

Resultados dos Algoritmos de Jacobi e de Gauss-Seidel

1) Resolver os sistemas lineares a seguir, utilizando os programas implementados, com Toler = 10^{-3} ou IterMax = 10.

a) $A = \begin{bmatrix} 10 & 2 & -3 & 5 \\ 1 & 8 & -1 & 2 \\ 2 & -1 & -5 & 1 \\ -1 & 2 & 3 & 20 \end{bmatrix}$
 $b = [48; 4; -11; 150]$

a) Jacobi

```
>> 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-03
IterMax = 10
Solução de sistema linear pelo método de Jacobi
Iter  x1    x2    x3    x4    NormRel
0      4.800000    0.500000    2.200000    7.500000
1      1.610000   -1.700000    5.520000    7.360000    0.451087
2      3.116000   -0.851250    4.656000    6.922500    0.217551
3      2.905800   -1.038125    5.001150    7.042525    0.049009
4      2.986708   -0.998713    4.978450    6.998930    0.011560
5      2.993813   -1.000765    4.994212    7.002439    0.002251
6      2.997197   -1.000560    4.998166    7.000635    0.000565

x =
    2.9972   -1.0006    4.9982    7.0006

Iter = 6
Info = 0
r =
    2.0472e-02
    4.1774e-03
   -4.7601e-03
   -8.8880e-03

>> |
```

Name	Class	Dimension	Value	Attribute
A	double	4x4	[10, 2, -3, 5; ...]	
Info	double	1x1	0	
Iter	double	1x1	6	
IterMax	double	1x1	10	
Toler	double	1x1	1.0000e-03	
b	double	4x1	[48; 4; -11; ...]	
n	double	1x1	4	
r	double	4x1	[0.020472; ...]	

```
run "/home/jjsbastos/Downloads/lagrange/principal.m"
run "/home/jjsbastos/Downloads/lagrange/principal.m"
run "/home/jjsbastos/Downloads/lagrange/principal.m"
run "/home/jjsbastos/Downloads/lagrange/principal.m"
# Octave 6.4.0, Fri May 10 16:59:22 2024 -03 <jjsbastos@labgrad>
run "/home/jjsbastos/Documents/principal.m"
run "/home/jjsbastos/Documents/principal.m"
run "/home/jjsbastos/Documents/principal.m"
run "/home/jjsbastos/Documents/principal.m"
run "/home/jjsbastos/Documents/principal.m"
exit
```

a) Gauss-Seidel

The screenshot shows the Octave environment with the following components:

- File Browser:** Displays the current directory structure, including files `gauss_seidel.m` and `principal.m`.
- Command Window:** Shows the execution of the `principal` script. It defines the matrix `A` and vector `b`, sets the tolerance (`Toler = 1.0000e-03`) and maximum iterations (`IterMax = 10`), and displays the solution vector `x` and the residual `r`.
- Workspace:** A table listing the variables defined in the workspace, including their names, classes, dimensions, and values.
- Command History:** A list of commands entered in the Command Window, including `run`, `exit`, and `principal`.

Workspace Variables:

Name	Class	Dimension	Value	At
A	double	4x4	[10. 2. -3. 5. ...]	0
Info	double	1x1	0	1
Iter	double	1x1	4	2
IterMax	double	1x1	10	3
Toler	double	1x1	1.0000e-03	4
b	double	4x1	[48; 4; -11; 150]	5
n	double	1x1	4	6
r	double	4x1	[-0.011910; ...]	7
x	double	1x4	[3.0013, -0.9...	8

Command Window Output:

```
>> principal
n = 4
A =
    10     2     -3     5
     1     0     -1     2
     2     -1     -5     1
    -1     2     3    20

b =
    48
     4
    -11
    150

Toler = 1.0000e-03
IterMax = 10
Solução de sistema linear pelo método de Gauss-Seidel
Iter   x1      x2      x3      x4      NormaRel
0      4.8000000  0.5000000  2.2000000  7.5000000
1      1.6100000  -1.3012500  4.6042500  7.0199888
2      2.9315311  -1.0459070  4.9857910  7.0032999
3      3.0032700  -1.0030090  5.0025690  7.0000790
4      3.0013333  -0.9998650  5.0005220  6.9999750

x =
    3.0013   -0.9999    5.0005    7.0000

Iter = 4
Info = 0
r =
   -0.011910
   -0.001839
    0.000104
         0

>>
```

b) $A = \begin{bmatrix} 1 & 2 & -2 \\ 1 & 1 & 1 \\ 2 & 2 & 1 \end{bmatrix}$
 $b = [-1; 6; 9]$

b) Jacobi

The screenshot shows the Octave environment with the following components:

- File Browser:** Displays the current directory `/home/jjsbastos/Documents` containing files `principal.m` and `jacobi.m`.
- Command Window:** Shows the execution of the `principal` script. It defines `n = 3`, matrix `A`, and vector `b`. It then performs a Jacobi iteration, displaying the iteration table below.
- Workspace:** A table listing the variables in the current workspace.
- Command History:** A list of previously executed commands.

