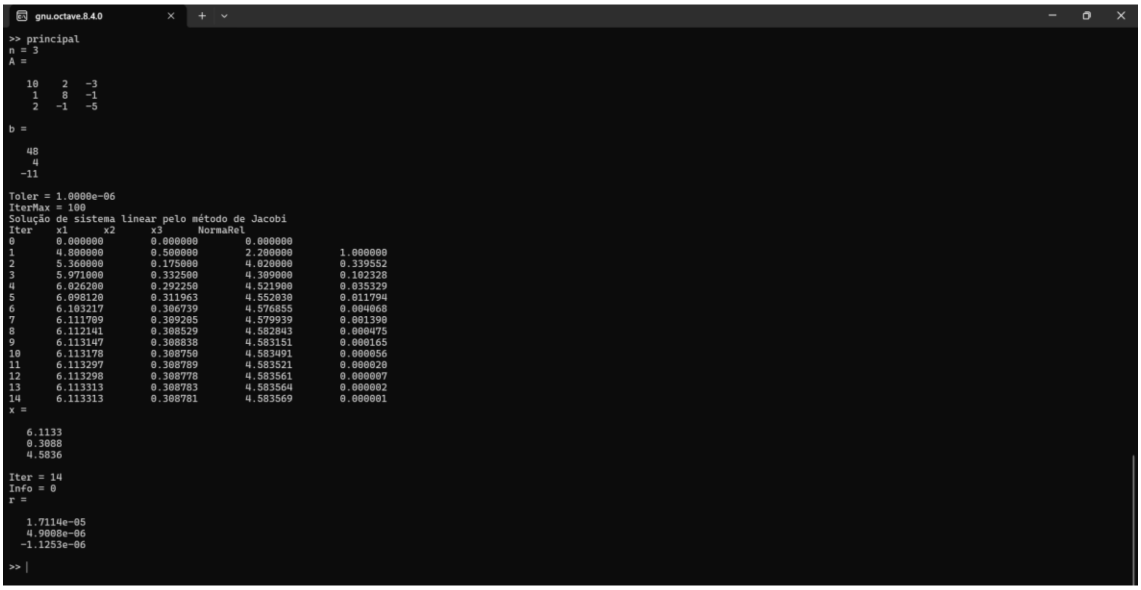
Exercício de Programação 2:
Métodos de Jacobi e de Gauss-Seidel

1) Soluções dos Sistemas

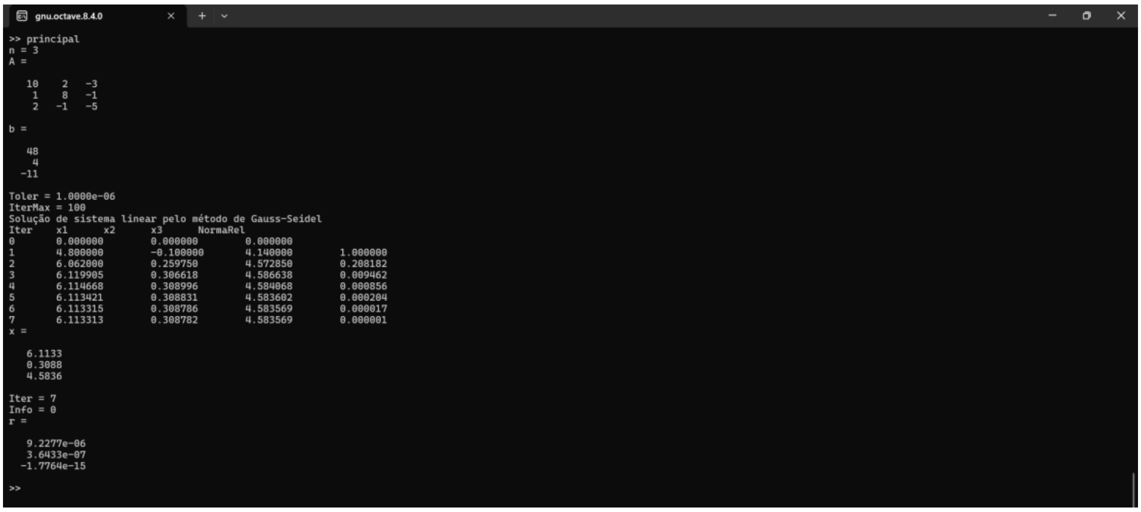
a) $$\begin{bmatrix} 10 & 2 & -3 \\ 1 & 8 & -1 \\ 2 & -1 & -5 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 48 \\ 4 \\ -11 \end{bmatrix}$$

