Métodos de Gauss-Seidl e Relaxação em Python

```
import numpy as np
# ------------------------ calculo da matriz inversa ---------------
def cami(x):     # usando np: np.linalg.inv(x)
 ta=len(x[0])
 y=np.identity(ta,float)
 i=0
 while(i<ta):
  j=0
  while(j<ta):
   if (j!=i):
    pivot=-x[j][i]/x[i][i]
    k=0
    while k<ta:
     x[j][k]=x[j][k]+x[i][k]*pivot
     y[j][k]=y[j][k]+y[i][k]*pivot
     k=k+1
   j=j+1
  alvo=x[i][i]
  k=0
  while k<ta:
   y[i][k]=y[i][k]/alvo
   x[i][k]=x[i][k]/alvo
   k=k+1
  i=i+1
 return y;

# ------------------- multiplicacao matricial ---------------------
def mm(a,b):    # definicao usando np: np.dot(a,b)
  ta=len(a)
  tb=len(a[0])
  tr=len(b[0])
  r=np.zeros((ta,tr))
  for i in range(ta):
    for j in range(tr):
      s=0
      for k in range(tb):
        s=s+a[i][k]*b[k][j]
      r[i][j]=s
  return r

#---------------- Método de Gauss Seidl (não testa convergência)---
def gaus_seid(x,b):
  ta=len(x[0])
  b=b.reshape((ta,1))
  D=np.zeros((ta,ta))
  S=np.zeros((ta,ta))
  I=np.zeros((ta,ta))
  F=np.zeros((ta,ta))
  G=np.zeros((ta,ta))
  for i in range(ta):
    for j in range(ta):
      if i<j:
        S[i][j]=x[i][j]
      elif i==j:
        I[i][j]=x[i][j]
      elif i>j:
        D[i][j]=x[i][j]
  F=D+I
  F=np.linalg.inv(F)   # OU cami(F)
  F=np.dot(F,b)  #   OU mm(F,b)
  F=np.around(F,decimals=6)   # se o exercicio pedir
```



```
  G=-(D+I)
  G=np.linalg.inv(G) #  OU cami(G)
  G=np.dot(G,S)  #    OU mm(G,S)
  G=np.around(G,decimals=6)    # se o exercicio pedir
  XN=np.zeros((ta,1),float)
  XV=np.ones((ta,1),float)
  while (0.00002<max(abs(XN-XV))):   #atualizar o erro...
    print('----------------------erro: ',max(abs(XN-XV)))
    XV=np.around(XN,decimals=6)
    XN=np.around((mm(G,XV))+F,decimals=6)
    print(XN)

a=np.array([[10.0,2,1],[1,5,1],[2,3,10]])
b=np.array([[7.0],[-8],[6]])
a=np.array([[42,9,21,4,3.0],[11,71,16,22,17],[19,24,89,10,31],
            [20,35,23,91,5],[12,30,25,14,82]])
b=np.array([[655.0],[1284],[1275],[1812],[1929]])

gaus_seid(a,b)
#----------------- relaxacao -----------------
import numpy as np
def relax(a,b):
  xv=np.zeros(len(a),float)
  quem = 1.0
  while 1==1:
    re=np.zeros(len(a),float)
    for j in range(len(a)):
      for k in range(len(a)):
        re[j]=re[j]+a[j][k]*xv[k]
      re[j]=re[j]-b[j]
    re=list(re)
    quem=max(np.abs(re))
    if quem<0.00001:    # alterar criterio de parada
      return
    for k in range(len(re)):
      if quem==abs(re[k]):
        ind=k
    novo=0
    for k in range(len(a)):
      if (k!=ind):
        novo=novo-a[ind][k]*xv[k]
    novo=novo+b[ind]
    xv[ind]=novo/a[ind][ind]
    print('novo xv=',xv)

a=np.array([[10.0,2,1],[1,-15,1],[2,3,10]])
b=np.array([7.0,32,6])
a=np.array([[42,9,21,4,3.0],[11,71,16,22,17],[19,24,89,10,31],
            [20,35,23,91,5],[12,30,25,14,82]])
b=np.array([655.0,1284,1275,1812,1929])
relax(a,b)
```