Prova 2 Estatística

Gabaritos

In [25]:
```
import numpy as np
def q1(a,b):
    print(a,b)
    print('certo=',np.linalg.solve(a,b))
    m=list(abs(a[:,0]))
    i=m.index(max(m))
    guarda=a[i].copy()
    a[i]=a[0].copy()
    a[0]=guarda.copy()
    guarda=b[i]
    b[i]=b[0]
    b[0]=guarda
    print('matriz apos pivoteamento col 1 \n',a,b)
    pivot=a[1,0]/a[0,0]
    for j in range(3):
        a[1,j]=a[1,j]-(pivot*a[0,j])
    b[1]=b[1]-pivot*b[0]
    pivot=a[2,0]/a[0,0]
    for j in range(3):
        a[2,j]=a[2,j]-(pivot*a[0,j])
    b[2]=b[2]-pivot*b[0]
    print('matriz após zerar col 1 \n',a,b)
    if abs(a[1,1])<abs(a[2,1]):
        guarda=a[1].copy()
        a[1]=a[2].copy()
        a[2]=guarda.copy()
        guarda=b[1]
        b[1]=b[2]
        b[2]=guarda
    print('matriz após pivoteamento col 2 \n',a,b)
    pivot=a[2,1]/a[1,1]
    for j in range(3):
        a[2,j]=a[2,j]-(pivot*a[1,j])
    b[2]=b[2]-pivot*b[1]
    print('matriz após zerar col 2 \n',a,b)
    z=b[2]/a[2,2]
    y=(b[1]-a[1,2]*z)/a[1,1]
    x=(b[0]-(a[0,1]*y+a[0,2]*z))/a[0,0]
    print('resposta \n',np.around(a,4),np.around(b,4))
    print('x = ',round(x,4),round(y,4),round(z,4))
a=np.array([[9.3,6,2],[-5.2,5.9,4.2],[8.1,-8.4,-4]])
b=[135.7,-6.1,17.8]
q1(a,b)
```

```
[[ 9.3  6.   2. ]
 [-5.2  5.9  4.2]
 [ 8.1 -8.4 -4. ]] [135.7, -6.1, 17.8]
certo= [9.90171596 6.8956066  1.12020097]
matriz apos pivoteamento col 1
 [[ 9.3  6.   2. ]
 [-5.2  5.9  4.2]
 [ 8.1 -8.4 -4. ]] [135.7, -6.1, 17.8]
matriz após zerar col 1
 [[  9.3          6.           2.        ]
 [  0.           9.25483871   5.31827957]
 [  0.         -13.62580645  -5.74193548]] [135.7, 69.77526881720429, -100.3903225
8064513]
matriz após pivoteamento col 2
 [[  9.3          6.           2.        ]
 [  0.         -13.62580645  -5.74193548]
 [  0.           9.25483871   5.31827957]] [135.7, -100.39032258064513, 69.7752688
1720429]
matriz após zerar col 2
 [[  9.3          6.           2.        ]
 [  0.         -13.62580645  -5.74193548]
 [  0.           0.           1.41827652]] [135.7, -100.39032258064513, 1.58875473
4848493]
resposta
 [[  9.3      6.       2.    ]
 [  0.     -13.6258  -5.7419]
 [  0.       0.       1.4183]] [ 135.7    -100.3903    1.5888]
x =  9.9017 6.8956 1.1202
```

In [26]:

```
import numpy as np
def q2(aa,bb,x0,epsila):
    print(aa,bb)
    print('certo=',np.linalg.solve(aa,bb))
    jj=[0,0,0]
    for i in range(3):
        m=list(abs(aa[i,:]))
        jj[i]=m.index(max(m))
    a=np.array([aa[jj.index(0),:],aa[jj.index(1),:],aa[jj.index(2),:]])
    b=[bb[jj.index(0)],bb[jj.index(1)],bb[jj.index(2)]]
    print('depois=')
    print(a,b)
    for i in range(3):
        diva=a[i,i]
        for j in range(3):
            a[i,j]=a[i,j]/diva
        b[i]=b[i]/diva
    x=[0]*3
    erro=[0]*3
    print('resposta')
    for k in range(1,6):
        for i in range(3):
            soma=b[i]
            for j in range(3):
                if i!=j:
                    soma=soma-a[i,j]*x0[j]
            x[i]=soma
            erro[i]=abs(x0[i]-x[i])
        print(k,round(x[0],4),round(x[1],4),round(x[2],4),'        ',end='')
        print('erro=',max(erro))
        if max(erro)<epsila:
            return
        x0=x.copy()
a=np.array([[-4.6,-12.4,6],[-9.6,0.4,-7.2],[2.5,-7.4,-11.2]])
```

```
b=[-23.2,-100.4,-20.7]
q2(a,b,[8.0,0.5,3.3],0.001)
```

```
[[ -4.6 -12.4   6. ]
 [ -9.6   0.4  -7.2]
 [  2.5  -7.4 -11.2]] [-23.2, -100.4, -20.7]
certo= [8.0017196  0.50096683 3.30331647]
depois=
[[ -9.6   0.4  -7.2]
 [ -4.6 -12.4   6. ]
 [  2.5  -7.4 -11.2]] [-100.4, -23.2, -20.7]
resposta
1 8.0042 0.5 3.3036         erro= 0.004166666666668206
2 8.0015 0.5002 3.3045         erro= 0.00267857142857153
3 8.0008 0.5016 3.3038         erro= 0.0014436923963132564
4 8.0014 0.5015 3.3027         erro= 0.0011078735824812114
5 8.0022 0.5008 3.3029         erro= 0.0008270854243477288
```