i) Jacobi



```
gnu.octave.8.4.0
>> principal
n = 3
A =

   10    2   -3
    1    8   -1
    2   -1   -5

b =

   48
    4
  -11

Toler = 1.0000e-06
IterMax = 100
Solução de sistema linear pelo método de Jacobi
Iter    x1      x2      x3      NormaRel
0       0.000000        0.000000        0.000000
1       4.800000        0.500000        2.200000        1.000000
2       5.360000        0.175000        4.020000        0.339552
3       5.971000        0.332500        4.309000        0.102328
4       6.026200        0.292250        4.521900        0.035329
5       6.098120        0.311963        4.552030        0.011794
6       6.103217        0.306739        4.576855        0.004068
7       6.111709        0.309205        4.579939        0.001390
8       6.112141        0.308529        4.582843        0.000475
9       6.113147        0.308838        4.583151        0.000165
10      6.113178        0.308750        4.583491        0.000056
11      6.113297        0.308789        4.583521        0.000020
12      6.113298        0.308778        4.583561        0.000007
13      6.113313        0.308783        4.583564        0.000002
14      6.113313        0.308781        4.583569        0.000001
x =

   6.1133
   0.3088
   4.5836

Iter = 14
Info = 0
r =

   1.7114e-05
   4.9008e-06
  -1.1253e-06

>> |
```

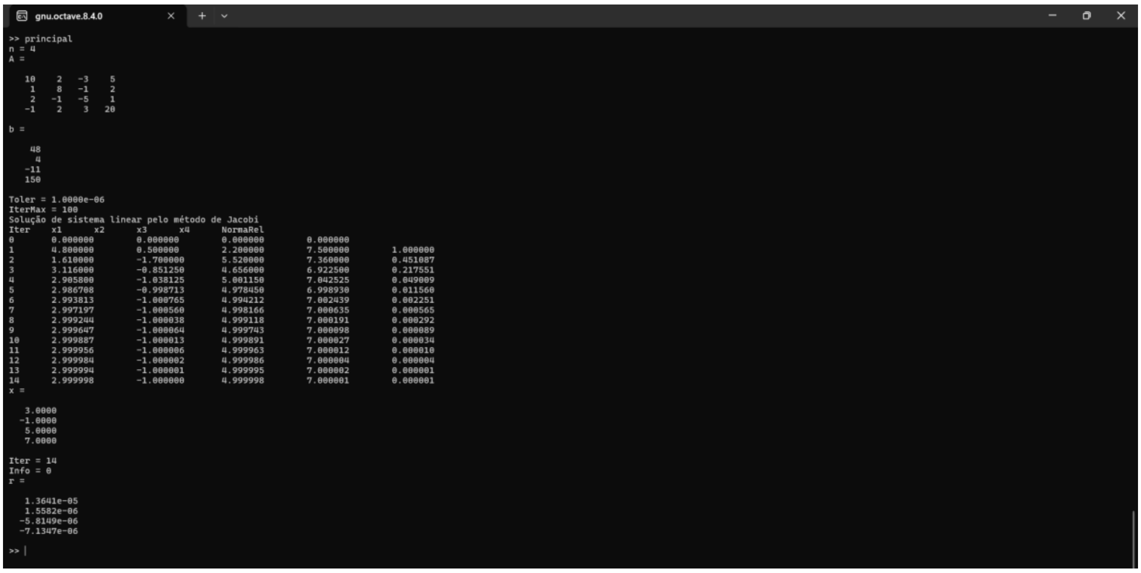
ii) Gauss-Seidel



```
gnu.octave.8.4.0
>> principal
n = 3
A =

   10    2   -3
    1    8   -1
    2   -1   -5

b =

   48
    4
  -11

Toler = 1.0000e-06
IterMax = 100
Solução de sistema linear pelo método de Gauss-Seidel
Iter    x1      x2      x3      NormaRel
0       0.000000        0.000000        0.000000
1       4.800000        -0.100000       4.140000        1.000000
2       6.062000        0.259750        4.572850        0.208182
3       6.119905        0.306618        4.586638        0.009462
4       6.114668        0.308996        4.584068        0.000856
5       6.113421        0.308831        4.583602        0.000204
6       6.113315        0.308786        4.583569        0.000017
7       6.113313        0.308782        4.583569        0.000001
x =

   6.1133
   0.3088
   4.5836

Iter = 7
Info = 0
r =

   9.2277e-06
   3.6433e-07
  -1.7764e-15

>>
```

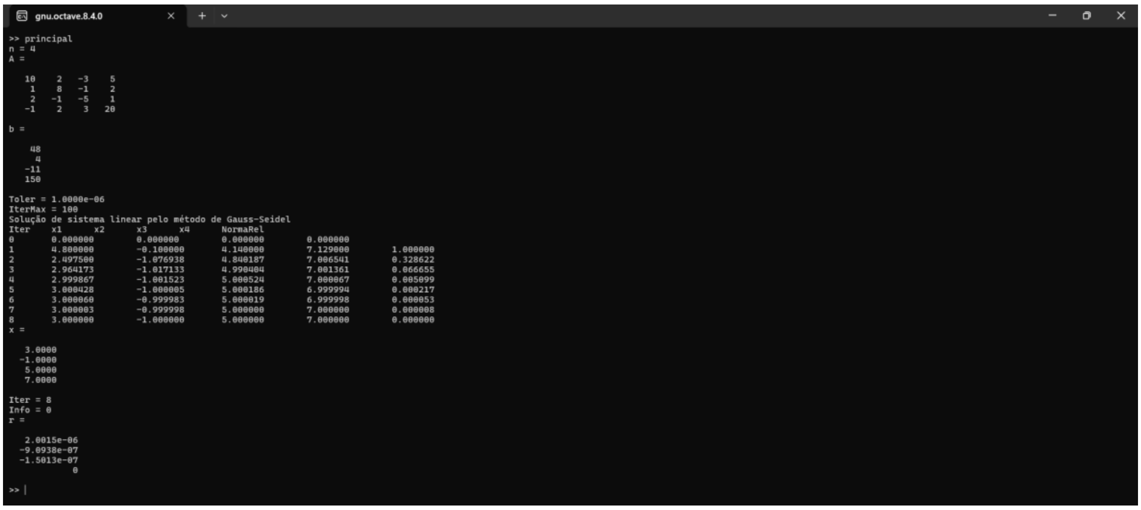
b) $$\begin{bmatrix} 10 & 2 & -3 & 5 \\ 1 & 8 & -1 & 2 \\ 2 & -1 & -5 & 1 \\ -1 & 2 & 3 & 20 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 48 \\ 4 \\ -11 \\ 150 \end{bmatrix}$$

i) Jacobi



```
gnu.octave.8.4.0
>> principal
n = 4
A =

   10    2   -3    5
    1    8   -1    2
    2   -1   -5    1
   -1    2    3   20

b =

    48
     4
   -11
   150

Toler = 1.0000e-06
IterMax = 100
Solução de sistema linear pelo método de Jacobi
Iter    x1      x2      x3      x4      NormaRel
0       0.000000        0.000000        0.000000        0.000000
1       4.800000        0.500000        2.200000        7.500000        1.000000
2       1.610000        -1.700000       5.520000        7.360000        0.451087
3       3.116000        -0.851250       4.656000        6.922500        0.217551
4       2.905800        -1.038125       5.001150        7.042525        0.049009
5       2.986708        -0.998713       4.978450        6.998930        0.011560
6       2.993813        -1.000765       4.994212        7.002439        0.002251
7       2.997197        -1.000560       4.998166        7.000635        0.000565
8       2.999244        -1.000038       4.999118        7.000191        0.000292
9       2.999647        -1.000064       4.999743        7.000098        0.000089
10      2.999887        -1.000013       4.999891        7.000027        0.000034
11      2.999956        -1.000006       4.999963        7.000012        0.000010
12      2.999984        -1.000002       4.999986        7.000004        0.000004
13      2.999994        -1.000001       4.999995        7.000002        0.000001
14      2.999998        -1.000000       4.999998        7.000001        0.000001
x =

   3.0000
  -1.0000
   5.0000
   7.0000

Iter = 14
Info = 0
r =

   1.3641e-05
   1.5582e-06
  -5.8149e-06
  -7.1347e-06

>> |
```

ii) Gauss-Seidel



```
gnu.octave.8.4.0
>> principal
n = 4
A =

   10    2   -3    5
    1    8   -1    2
    2   -1   -5    1
   -1    2    3   20

b =

    48
     4
   -11
   150

Toler = 1.0000e-06
IterMax = 100
Solução de sistema linear pelo método de Gauss-Seidel
Iter    x1      x2      x3      x4      NormaRel
0       0.000000        0.000000        0.000000        0.000000
1       4.800000        -0.100000       4.140000        7.129000        1.000000
2       2.497500        -1.076938       4.840187        7.006541        0.328622
3       2.964173        -1.017133       4.990404        7.001361        0.066655
4       2.999867        -1.001523       5.000524        7.000067        0.005099
5       3.000428        -1.000005       5.000186        6.999994        0.000217
6       3.000060        -0.999983       5.000019        6.999998        0.000053
7       3.000003        -0.999998       5.000000        7.000000        0.000008
8       3.000000        -1.000000       5.000000        7.000000        0.000000
x =

   3.0000
  -1.0000
   5.0000
   7.0000

Iter = 8
Info = 0
r =

   2.0015e-06
  -9.0938e-07
  -1.5013e-07
            0

>> |
```

2) Comparação dos Números de Iterações Gastas e dos Vetores Resíduos
Norma de Máxima Coluna do Vetor Resíduo, $|r|_\infty$

Sistema	Número de Iterações Gastas, k		Norma de Máxima Coluna do Vetor Resíduo, $|r|_\infty$	
	Jacobi	Gauss-Seidel	Jacobi	Gauss-Seidel
a				
b				