

---

## PROCESSAMENTO DE DADOS USANDO SCILAB: SOLUÇÃO DE SISTEMAS LINEARES E SUAS POSSIBILIDADES DE APLICAÇÃO NO CURSO DE ENGENHARIA CARTOGRÁFICA E AGRIMENSURA

ANA LÚCIA BEZERRA CANDEIAS<sup>1</sup>  
VINÍCIUS D'LUCAS BEZERRA E QUEIROZ<sup>2</sup>  
VANESSA SILVA DE MORAES<sup>3</sup>  
YASMIM RODRIGUES DE MENEZES<sup>4</sup>  
Universidade Federal de Pernambuco – UFPE  
Centro de Tecnologia e Geociências – CTG  
Departamento de Engenharia Cartográfica

[analucia@ufpe.br](mailto:analucia@ufpe.br)<sup>1</sup>  
[vinicius.dlucas@gmail.com](mailto:vinicius.dlucas@gmail.com)<sup>2</sup>  
[vanessamoraes\\_25@hotmail.com](mailto:vanessamoraes_25@hotmail.com)<sup>3</sup>  
[yasmimmenezes@gmail.com](mailto:yasmimmenezes@gmail.com)<sup>4</sup>

---

**RESUMO** – Os sistemas lineares está presente em diversas aplicações da Engenharia Cartográfica e Agrimensura tais como: ajustamento paramétrico, orientação interior e exterior, no georreferenciamento, na calibração de sistemas sensores, correção atmosférica das imagens entre outros. Este artigo apresenta a programação em SCILAB para solução de sistemas lineares utilizando os métodos de Gauss Seidel, Jacobi e Gauss Jordan. O SCILAB é um software livre e com programação similar ao MATLAB. Utiliza-se um exemplo bem condicionado e um outro mal condicionado e avalia-se os resultados. Esse estudo foi usado como recurso pedagógico na aprendizagem da programação do SCILAB.

**ABSTRACT** - Linear systems is present in many applications of Cartographic Engineering and Surveying such as parametric adjustment, interior and exterior orientation, georeferencing, the calibration sensor systems, atmospheric correction of images among others. This article presents SCILAB's programming for linear system solution using the methods of Gauss Seidel, Jacobi and Gauss Jordan. SCILAB's software is free and the programming is similar with MATLAB. Examples linear system and their conditioning are obtained. This study was used as a teaching resource in learning SCILAB programming.

---

### 1 INTRODUÇÃO

Os sistemas lineares diretos e/ou iterativos são aplicados em vários problemas nas Engenharias e áreas afins. Exemplos práticos de aplicação de sistemas lineares na Engenharia Cartográfica e Agrimensura são: ajustamento paramétrico, orientação interior e exterior, no georreferenciamento, na calibração de sistemas sensores, correção atmosférica das imagens entre outros.

Diante das restrições das universidades públicas em adquirir licenças de softwares proprietários tais como MATLAB, e a necessidade de os alunos desenvolverem programações aplicadas a Engenharia Cartográfica e Agrimensura, houve a necessidade, na disciplina de processamento de dados, do curso de Graduação em Engenharia Cartográfica da Universidade Federal de Pernambuco (UFPE), da utilização de software livre que tivessem as mesmas facilidades nos cálculos matriciais e na programação que o MATLAB. Escolheu então a plataforma SCILAB.

Este artigo tem o intuito apresentar a programação com SCILAB aplicado aos métodos de Jacobi, Gauss Seidel e Gauss Jordan. Analisa-se os resultados obtidos da programação para um exemplo bem condicionado e para um exemplo mal condicionado. Os usuários de MATLAB irão verificar grande semelhança da programação com SCILAB.

O SCILAB foi criado em 1989 por um grupo de pesquisadores da INRIA e da ENPC e está disponível como software livre desde 1994 pelo site <http://www.scilab.org>. Os Sistemas Operacionais que rodam SCILAB são: Linux, Windows, Solaris e Unix. O SCILAB utiliza interpretador de comandos e, portanto, o código gerado não é compilado. Os programadores que utilizam MATLAB facilmente podem converter seus programas para SCILAB. Essa linguagem possui a facilidade e simplicidade da linguagem estruturada. E como no MATLAB não existe necessidade de declaração prévia das variáveis.