In [27]:

```
import numpy as np
def q3(aa,bb,x0,epsila):
    print(aa,bb)
    print('certo=',np.linalg.solve(aa,bb))
    jj=[0,0,0,0]
    for i in range(4):
        m=list(abs(aa[i,:]))
        jj[i]=m.index(max(m))
    a=np.array([aa[jj.index(0),:],aa[jj.index(1),:],aa[jj.index(2),:],aa[jj.index(
    b=[bb[jj.index(0)],bb[jj.index(1)],bb[jj.index(2)],bb[jj.index(3)]]
    print('depois=')
    print(a,b)
    for i in range(4):
        diva=a[i,i]
        for j in range(4):
            a[i,j]=a[i,j]/diva
        b[i]=b[i]/diva
    erro=[0.0]*4
    print('resposta')
    for k in range(5):
        x=x0.copy()
        for i in range(4):
            soma=b[i]
            for j in range(4):
                if i!=j:
                    soma=soma-a[i,j]*x0[j]
            x0[i]=soma
            erro[i]=abs(x0[i]-x[i])
        print(k,round(x[0],4),round(x[1],4),round(x[2],4),round(x[3],4),'    ','erro
        if max(erro)<epsila:
            return
a=np.array([[6.4,2.6,-17,7],[-8.3,-23.1,-6.4,-7.5],[-3.7,3.1,-0.2,-0.3],[-3,4.2,-9
b=[-34.3,-89.1,30.2,140.1]
q3(a,b,[-6.5,2.9,3.1,7.5],0.001)
```

```
[[  6.4   2.6 -17.    7. ]
 [ -8.3 -23.1  -6.4  -7.5]
 [ -3.7   3.1  -0.2  -0.3]
 [ -3.    4.2  -9.3  18.3]] [-34.3, -89.1, 30.2, 140.1]
certo= [-6.50648039  2.90174055  3.09946574  7.49826666]
depois=
[[ -3.7   3.1  -0.2  -0.3]
 [ -8.3 -23.1  -6.4  -7.5]
 [  6.4   2.6 -17.    7. ]
 [ -3.    4.2  -9.3  18.3]] [30.2, -89.1, -34.3, 140.1]
resposta
0 -6.5 2.9 3.1 7.5     erro= 0.00810810810810736
1 -6.5081 2.9016 3.0995 7.4981     erro= 0.0015337056802824378
2 -6.5066 2.9018 3.0994 7.4982     erro= 0.00016941041148310632
```

In [28]:
```
# Felipe Turatti
```

In [29]:
```
a=np.array([[8.4,6.8,-6.1],[6.3,2.1,6.5],[0.8,-6.3,2.9]])
b=[-67.3,-78.9,-5.7]
q1(a,b)
```

```
[[ 8.4  6.8 -6.1]
 [ 6.3  2.1  6.5]
 [ 0.8 -6.3  2.9]] [-67.3, -78.9, -5.7]
certo= [-8.89154465 -1.60602703 -3.0016326 ]
matriz apos pivoteamento col 1
 [[ 8.4  6.8 -6.1]
 [ 6.3  2.1  6.5]
 [ 0.8 -6.3  2.9]] [-67.3, -78.9, -5.7]
matriz após zerar col 1
 [[ 8.40000000e+00  6.80000000e+00 -6.10000000e+00]
 [-8.88178420e-16 -3.00000000e+00  1.10750000e+01]
 [ 0.00000000e+00 -6.94761905e+00  3.48095238e+00]] [-67.3, -28.42500000000001, 0.
7095238095238088]
matriz após pivoteamento col 2
 [[ 8.40000000e+00  6.80000000e+00 -6.10000000e+00]
 [ 0.00000000e+00 -6.94761905e+00  3.48095238e+00]
 [-8.88178420e-16 -3.00000000e+00  1.10750000e+01]] [-67.3, 0.7095238095238088, -2
8.42500000000001]
matriz após zerar col 2
 [[ 8.40000000e+00  6.80000000e+00 -6.10000000e+00]
 [ 0.00000000e+00 -6.94761905e+00  3.48095238e+00]
 [-8.88178420e-16  0.00000000e+00  9.57191570e+00]] [-67.3, 0.7095238095238088, -2
8.731374228923933]
resposta
 [[ 8.4     6.8    -6.1   ]
 [ 0.     -6.9476  3.481 ]
 [-0.      0.      9.5719]] [-67.3      0.7095 -28.7314]
x =  -8.8915 -1.606 -3.0016
```