Iteration Table:

Iter	x1	x2	x3	NormaRel
0	-1.000000	6.000000	9.000000	2.000000
1	5.000000	-2.000000	-1.000000	1.333333
2	1.000000	2.000000	3.000000	0.000000
3	1.000000	2.000000	3.000000	0.000000

Workspace Table:

Name	Class	Dimension	Value
A	double	3x3	[1, 2, -2; 1, ...]
Info	double	1x1	0
Iter	double	1x1	3
IterMax	double	1x1	10
Toler	double	1x1	1.0000e-03
b	double	3x1	[-1; 6; 9]

Command History:

```
run "/home/jjsbastos/Downloads/lagrange/principal.m"
run "/home/jjsbastos/Downloads/lagrange/principal.m"
run "/home/jjsbastos/Downloads/lagrange/principal.m"
run "/home/jjsbastos/Downloads/lagrange/principal.m"
# Octave 6.4.0, Fri May 10 16:59:22 2024 -03 <jjsbastos
run "/home/jjsbastos/Documents/principal.m"
run "/home/jjsbastos/Documents/principal.m"
run "/home/jjsbastos/Documents/principal.m"
run "/home/jjsbastos/Documents/principal.m"
```

b) Gauss-Seidel

The screenshot shows the Octave software interface with the following components:

- File Browser:** Displays the current directory structure, including `gauss_seidel.m` and `principal.m`.
- Command Window:** Shows the execution of the `principal` script. The matrix `A` is defined as:
$$A = \begin{bmatrix} 1 & 2 & -2 \\ 1 & 1 & 1 \\ 2 & 2 & 1 \end{bmatrix}$$
The vector `b` is defined as:
$$b = \begin{bmatrix} -1 \\ 6 \\ 9 \end{bmatrix}$$
The tolerance is set to `Toler = 1.0000e-03` and the maximum number of iterations is `IterMax = 10`. The solution is calculated using the Gauss-Seidel method, and the results are displayed in a table:

Iter	x1	x2	x3	NormaRel
0	-1.000000	6.000000	9.000000	
1	5.000000	-8.000000	15.000000	0.933333
2	45.000000	-54.000000	27.000000	0.851852
3	161.000000	-182.000000	51.000000	0.703297
4	465.000000	-510.000000	99.000000	0.643137
5	1217.000000	-1310.000000	195.000000	0.610687
6	3009.000000	-3198.000000	387.000000	0.590369
7	7169.000000	-7550.000000	771.000000	0.576424
8	16641.000000	-17406.000000	1539.000000	0.566242
9	37889.000000	-39422.000000	3075.000000	0.558470
10	84993.000000	-88062.000000	6147.000000	0.552338

The final solution vector `x` is displayed as:
$$x = \begin{bmatrix} 84993 & -88062 & 6147 \end{bmatrix}$$
- Workspace:** Shows the variables defined in the script, including `A`, `b`, `Toler`, `IterMax`, `n`, `r`, and `x`.
- Command History:** Shows the sequence of commands executed, including `run`, `principal`, `clc`, and `exit`.

c) $A = \begin{bmatrix} 1 & -0.5 & 0.5 \\ 1 & 1 & 1 \\ -0.5 & -0.5 & 1 \end{bmatrix}$
 $b = \begin{bmatrix} 3 \\ 12 \\ 3 \end{bmatrix}$

c) Jacobi

The screenshot shows the Octave environment with the following components:

- File Browser:** Shows the current directory as `/home/jjsbastos/Documents` with files `principal.m` and `Jacobi.m`.
- Command Window:**

```
>> principal
n = 3
A =
    1.0000   -0.5000    0.5000
    1.0000    1.0000    1.0000
   -0.5000   -0.5000    1.0000

b =
     3
    12
     3

Toler = 1.0000e-03
IterMax = 10
Solução de sistema linear pelo método de Jacobi
Iter  x1    x2    x3  NormaRel
0      3.000000  12.000000  3.000000
1      7.500000  6.000000  10.500000  0.714286
2      0.750000  -6.000000  9.750000  1.230769
3      4.875000  1.500000  0.375000  1.923077
4      3.562500  16.500000  1.312500  0.909091
5     10.593750  7.125000  13.031250  0.899281
6      0.046875  -11.625000  11.059375  1.581028
7     -8.742188  0.093750  -2.789062  1.675603
8      4.441406  23.531250  -1.324219  0.996016
9     15.427734  0.882012  16.986328  1.077958
10     -1.051758  -20.414062  15.155273  1.435132

x =
   -1.0518   -20.4141   15.1553

Iter = 10
Info = 1
r =
   -13.733
    18.311
   -22.888

>>
```
- Workspace:** A table showing the variables defined in the script:

Name	Class	Dimension	Value	Attributes
A	double	3x3	[1, -0.5000, ...]	
Info	double	1x1	1	
Iter	double	1x1	10	
IterMax	double	1x1	10	
Toler	double	1x1	1.0000e-03	
b	double	3x1	[3; 12; 3]	
- Command History:** Shows the sequence of commands executed, including running the `principal.m` script multiple times.

c) Gauss-Seidel

The screenshot shows the Octave software interface with the following components:

- File Browser:** Displays the current directory structure, including files like `gauss_seidel.m` and `principal.m`.
- Command Window:** Shows the execution of the `principal` script. It defines the matrix `A`, vector `b`, and the tolerance `Toler`. It then displays the iterative solution process using the Gauss-Seidel method, showing the values of `x1`, `x2`, and `x3` for each iteration.
- Workspace:** A table showing the variables defined in the workspace, including their class, dimension, and value.
- Command History:** A list of commands entered in the Command Window, including `run`, `principal`, and `clc`.

Command Window Output:

```
>> principal
n = 3
A =
    1.0000    -0.5000     0.5000
    1.0000     1.0000     1.0000
   -0.5000    -0.5000     1.0000

b =
     3
    12
     3

Toler = 1.0000e-03
IterMax = 10
Solução de sistema linear pelo método de Gauss-Seidel
Iter  x1  x2  x3  NormaRel
0      3.000000    12.000000    3.000000
1      7.500000    1.500000    7.500000
2      0.000000    4.500000    5.250000
3      2.625000    4.125000    6.375000
4      1.875000    3.750000    5.812500
5      1.968750    4.218750    6.093750
6      2.062500    3.843750    5.953125
7      1.945312    4.101562    6.023438
8      2.039062    3.937500    5.988281
9      1.974609    4.037109    6.005859
10     2.015625    3.978516    5.997070
x =
    2.0156    3.9785    5.9971

Iter = 10
Info = 1
r =
   -0.024902
    0.008789
         0

>> |
```

Workspace Table:

Name	Class	Dimension	Value	At
A	double	3x3	[1, -0.5000, 0...	
Info	double	1x1	1	
Iter	double	1x1	10	
IterMax	double	1x1	10	
Toler	double	1x1	1.0000e-03	
b	double	3x1	[3; 12; 3]	
n	double	1x1	3	
r	double	3x1	[-0.024902; 8...	
x	double	1x3	[2.0156, 3.97...	

Command History:

```
run "/mnt/c/Users/jjako/Documents/programacao/ufes/...
run "/mnt/c/Users/jjako/Documents/programacao/ufes/...
exit
# Octave 6.4.0, Sun May 19 18:46:26 2024 -03 <jjako
principal
clc
principal
clc
principal
clc
principal
clc
principal
```

2) Comparar os números gastos de iterações, Iter, as informações sobre a convergência, Info, e os resíduos, $r = b - A * x$, das soluções dos sistemas lineares acima.

a) $A = [10 \ 2 \ -3 \ 5; \ 1 \ 8 \ -1 \ 2; \ 2 \ -1 \ -5 \ 1; \ -1 \ 2 \ 3 \ 20]$
 $b = [48; 4; -11; 150]$

Método	Iter	Info	r
Jacobi	Iter = 6	Info = 0	$r =$ $2.0472e-02$ $4.1774e-03$ $-4.7601e-03$ $-8.8880e-03$
Gauss-Seidel	Iter = 4	Info = 0	$r =$ -0.011910 -0.001839 0.000104 0

b) $A = [1 \ 2 \ -2; \ 1 \ 1 \ 1; \ 2 \ 2 \ 1]$
 $b = [-1; 6; 9]$

Método	Iter	Info	r
Jacobi	Iter = 3	Info = 0	$r =$ 0 0 0
Gauss-Seidel	Iter = 10	Info = 1	$r =$ 103424 -3072 0

c) $A = [1 \ -0.5 \ 0.5; \ 1 \ 1 \ 1; \ -0.5 \ -0.5 \ 1]$
 $b = [3; 12; 3]$

Método	Iter	Info	r
Jacobi	Iter = 10	Info = 1	$r =$ -13.733 18.311 -22.888
Gauss-Seidel	Iter = 10	Info = 1	$r =$ -0.024902 0.008789 0