---

As principais características do SCILAB são (LACERDA, 2011):

1. Distribuição gratuita, com código fonte disponível. Sua linguagem é simples e de fácil aprendizado;
2. Possui um sistema ajuda para o usuário, help;
3. Implementa diversas funções para manipulação de matrizes. As operações de concatenação, acesso e extração de elementos, transposição, adição e multiplicação de matrizes são facilmente realizadas;
4. Permite trabalhar com polinômios, funções de transferência, sistemas lineares e grafos;
5. Apresenta facilidades para a definição de funções;
6. Permite o acesso a rotinas escritas nas linguagens FORTRAN ou C;
7. Pode ser acessado por programas de computação simbólica como o Maple, que é um software comercial, ou o MuPAD, que é livre para uso em instituições de ensino/pesquisa;
8. Suporta o desenvolvimento de conjuntos de funções voltadas para aplicações específicas, os chamados toolboxes.

### 1.1 Solução de sistemas lineares

A solução de sistemas lineares pode ser vista Santos e Silva (2010) e em Pal (2009). Nesse artigo utiliza-se uma adaptação de Pal (2009) utilizando leitura e escrita em arquivo de dados e também apresenta-se uma representação gráfica.

Os métodos de Jacobi e Gauss Seidel estão apresentados e linguagem algorítmica nas Figuras 1 e 2 e o método de Gauss Jordan pode ser visto em Santos e Silva (2010). Os exemplos desenvolvidos fizeram parte da disciplina de Processamento de Dados do Curso de Engenharia Cartográfica e Agrimensura da Universidade Federal de Pernambuco do 1º semestre de 2016.

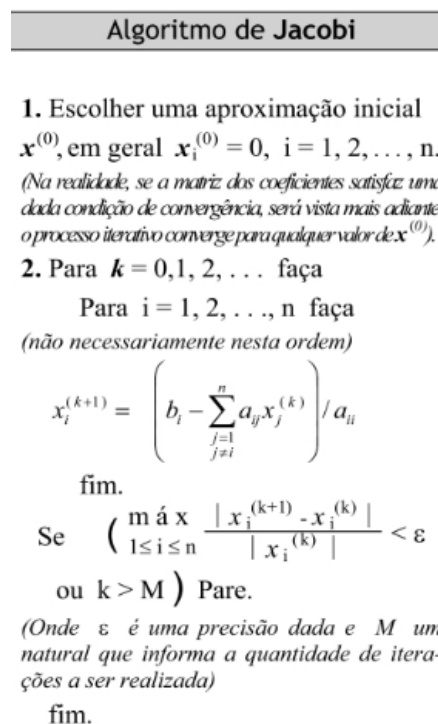


Figura 1 – Jacobi. Fonte: Santos e Silva (2010).

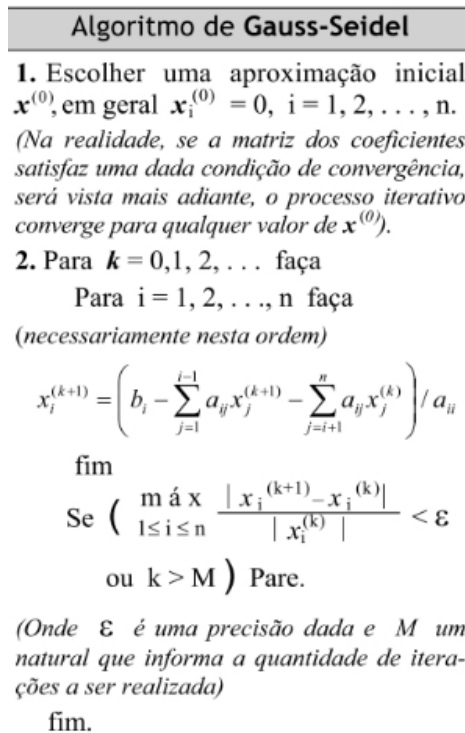


Figura 2 – Gauss Seidel. Fonte: Santos e Silva (2010).

## 2. METODOLOGIA DO TRABALHO

Utilizou-se os métodos de eliminação de Gauss, Jacobi e Gauss Jordan empregando SCILAB. A metodologia do trabalho consistiu em:

- Adaptar a programação desenvolvida em Pal (2009) para leitura de arquivo de dados
- Gerar os resultados de programação
- Aplicar exemplo de solução de sistema linear avaliar seu condicionamento

## 3. RESULTADOS E DISCUSSÃO

Os programas são apresentados nas seções 3.1, 3.2 e 3.3 e mostram os métodos de Gauss Seidel, Jacobi e Gauss Jordan usando SCILAB. Para exemplificação utiliza-se dois exemplos (Equação 1 e Equação 2) vistos em Gemael (2004) e que possuem respectivamente determinante da matriz A igual a -3 (bem condicionado) e -0,001 (mal condicionado).

Supõe-se o sistema linear  $AX = L$ . Onde  $A_{n \times n}$  é uma matriz,  $L_{n \times 1}$  é um vetor ambos conhecidos e  $X_{n \times 1}$  é o vetor a ser determinado. A solução será dada por  $X = A^{-1} L$ . Nesse artigo  $A^{-1}$  é obtido pelos programas das seções 3.1, 3.2 e 3.3. A seguir apresenta-se os exemplos usados.

### Exemplo 1:

$$\begin{bmatrix} 1 & 1 \\ 1 & -2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 10 \\ -5 \end{bmatrix} \quad \text{com } \det(A) = -3 \quad (1)$$

### Exemplo 2:

$$\begin{bmatrix} 1 & 1 \\ 1,001 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 10 \\ 10,1 \end{bmatrix} \quad \text{com } \det(A) = -0,001 \quad (2)$$

Os arquivos em .txt a serem lidos nos programas apresentados nas seções 3.1, 3.2 e 3.3 utilizam o formato: **A|L**. Para os exemplos:

<u>Arq1.txt</u>			<u>Arq2.txt</u>		
1	1	10	1	1	10
1	-2	-5	1,001	1	10,1

A programação permite a entrada de qualquer tamanho de matriz A quadrada

### 3.1 Programação do Método de Gauss-Seidel usando SCILAB

```
//~~~~~
//      Método de Gauss-Seidel - adaptado de Pal (2009)
//~~~~~
clc ;
clear ;
close ;

disp('Método de Gauss-Seidel')

//~~~~~
//      Solicitação ao usuário um arquivo .txt para ser guardado na variável "arq"
//~~~~~

arq=uigetfile("*.txt",pwd(),'selecione o arquivo com a matriz de entrada do sistema linear')

//~~~~~
//      Abertura do arquivo
//~~~~~

G=mopen(arq,'r')

//~~~~~
//      Leitura do arquivo e atribuindo-o a matriz "A".
//~~~~~

A=fscanfMat(arq)
[nl,nc]=size(A)
B=A(:,nc);
o=input('Digite o número de iterações: ');
format('v',7);
for i=0:o+1
    X(i+1,1)=i ;
end
for i=2:nc
    X(1,i)=0;
end
for r=1:o+1
    for i=1:nl
        k1=0;
        for j=1:i-1
            k1=k1-A(i,j)*X(r+1,j+1) ;
        end
        k2=0;
        for j= i +1:nl
```

```

        k2=k2-A(i,j)*X(r,j+1) ;
    end
    X(r+1,i+1)=(k1+k2+B(i,1))/A(i, i) ;
end
end
printf ( 'r\t ' )
for p=1:nc-1
    printf(' x(%i) \t',p) ;
end
fclose(G)
//~~~~~
//  Armazenamento do resultado em arquivo no diretório corrente
//~~~~~

G=mopen('saida_Gauss_Seidel.txt','w')
printf ( '\n ----- ' )
disp ( X )
printf( '\n\n Com %g iterações, a solução é: \n ',o)
mfprintf(G,'A solução do sistema com %i iterações: \n', o)
for q=1:nc-1
    printf('x(%i)=%.4f\t',q,X(o+1,q+1));
    mfprintf(G,'x(%i)=%.4f\t',q,X(o+1,q+1));
end
fclose(G)
plot(X(:,2),'r*-' )
xlabel('Numero de iteracoes')
ylabel('Resultado da variavel x1')
title('Metodo de Gauss Seidel')
legend ('x1');
plot(X(:,2),'r*-' )
figure
plot(X(:,3),'b*-' )
xlabel('Numero de iteracoes')
ylabel('Resultado da variavel x2')
title('Metodo de Gauss Seidel')
legend ('x2');

```

### 3.2 Programação do Método de Jacobi usando SCILAB

```

//~~~~~
//  Método de Jacobi - adaptado de Pal (2009)
//~~~~~

clc ; clear ; close ;
disp('Método de Jacobi')

//~~~~~
//  Solicitação ao usuário um arquivo .txt para ser guardado na variável "arq"
//~~~~~

arq=uigetfile("*.txt",pwd(),'selecione o arquivo com a matriz para resolução de sistema linear')

//~~~~~
//  Abertura do arquivo
//~~~~~

G=mopen(arq,'r')

```

```

//~~~~~
//      Leitura do arquivo e atribuindo-o a matriz "A".
//~~~~~

A=fscanfMat(arq)
[nl,nc]=size(A)
B=A(:,nc);
format('v',7);
o=input('Digite o número de iterações: ');

for i =0:o+1
    X(i+1,1)= i ;
end
for i=2:nc
    X(1,i)=0;
end
for r=1:o+1
    for i=1:nl
        k=0;
        for j=1:nl
            if( i~=j )
                k=k-A(i,j)*X(r,j +1) ;
            end
        end
        X(r+1,i+1)=(k+B(i,1))/A(i , i) ;
    end
end
printf ( ' r \t ' )
for p=1:nc-1
    printf( ' x(%i) \t',p) ;
end
printf ( '\n ----- ' )
disp ( X )
printf( '\n\n Depois de %g iterações a solução é:\n',o)
mclose(G)

//~~~~~
//      Armazenamento do resultado em arquivo no diretório corrente
//~~~~~

G=mopen('saida_Jacobi.txt','w')
mfprintf(G,'A solução do sistema com %i iterações é:\n', o)

format('v',7);
for q=1:nc-1
    printf('x(%i)=%.4f\t',q,X(o+2,q+1));
    mfprintf(G,'x(%i)=%.4f\t',q,X(o+2,q+1));
end
mclose(G)
plot(X(:,2),'r*-' )
xlabel('Numero de iteracoes')
ylabel('Resultado da variavel x1')
title('Metodo de jacobi')
legend ('x1');
plot(X(:,2),'r*-' )
figure
plot(X(:,3),'b*-' )
xlabel('Numero de iteracoes')

```

```
ylabel('Resultado da variavel x2')
title('Metodo de jacobi')
legend('x2');
```

### 3.3 Programação do Método de Gauss-Jordan usando SCILAB

```
//~~~~~
//      Método de Gauss–Jordan - adaptado de Pal (2009)
//~~~~~

clc ; clear ; close ;
disp("Método de Gauss–Jordan")
disp(' ')

//~~~~~
// Solicitação ao usuário um arquivo .txt para ser guardado na variável "arq"
//~~~~~

arq=uigetfile("*.txt",pwd(),'Selecione o arquivo com matriz para resolução do sistema ')

//~~~~~
//      Abertura do arquivo
//~~~~~

G=mopen(arq,'r')

//~~~~~
//      Leitura do arquivo e atribuindo-o a matriz "A".
//~~~~~

A=fscanfMat(arq)

[nl,nc]=size(A)

format('v',7);
for i=1:nl
    j=i
    while ( A ( i , i ) ==0 & j <=nl)
        for k=1:nc
            B(1 , k)= A ( j +1 , k)
            A(j +1 , k) = A ( i , k)
            A(i , k)= B ( 1 , k)
        end
        disp ( A )
        j = j+1
    end
    disp ( A )
    for k =nc:-1:i
        A(i,k)=A(i,k)/A(i,i)
    end
    disp(A)
    for k=1:nl
        if(k ~=i) then
            l=A(k,i)/A(i,i)
            for m=i:nc
                A(k,m)=A(k,m)-l*A(i,m)
            end
        end
    end
end
```

```

end

end
disp ( A )
end
printf('A solução do sistema é:\n')
for i=1:nl
    printf ( '\n x(%i)=%g\n ',i , A(i,nc) )
end
mclose(G)
G=mopen('saida_Gauss_Jordan.txt','w')

mfprintf(G,' A solução do sistema é:\n')

for i=1:nl
    mfprintf ( G,'\n x(%i)=%g\n ',i , A(i,nc) )
end
mclose(G)

```

### 3.4 Resultados obtidos com os programas

Para o exemplo 1 (Equação 1) tem-se a Tabela 1 e as Figura 4 e 5. Observa-se que os resultados convergem para os valores de x e y iguais a cinco (5). O determinante é igual a -3 e, portanto, não há mal condicionamento.

Tabela 1 – Resultados obtidos com o exemplo 1

Valor	x	y
Gauss Jordan - Exemplo 1 SCILAB	5.0000	5.0000
Jacobi - Exemplo 1 (SCILAB) com 30 iterações	4.9998	5.0001
Gauss-Jordan - Exemplo 1 (SCILAB) com 30 iterações	5.0000	5.0000

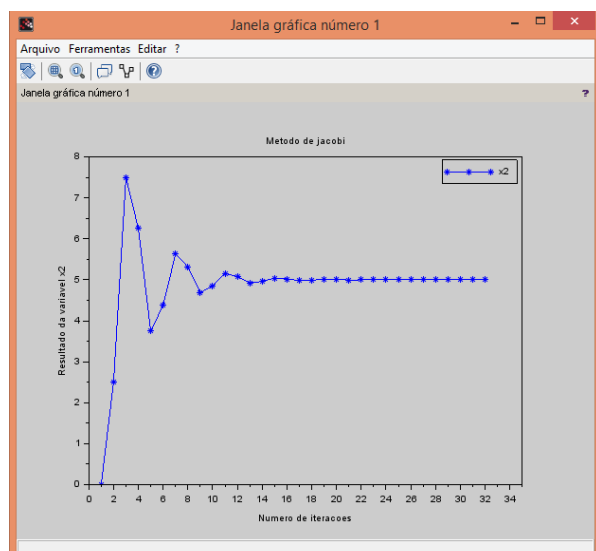
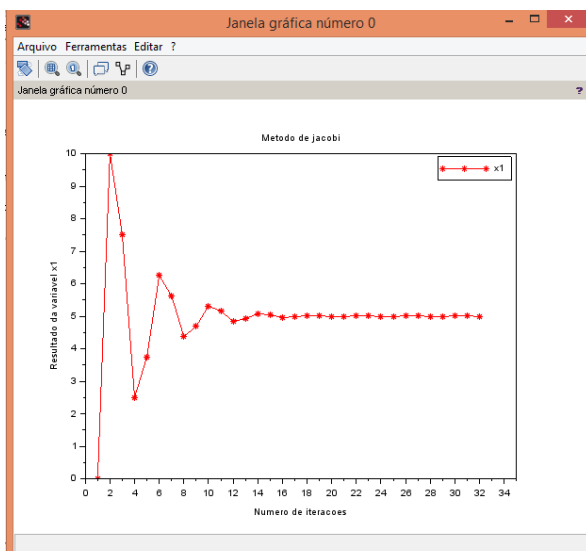


Figura 4 – Resultado de x e y para o método de Jacobi - Exemplo 1



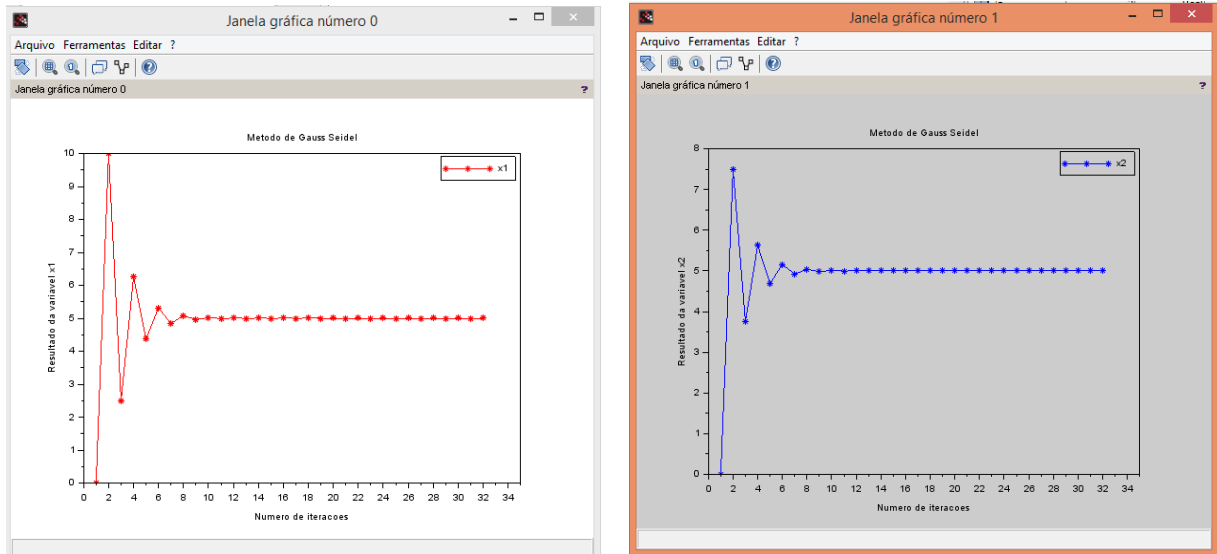


Figura 5 – Resultado de x e y para o método de Gauss Seidel.

Para o exemplo 2 (Equação 2) tem-se a Tabela 2 e as Figuras 6 e 7. Observa-se que os resultados de x e y não convergem e cada método gera um resultado distinto. Isso é devido ao determinante estar muito próximo a zero. O sistema é mal condicionado.

Tabela 1 – Resultados obtidos com o exemplo 2

Valor	x	y
Gauss Jordan Exemplo 1 SCILAB	100	-90
Jacobi - Exemplo 2 (SCILAB) com 30 iterações	8.6405	11.6121
Gauss-Jordan - Exemplo 2 (SCILAB) com 30 iterações	7.3531	2.7395

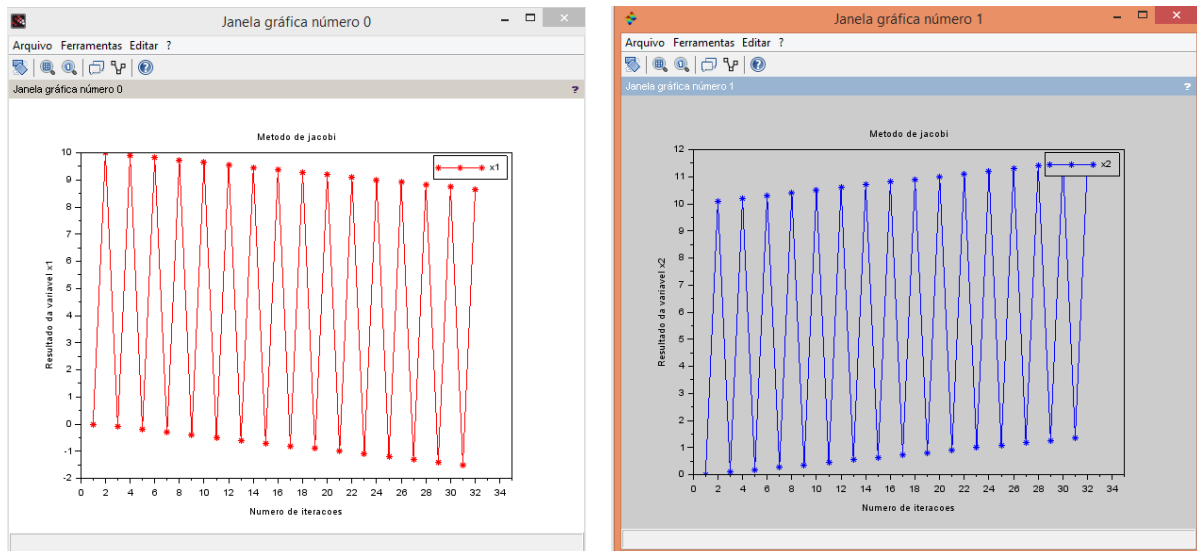


Figura 6 – Resultado de x e y para o método de Jacobi que não convergiu em 30 iterações - Exemplo 2.

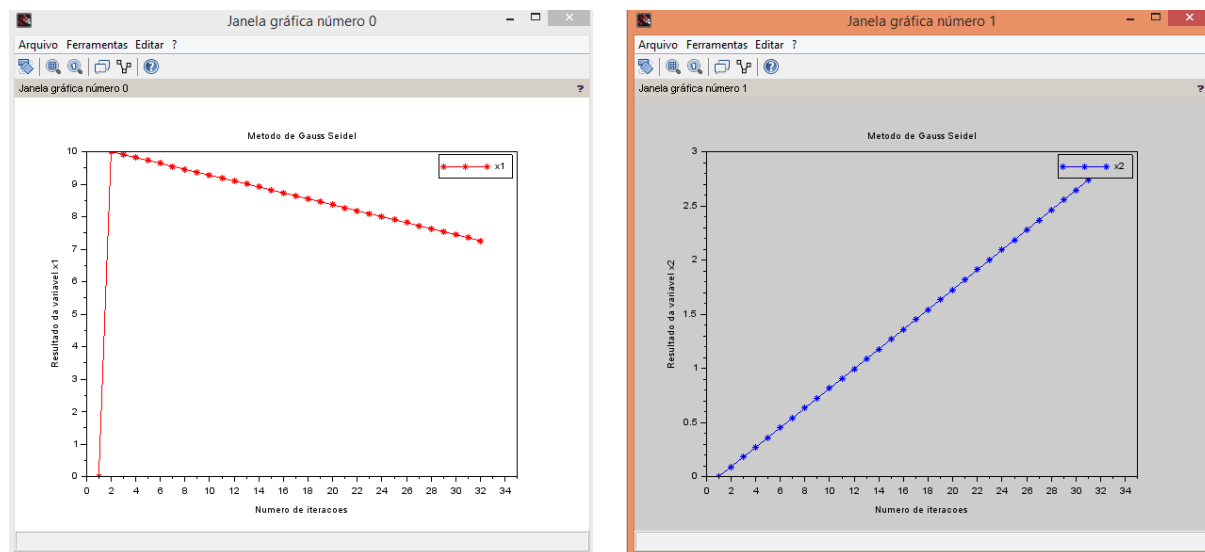


Figura 7 – Resultado de x e y para o método de Gauss Seidel que não convergiu em 30 iterações - Exemplo 2.

#### 4. CONCLUSÕES

A solução de sistemas lineares está intrinsecamente ligada a diversas aplicações da Engenharia Cartográfica e Agrimensura tais como: ajustamento paramétrico, orientação interior e exterior, no georreferenciamento, na calibração de sistemas sensores, correção atmosférica das imagens entre outros. Este trabalho apresentou solução de sistema linear com a plataforma SCILAB que é um software livre e com programação semelhante ao MATLAB. Os métodos de Gauss Seidel, Jacobi e Gauss Jordan foram apresentados e avaliou-se seus condicionamentos.

Este trabalho serviu para desenvolver nos alunos da disciplina de processamento de dados do curso de Engenharia Cartográfica e Agrimensura, a programação com software livre e mostrar sua eficiência na programação com sistemas lineares.

#### 5. AGRADECIMENTOS

Agradecimentos ao Laboratório de Informática (LAINFO) do departamento de Engenharia Cartográfica da UFPE.

#### 5. REFERENCIAS BIBLIOGRAFICAS

SANTOS, J. D.; SILVA, Z. C. **Métodos Numéricos**. Editora Universitária da UFPE. 2010. Disponível em: <<http://www.loja.ufpe.com.br/portal/spring/livro/detalhe/67>>. Acesso em 11 de junho de 2016

PAL, S. **Numerical Methods: Principles, Analysis, And Algorithms**. Oxford University Press. 2009. Disponível em: <[http://www.scilab.in/textbook\\_companion/generate\\_book/1332](http://www.scilab.in/textbook_companion/generate_book/1332)>. Acesso em 11 de junho de 2016

GEMAEL, C. **Introdução ao ajustamento de observações: aplicações geodésicas**. Ed. UFPR. 2004. 319p.