In [30]:
```
a=np.array([[9.9,19,8.6],[-11.2,-5.8,-3.5],[-0.1,0.1,1.9]])
b=[34.9,6.6,12.5]
q2(a,b,[-2.8,0.4,6.4],0.001)
```

```
[[  9.9  19.    8.6]
 [-11.2  -5.8  -3.5]
 [ -0.1   0.1   1.9]] [34.9, 6.6, 12.5]
certo= [-2.79556063  0.39156293  6.41120402]
depois=
[[-11.2  -5.8  -3.5]
 [  9.9  19.    8.6]
 [ -0.1   0.1   1.9]] [6.6, 34.9, 12.5]
resposta
1 -2.7964 0.3989 6.4105         erro= 0.010526315789474161
2 -2.7992 0.3923 6.4108         erro= 0.006625445191927426
3 -2.7958 0.3936 6.411         erro= 0.0033549804963537433
4 -2.7966 0.3918 6.4111         erro= 0.0018405793802083181
5 -2.7956 0.3921 6.4111         erro= 0.0009196837959462023
```

In [31]:
```
a=np.array([[18.6,-6,-2.6,8.7],[5.8,9.3,21.8,-5.1],[6.7,-9,-3.1,19],[-0.1,-10.3,-3
b=[64.6,15.7,138.4,74]
q3(a,b,[-0.3,-6,4.6,5.3],0.001)
```

```
[[ 18.6  -6.   -2.6   8.7]
 [  5.8   9.3  21.8  -5.1]
 [  6.7  -9.   -3.1  19. ]
 [ -0.1 -10.3  -3.7   5.5]] [64.6, 15.7, 138.4, 74]
certo= [-0.29798989 -6.00713392  4.60075282  5.29445582]
depois=
[[ 18.6  -6.   -2.6   8.7]
 [ -0.1 -10.3  -3.7   5.5]
 [  5.8   9.3  21.8  -5.1]
 [  6.7  -9.   -3.1  19. ]] [64.6, 74, 15.7, 138.4]
resposta
0 -0.3 -6 4.6 5.3     erro= 0.0038991544002504597
1 -0.2984 -6.0039 4.6008 5.2961     erro= 0.0023505173415268032
2 -0.2977 -6.0062 4.6007 5.2948     erro= 0.000696340670216955
```

In [32]:
```
# Luiza Paula
```

In [33]:
```
a=np.array([[3.9,-6.8,0.7],[7.5,-8.1,-4.2],[7.9,9.8,-5.3]])
b=[30.3,11.2,-15]
q1(a,b)
```

```
[[ 3.9 -6.8  0.7]
 [ 7.5 -8.1 -4.2]
 [ 7.9  9.8 -5.3]] [30.3, 11.2, -15]
certo= [ 4.49955206 -1.10355504  7.49653244]
matriz apos pivoteamento col 1
 [[ 7.9  9.8 -5.3]
 [ 7.5 -8.1 -4.2]
 [ 3.9 -6.8  0.7]] [-15, 11.2, 30.3]
matriz após zerar col 1
 [[  7.9          9.8         -5.3       ]
 [  0.         -17.40379747   0.83164557]
 [  0.         -11.63797468   3.3164557 ]] [-15, 25.440506329113923, 37.7050632911
3924]
matriz após pivoteamento col 2
 [[  7.9          9.8         -5.3       ]
 [  0.         -17.40379747   0.83164557]
 [  0.         -11.63797468   3.3164557 ]] [-15, 25.440506329113923, 37.7050632911
3924]
matriz após zerar col 2
 [[ 7.90000000e+00  9.80000000e+00 -5.30000000e+00]
 [ 0.00000000e+00 -1.74037975e+01  8.31645570e-01]
 [ 0.00000000e+00  1.77635684e-15  2.76033166e+00]] [-15, 25.440506329113923, 20.6
9291584842534]
resposta
 [[  7.9      9.8     -5.3   ]
 [  0.     -17.4038   0.8316]
 [  0.       0.       2.7603]] [-15.      25.4405  20.6929]
x =  4.4996 -1.1036 7.4965
```

