In [2]:

```
import numpy as np
import scipy
import pandas as pd
```

1 Problema de Fluxo de Carga

1.1 Conseguir matriz admitância

In [8]:

```
def get_admitance(R,X):
    i =1j
    return R/(R*R+X*X)-X*i/(R*R+X*X)
```

In [21]:

```
lines = pd.DataFrame([[0.0050, 0.200],
                      [0.0090, 0.100],
                      [0.0030, 0.060],
                      [0.00225, 0.100],
                      [0.0010, 0.05]],
                    columns=['r', 'x'],
                    index=['y12','y13','y13*','y23','y34'])
```

In [24]:

```
y = get_admitance(lines.r,lines.x)
y
```

Out[24]:

```
y12      0.124922-4.996877j
y13      0.892769-9.919651j
y13*    0.831255-16.625104j
y23      0.224886-9.994940j
y34     0.399840-19.992003j
dtype: complex128
```

In [26]:

```
y['y12']+y['y13']+y['y13*']+0.54j/2+0.82j/2+0.22j/2
```

Out[26]:

```
(1.8489456936887732-30.751631687095323j)
```

In [27]:

```
-(y['y13']+y['y13*'])
```

Out[27]:

```
(-1.7240237698911467+26.54475473519026j)
```

In [28]:

```
y['y12']+y['y23']+0.54j/2+0.88j/2
```

Out[28]:

```
(0.34980807518348733-14.281817013498875j)
```

In [29]:

```
y['y13']+y['y13*']+y['y34']+y['y23']+0.82j/2+0.22j/2+0.88j/2+0.44/2
```

Out[29]:

```
(2.5687499852514177-55.57169799550459j)
```

In [30]:

```
y['y34']+0.44j/2
```

Out[30]:

```
(0.39984006397441013-19.772003198720512j)
```

1.2 Calcular Potências

In [170…

```
def get_real_power(Vi,Vm,Gim,Bim,TETAim):
    Gim = np.array(Gim)
    Bim = np.array(Bim)
    row = Vm*(Gim*np.cos(TETAim)+Bim*np.sin(TETAim))

    return Vi#*sum(row)
```

In [171…

```
def get_complex_power(Vi,Vm,Gim,TETAim,Bim):
    row = Vm*(Gim*np.sin(TETAim)-Bim*np.cos(TETAim))
    return Vi*sum(row)
```

In [172…

```
V = [1, 1, 1.02, 1]

G = [[1.84,-0.125,-0.724,0],
     [0.125,0.35,-0.225,0],
     [-1.724,0.225,2.57,-0.4],
     [0,0,-0.4,0.4]]

B = [[-30.8,5,26.6,0],
     [5, -14.3, 10, 0],
     [26.6, 10, -55.6,20],
     [0, 0, 20, 19.8]]

TETAim=[0,0,0,0]
```

In [173…

```
Gim = [0,-3.73,3.73,0]
Gim *np.cos(TETAim)
```

Out[173…

```
array([ 0.  , -3.73,  3.73,  0.  ])
```

In [174…

```
get_real_power(1.03,[1,1,1.03,1],[0,-3.73,3.73,0],
              [0,49.7,-49.7,0],0)
```

Out[174…

```
1.03
```

In [109…

```
1.03*(-3.72+1.03*3.72)
```

Out[109…

```
0.11494800000000015
```

In [112…

```
1.03*3.72
```

Out[112…

```
3.8316000000000003
```

In [ ]: