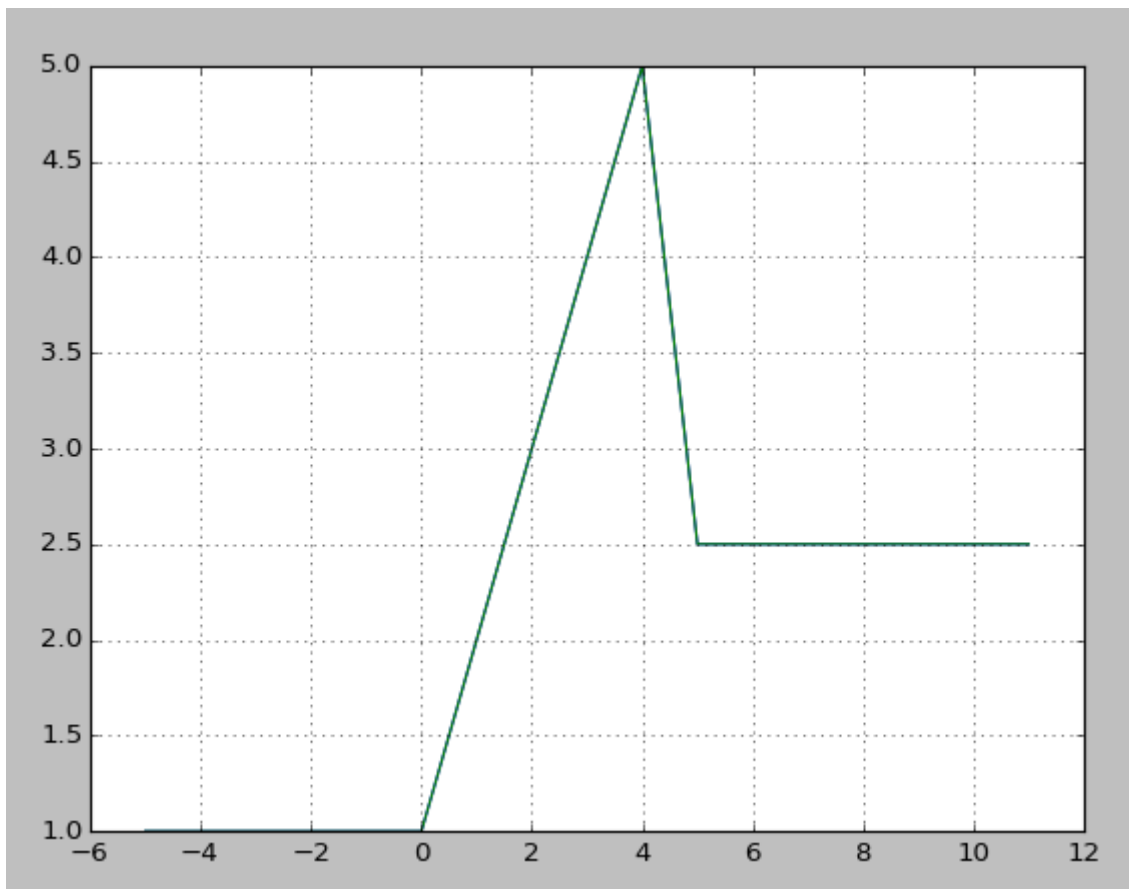
```
#Plotagem
```

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

```
#plt.plot(tabela[:,0], tabela[:,1])
```

```
plt.grid(True)
```

`plt.show()`



Para: $a = -5.0$; $b = 5.0$; $c = 12.0$; $teta = 45$ graus

Problema 5:

Created on Mon Apr 03 18:37:25 2017

@author: User

```
def funcao_Vitor(a,b,c,teta,passo):
```

```
    import matplotlib.pyplot as plt
```

```

import numpy as np

import math

#Variaveis

x = np.arange(a,c,passo)

y = 0*x

cont = a;#começo

posy = 0;#controle do indice do vetor y

angg = (57.3)*teta #conversao de graus para radianos, para aplciar na
funcao tangente da eq da reta

espacamento_total = len(x) #porque a funcao range so aceita inteiros
como parametros

                                #e o numero de elementos de x compreende a
quantidade de valores (c-a)/passo


#Funcao

for ctrl in range(0,espacamento_total):

    if cont <= 0:

        y[posy] = 1

    elif 1 < cont < b:

        y[posy] = math.tan(angg)*cont

    elif b <= cont < c:

        y[posy] = 2.5

    cont = cont + passo

    posy = posy + 1

return x, y

```

```
import matplotlib.pyplot as plt

import numpy as np

#Programa
x, y = funcao_Vitor(-2,2,5,75,0.01)

tabela = np.array([x,y])

#Imprimir
plt.plot(x,y)
#plt.plot(tabela[:,0], tabela[:,1])
plt.grid(True)
plt.show()
```