In [34]:
```
a=np.array([[-0.4,6.3,-4.9],[7,-1.7,4.8],[-1,-0.1,-2.7]])
b=[25.6,40.4,-16.2]
q2(a,b,[4.9,7.4,3.9],0.001)
```

```
[[-0.4  6.3 -4.9]
 [ 7.  -1.7  4.8]
 [-1.  -0.1 -2.7]] [25.6, 40.4, -16.2]
certo= [4.88880096 7.41855888 3.91457154]
depois=
[[ 7.  -1.7  4.8]
 [-0.4  6.3 -4.9]
 [-1.  -0.1 -2.7]] [40.4, 25.6, -16.2]
resposta
1 4.8943 7.4079 3.9111         erro= 0.011111111111110183
2 4.8886 7.4162 3.9129         erro= 0.008279163517258326
3 4.8894 7.4173 3.9147         erro= 0.0018013680094806084
4 4.8884 7.4187 3.9144         erro= 0.0014493794954910655
5 4.8889 7.4184 3.9147         erro= 0.0005720755705800329
```

In [35]:
```
a=np.array([[23.9,-8,9,6.4],[-3.4,9.6,16.4,-1.7],[-6.2,6.3,6,19],[-8.4,21.5,-7.8,3
b=[339.6,38.5,-10.9,-267.8]
q3(a,b,[8.5,-6.5,8.1,1.8],0.001)
```

```
[[23.9 -8.   9.   6.4]
 [-3.4  9.6 16.4 -1.7]
 [-6.2  6.3  6.  19. ]
 [-8.4 21.5 -7.8  3.6]] [339.6, 38.5, -10.9, -267.8]
certo= [ 8.50348544 -6.49609951  8.09938991  1.79740512]
depois=
[[23.9 -8.   9.   6.4]
 [-8.4 21.5 -7.8  3.6]
 [-3.4  9.6 16.4 -1.7]
 [-6.2  6.3  6.  19. ]] [339.6, -267.8, 38.5, -10.9]
resposta
0 8.5 -6.5 8.1 1.8     erro= 0.003102446319116714
1 8.5013 -6.4972 8.0998 1.7969     erro= 0.0018368804119344162
2 8.5031 -6.496 8.0992 1.7973     erro= 0.0005212468630357847
```

In [36]:
```
# Vitor Kiemo
```

In [37]:
```
a=np.array([[5.3,-5.7,-8.5],[8.3,-9.6,-3.1],[8.4,-2.5,-3.7]])
b=[35.6,-32.4,40.3]
q1(a,b)
```

```
[[ 5.3 -5.7 -8.5]
 [ 8.3 -9.6 -3.1]
 [ 8.4 -2.5 -3.7]] [35.6, -32.4, 40.3]
certo= [ 4.10022831  9.50515183 -8.00566534]
matriz apos pivoteamento col 1
 [[ 8.4 -2.5 -3.7]
 [ 8.3 -9.6 -3.1]
 [ 5.3 -5.7 -8.5]] [40.3, -32.4, 35.6]
matriz após zerar col 1
 [[ 8.4        -2.5        -3.7       ]
 [ 0.         -7.1297619   0.55595238]
 [ 0.         -4.12261905 -6.16547619]] [40.3, -72.2202380952381, 10.1726190476190
51]
matriz após pivoteamento col 2
 [[ 8.4        -2.5        -3.7       ]
 [ 0.         -7.1297619   0.55595238]
 [ 0.         -4.12261905 -6.16547619]] [40.3, -72.2202380952381, 10.1726190476190
51]
matriz após zerar col 2
 [[ 8.4        -2.5        -3.7       ]
 [ 0.         -7.1297619   0.55595238]
 [ 0.          0.         -6.48694273]] [40.3, -72.2202380952381, 51.9322925363165
9]
resposta
 [[ 8.4    -2.5    -3.7   ]
 [ 0.     -7.1298  0.556 ]
 [ 0.      0.     -6.4869]] [ 40.3    -72.2202  51.9323]
x =  4.1002 9.5052 -8.0057
```

In [38]:
```
a=np.array([[-7.2,7.9,15.7],[0.8,7.4,-5.6],[13.1,-3,-8.7]])
b=[-59.8,-18.1,-61]
q2(a,b,[-9.5,-5.5,-5.4],0.001)
```

```
[[-7.2  7.9 15.7]
 [ 0.8  7.4 -5.6]
 [13.1 -3.  -8.7]] [-59.8, -18.1, -61]
certo= [-9.5010082  -5.50304194 -5.39702087]
depois=
[[13.1 -3.  -8.7]
 [ 0.8  7.4 -5.6]
 [-7.2  7.9 15.7]] [-61, -18.1, -59.8]
resposta
1 -9.5023 -5.5054 -5.3981         erro= 0.005405405405404906
2 -9.5023 -5.5037 -5.3964         erro= 0.001693607839375666
3 -9.5008 -5.5025 -5.3973         erro= 0.0014967280647084635
4 -9.501 -5.5032 -5.3972         erro= 0.0007959062275473272
```

In [39]:
```
a=np.array([[0.8,2.8,10.8,5.6],[-11.8,2.7,-5.3,-3.2],[4.1,2.3,-9.6,-17.9],[-8.7,-14
b=[18.4,-76,-75,-109.6]
q3(a,b,[7.7,6.2,-5.5,9.7],0.001)
```

```
[[  0.8   2.8  10.8   5.6]
 [-11.8   2.7  -5.3  -3.2]
 [  4.1   2.3  -9.6 -17.9]
 [ -8.7 -14.3  -1.5   3.9]] [18.4, -76, -75, -109.6]
certo= [ 7.70173695  6.20300003 -5.50694866  9.70450999]
depois=
[[-11.8   2.7  -5.3  -3.2]
 [ -8.7 -14.3  -1.5   3.9]
 [  0.8   2.8  10.8   5.6]
 [  4.1   2.3  -9.6 -17.9]] [-76, -109.6, 18.4, -75]
resposta
0 7.7 6.2 -5.5 9.7     erro= 0.004318500081211596
1 7.6992 6.2026 -5.5043 9.7025     erro= 0.001871153470817788
2 7.701 6.2026 -5.5057 9.7036     erro= 0.0007091628238420356
```

In [40]:
```
# Thaislaine Reis
```

In [41]:
```
a=np.array([[1.1,-6.9,-5.9],[1.7,-2.4,-4.5],[2.9,7.3,1.4]])
b=[36.2,11.6,-32.3]
q1(a,b)
```

```
[[ 1.1 -6.9 -5.9]
 [ 1.7 -2.4 -4.5]
 [ 2.9  7.3  1.4]] [36.2, 11.6, -32.3]
certo= [ 5.45973277 -7.23478049  3.34333753]
matriz apos pivoteamento col 1
 [[ 2.9  7.3  1.4]
 [ 1.7 -2.4 -4.5]
 [ 1.1 -6.9 -5.9]] [-32.3, 11.6, 36.2]
matriz após zerar col 1
 [[ 2.90000000e+00  7.30000000e+00  1.40000000e+00]
 [ 2.22044605e-16 -6.67931034e+00 -5.32068966e+00]
 [ 0.00000000e+00 -9.66896552e+00 -6.43103448e+00]] [-32.3, 30.534482758620683, 4
8.45172413793104]
matriz após pivoteamento col 2
 [[ 2.90000000e+00  7.30000000e+00  1.40000000e+00]
 [ 0.00000000e+00 -9.66896552e+00 -6.43103448e+00]
 [ 2.22044605e-16 -6.67931034e+00 -5.32068966e+00]] [-32.3, 48.45172413793104, 30.
534482758620683]
matriz após zerar col 2
 [[ 2.90000000e+00  7.30000000e+00  1.40000000e+00]
 [ 0.00000000e+00 -9.66896552e+00 -6.43103448e+00]
 [ 2.22044605e-16 -8.88178420e-16 -8.78138374e-01]] [-32.3, 48.45172413793104, -2.
9359129814550684]
resposta
 [[ 2.9     7.3     1.4   ]
 [ 0.     -9.669  -6.431 ]
 [ 0.     -0.     -0.8781]] [-32.3     48.4517  -2.9359]
x =  5.4597 -7.2348 3.3433
```

In [42]:
```
a=np.array([[6.7,-12.2,4.4],[13.7,-9.3,-2.6],[5+9,9.5,16.3]])
b=[-42.2,-127.8,147.8]
q2(a,b,[-5.1,3.8,8.7],0.001)
```

```
[[  6.7 -12.2   4.4]
 [ 13.7  -9.3  -2.6]
 [ 14.    9.5  16.3]] [-42.2, -127.8, 147.8]
certo= [-4.23385671  4.72287078  9.95133261]
depois=
[[ 13.7  -9.3  -2.6]
 [  6.7 -12.2   4.4]
 [ 14.    9.5  16.3]] [-127.8, -42.2, 147.8]
resposta
1 -5.0978 3.7959 11.2331         erro= 2.5331288343558303
2 -4.6199 4.7107 11.2336         erro= 0.9147900331157683
3 -3.9988 4.9734 10.29          erro= 0.9436756339760546
4 -3.9996 4.9741 9.6034          erro= 0.6865362087348164
5 -4.1293 4.7261 9.6037          erro= 0.24803387960311607
```

In [43]:
```
a=np.array([[-6.8,2.8,15.8,4.7],[-17.9,7.4,4.6,4.5],[2.1,7.9,3.1,13.5],[-8.1,18.4,
b=[-131.2,-0.4,-105.9,72.1]
q3(a,b,[-3.5,-0.2,-8.2,-5.3],0.001)
```

```
[[ -6.8   2.8  15.8   4.7]
 [-17.9   7.4   4.6   4.5]
 [  2.1   7.9   3.1  13.5]
 [ -8.1  18.4  -7.2   2.2]] [-131.2, -0.4, -105.9, 72.1]
certo= [-3.49808869 -0.19478529 -8.19700182 -5.30403736]
depois=
[[-17.9   7.4   4.6   4.5]
 [ -8.1  18.4  -7.2   2.2]
 [ -6.8   2.8  15.8   4.7]
 [  2.1   7.9   3.1  13.5]] [-0.4, 72.1, -131.2, -105.9]
resposta
0 -3.5 -0.2 -8.2 -5.3     erro= 0.00271739130434806
1 -3.5 -0.1973 -8.1986 -5.3019     erro= 0.0012264451186182468
2 -3.499 -0.1961 -8.1978 -5.303     erro= 0.0006287879961298648
```

In [44]:
```
# Luiz Costa
```

In [45]:
```
a=np.array([[4.2,3.1,9.1],[1.5,7.3,7.4],[0.7,-8.7,8.1]])
b=[-41.3,-2.3,-133.8]
q1(a,b)
```

```
[[ 4.2  3.1  9.1]
 [ 1.5  7.3  7.4]
 [ 0.7 -8.7  8.1]] [-41.3, -2.3, -133.8]
certo= [ 2.88612115  7.70280735 -8.49455072]
matriz apos pivoteamento col 1
 [[ 4.2  3.1  9.1]
 [ 1.5  7.3  7.4]
 [ 0.7 -8.7  8.1]] [-41.3, -2.3, -133.8]
matriz após zerar col 1
 [[ 4.2         3.1         9.1       ]
 [ 0.          6.19285714  4.15      ]
 [ 0.         -9.21666667  6.58333333]] [-41.3, 12.45, -126.91666666666669]
matriz após pivoteamento col 2
 [[ 4.2         3.1         9.1       ]
 [ 0.         -9.21666667  6.58333333]
 [ 0.          6.19285714  4.15      ]] [-41.3, -126.91666666666669, 12.45]
matriz após zerar col 2
 [[ 4.2         3.1         9.1       ]
 [ 0.         -9.21666667  6.58333333]
 [ 0.          0.          8.57346939]] [-41.3, -126.91666666666669, -72.827770601
91168]
resposta
 [[ 4.2     3.1     9.1   ]
 [ 0.     -9.2167  6.5833]
 [ 0.      0.      8.5735]] [ -41.3    -126.9167  -72.8278]
x =  2.8861 7.7028 -8.4946
```

In [46]:
```
a=np.array([[-4.5,-3.6,-0.5],[-6.1,-9.3,2.2],[1.6,-2.7,-6.1]])
b=[-48,-80.5,-60.8]
q2(a,b,[3.3,8.2,7.2],0.001)
```

```
[[-4.5 -3.6 -0.5]
 [-6.1 -9.3  2.2]
 [ 1.6 -2.7 -6.1]] [-48, -80.5, -60.8]
certo= [3.31507432 8.18773171 7.21266283]
depois=
[[-4.5 -3.6 -0.5]
 [-6.1 -9.3  2.2]
 [ 1.6 -2.7 -6.1]] [-48, -80.5, -60.8]
resposta
1 3.3067 8.1946 7.2033         erro= 0.006666666666666821
2 3.3106 8.191 7.2074         erro= 0.004128327163757994
3 3.313 8.1894 7.21         erro= 0.0026247809824901935
4 3.314 8.1885 7.2114         erro= 0.001345169960551118
5 3.3146 8.1881 7.2121         erro= 0.0006878800521210593
```

In [47]:
```
a=np.array([[24.1,-7.8,-6.5,-8.2],[8.4,8.8,3.6,-22.3],[-2.9,7.5,-13.6,2.1],[0.4,7.4
b=[-71.6,-97.4,170.5,86.7]
q3(a,b,[-1.8,5.7,-8.3,4.6],0.001)
```

```
[[ 24.1  -7.8  -6.5  -8.2]
 [  8.4   8.8   3.6 -22.3]
 [ -2.9   7.5 -13.6   2.1]
 [  0.4   7.4  -5.5  -0.1]] [-71.6, -97.4, 170.5, 86.7]
certo= [-1.79174616  5.71105092 -8.29375994  4.60758047]
depois=
[[ 24.1  -7.8  -6.5  -8.2]
 [  0.4   7.4  -5.5  -0.1]
 [ -2.9   7.5 -13.6   2.1]
 [  8.4   8.8   3.6 -22.3]] [-71.6, 86.7, 170.5, -97.4]
resposta
0 -1.8 5.7 -8.3 4.6     erro= 0.006734327688683983
1 -1.7996 5.7067 -8.2956 4.6026     erro= 0.004247423029807296
2 -1.7953 5.7098 -8.2945 4.6056     erro= 0.00232238965780307
3 -1.793 5.7106 -8.2941 4.6069     erro= 0.0007909508255989905
```

In [48]:
```
# Gabriel Ferrarini
```

In [49]:
```
a=np.array([[-3.5,0.5,-7.1],[-1.9,0.4,-1.5],[4.1,-1,7.1]])
b=[72,19.4,-76.4]
q1(a,b)
```

```
[[-3.5  0.5 -7.1]
 [-1.9  0.4 -1.5]
 [ 4.1 -1.   7.1]] [72, 19.4, -76.4]
certo= [-1.97976274  6.42428472 -8.71249128]
matriz apos pivoteamento col 1
 [[ 4.1 -1.   7.1]
 [-1.9  0.4 -1.5]
 [-3.5  0.5 -7.1]] [-76.4, 19.4, 72]
matriz após zerar col 1
 [[ 4.1         -1.           7.1        ]
 [ 0.         -0.06341463  1.7902439 ]
 [ 0.         -0.35365854 -1.03902439]] [-76.4, -16.00487804878049, 6.780487804878
035]
matriz após pivoteamento col 2
 [[ 4.1         -1.           7.1        ]
 [ 0.         -0.35365854 -1.03902439]
 [ 0.         -0.06341463  1.7902439 ]] [-76.4, 6.780487804878035, -16.00487804878
049]
matriz após zerar col 2
 [[ 4.1         -1.           7.1        ]
 [ 0.         -0.35365854 -1.03902439]
 [ 0.          0.          1.97655172]] [-76.4, 6.780487804878035, -17.22068965517
2414]
resposta
 [[ 4.1    -1.      7.1   ]
 [ 0.     -0.3537 -1.039 ]
 [ 0.      0.      1.9766]] [-76.4      6.7805 -17.2207]
x =  -1.9798 6.4243 -8.7125
```

In [50]:
```
a=np.array([[-0.3,-3.2,1.9],[6.5,3,-2.7],[-9.9,2.1,-13.7]])
b=[3.4,26.7,-183.1]
q2(a,b,[6.2,4,9.5],0.001)
```

```
[[ -0.3  -3.2   1.9]
 [  6.5   3.   -2.7]
 [ -9.9   2.1 -13.7]] [3.4, 26.7, -183.1]
certo= [6.20813904 3.99045052 9.49046493]
depois=
[[  6.5   3.   -2.7]
 [ -0.3  -3.2   1.9]
 [ -9.9   2.1 -13.7]] [26.7, 3.4, -183.1]
resposta
1 6.2077 3.9969 9.4978         erro= 0.007692307692306777
2 6.2082 3.9949 9.4918         erro= 0.006037689500280052
3 6.2066 3.9912 9.4911         erro= 0.0036348193606112744
4 6.208 3.991 9.4917          erro= 0.001389004583343123
5 6.2084 3.9912 9.4906         erro= 0.0010443338469610097
```

In [51]:
```
a=np.array([[-1.8,13.7,8,3.8],[-12.9,0.2,8.6,2.4],[2.3,-9.4,5.8,-17.6],[5.9,4.9,-2
b=[-57,102.6,94.4,-101.1]
q3(a,b,[-9.3,-3.7,-0.6,-4.8],0.001)
```

```
[[ -1.8  13.7   8.    3.8]
 [-12.9   0.2   8.6   2.4]
 [  2.3  -9.4   5.8 -17.6]
 [  5.9   4.9 -21.1   8.5]] [-57, 102.6, 94.4, -101.1]
certo= [-9.30856031 -3.69713035 -0.60567305 -4.80508904]
depois=
[[-12.9   0.2   8.6   2.4]
 [ -1.8  13.7   8.    3.8]
 [  5.9   4.9 -21.1   8.5]
 [  2.3  -9.4   5.8 -17.6]] [102.6, -57, -101.1, 94.4]
resposta
0 -9.3 -3.7 -0.6 -4.8     erro= 0.0038759689922471807
1 -9.3039 -3.7012 -0.6033 -4.8015     erro= 0.0024745507358634455
2 -9.3064 -3.6992 -0.6041 -4.8032     erro= 0.0008357632972444407
```

In [52]:
```
# Douglas Magalhães
```

In [53]:
```
a=np.array([[-2.2,1.8,-0.9],[3.9,2.1,3.7],[9.2,-0.1,-3.6]])
b=[7.2,-24.8,49]
q1(a,b)
```

```
[[-2.2  1.8 -0.9]
 [ 3.9  2.1  3.7]
 [ 9.2 -0.1 -3.6]] [7.2, -24.8, 49]
certo= [ 1.70061865  1.42618296 -9.30470188]
matriz apos pivoteamento col 1
 [[ 9.2 -0.1 -3.6]
 [ 3.9  2.1  3.7]
 [-2.2  1.8 -0.9]] [49, -24.8, 7.2]
matriz após zerar col 1
 [[ 9.20000000e+00 -1.00000000e-01 -3.60000000e+00]
 [-4.44089210e-16  2.14239130e+00  5.22608696e+00]
 [ 0.00000000e+00  1.77608696e+00 -1.76086957e+00]] [49, -45.571739130434786, 18.9
17391304347827]
matriz após pivoteamento col 2
 [[ 9.20000000e+00 -1.00000000e-01 -3.60000000e+00]
 [-4.44089210e-16  2.14239130e+00  5.22608696e+00]
 [ 0.00000000e+00  1.77608696e+00 -1.76086957e+00]] [49, -45.571739130434786, 18.9
17391304347827]
matriz após zerar col 2
 [[ 9.20000000e+00 -1.00000000e-01 -3.60000000e+00]
 [-4.44089210e-16  2.14239130e+00  5.22608696e+00]
 [ 3.68159193e-16  0.00000000e+00 -6.09340436e+00]] [49, -45.571739130434786, 56.6
9731100963979]
resposta
 [[ 9.2    -0.1    -3.6   ]
 [-0.      2.1424  5.2261]
 [ 0.      0.     -6.0934]] [ 49.     -45.5717  56.6973]
x =  1.7006 1.4262 -9.3047
```

In [55]:
```
a=np.array([[1.9,1.4,4.6],[-7.1,3.3,-2.3],[-3.8,-13.6,-9.6]])
b=[-47,-12.2,165.6]
q2(a,b,[1.7,-6.3,-9],0.001)
```

```
[[  1.9   1.4   4.6]
 [ -7.1   3.3  -2.3]
 [ -3.8 -13.6  -9.6]] [-47, -12.2, 165.6]
certo= [ 1.7099061  -6.29601937 -9.00747706]
depois=
[[ -7.1   3.3  -2.3]
 [ -3.8 -13.6  -9.6]
 [  1.9   1.4   4.6]] [-12.2, 165.6, -47]
resposta
1 1.7056 -6.2985 -9.0022         erro= 0.005633802816901623
2 1.707 -6.2986 -9.0049         erro= 0.0027745758438104673
3 1.7079 -6.297 -9.0055         erro= 0.001570773747342713
4 1.7088 -6.2968 -9.0064         erro= 0.0009118546334376809
```

In [56]:
```
a=np.array([[-5.3,9.5,-3.2,19.8],[0.6,5.9,15.9,8.2],[-4.4,-22.8,9.2,-8.9],[12.5,-0
b=[172.1,114.3,-178.1,125.8]
q3(a,b,[6,3.7,1.1,8.7],0.001)
```

```
[[ -5.3   9.5  -3.2  19.8]
 [  0.6   5.9  15.9   8.2]
 [ -4.4 -22.8   9.2  -8.9]
 [ 12.5  -0.3   6.1   5.2]] [172.1, 114.3, -178.1, 125.8]
certo= [5.99653528 3.7035809  1.10223529 8.69822077]
depois=
[[ 12.5  -0.3   6.1   5.2]
 [ -4.4 -22.8   9.2  -8.9]
 [  0.6   5.9  15.9   8.2]
 [ -5.3   9.5  -3.2  19.8]] [125.8, -178.1, 114.3, 172.1]
resposta
0 6 3.7 1.1 8.7     erro= 0.0031999999999987594
1 5.9968 3.7019 1.1019 8.699     erro= 0.0012465891113113692
2 5.9963 3.7032 1.102 8.6983     erro= 0.0003021325656504814
```

In [ ]: