Fundação Getulio Vargas

Escola de Matemática Aplicada (EMAp)

Nome: Lucas Machado Moschen

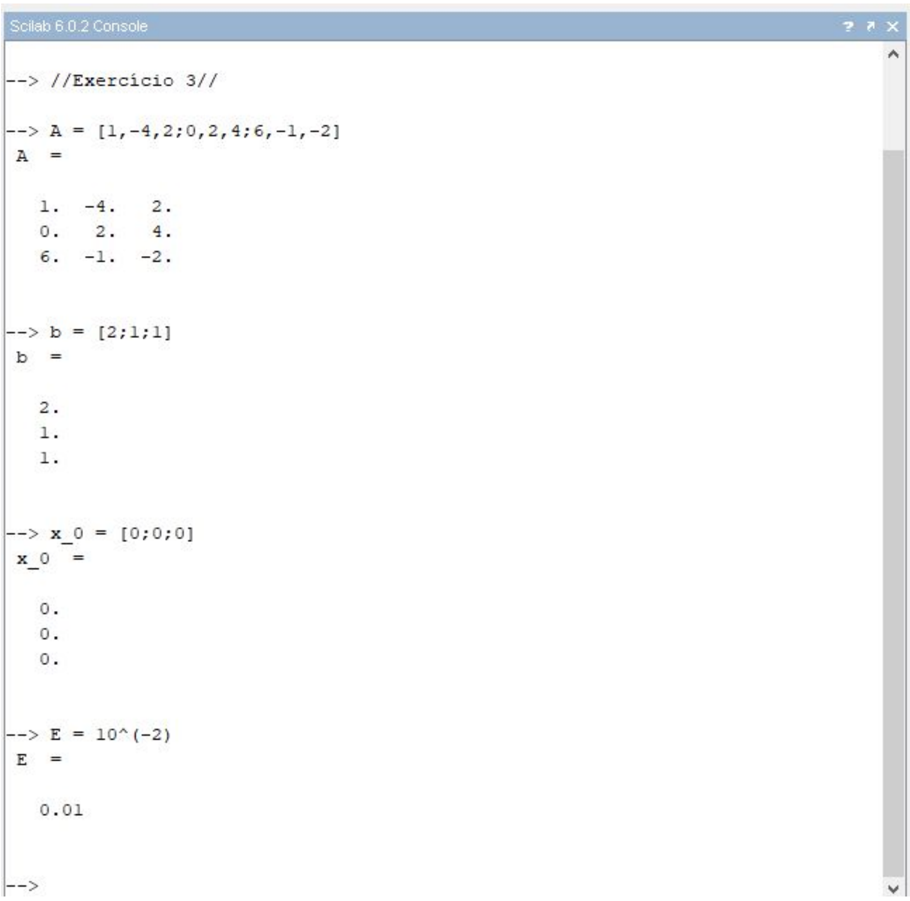
Data: 02 de abril de 2019

Professor Antônio Branco

Aula Prática 2

Exercícios 1 e 2: Realizam-se as atividades de código sugeridas. Foi utilizado o scilab para o fim de programar. Os arquivos seguem anexados com esse documento.

Exercício 3: Ambas as funções que contém os métodos foram testadas. Todavia, ambas apresentam problema de convergência para o resultado. Notamos que a matiz não é estritamente dominante. Porém, após a alteração das linhas, para a matriz ter diagonal estritamente dominante, a convergência é alcançada. A primeira parte é basicamente os valores a serem testados.



```
Scilab 6.0.2 Console

--> //Exercício 3//

--> A = [1,-4,2;0,2,4;6,-1,-2]
 A  =

   1.  -4.   2.
   0.   2.   4.
   6.  -1.  -2.


--> b = [2;1;1]
 b  =

   2.
   1.
   1.


--> x_0 = [0;0;0]
 x_0  =

   0.
   0.
   0.


--> E = 10^(-2)
 E  =

   0.01


-->
```

Note que nessa imagem, após a execução do método de Jacobi, o número de iterações k = 51 > M = 50 e o resíduo ultrapassou a casa dos 10^{26}. Na imagem logo abaixo, o processo de não aproximação do resultado é similar, porém com um erro ainda maior.

```
Scilab 6.0.2 Console

-->
-->
-->
--> exec jacobi_solver.sce



-->
-->
--> [xk, dif, k, residuo] = Jacobi_solver(A,b,x_0,E,M,1)

 k>M: o número de iterações ultrapassou o máximo
 residuo  =

   3.421D+26

 k  =

   51.

 dif  =

   7.592D+25

 xk  =

  -4.608D+25
  -2.234D+25
  -4.889D+24


-->
```

```
Scilab 6.0.2 Console

   7.127D+34

 xk  =

   2.212D+34
   6.144D+34
   3.565D+34


--> [xk, dif, k, residuo] = GaussSeidel_solver(A,b,x_0,E,M,2)

 k>M: o número de iterações ultrapassou o máximo
 residuo  =

   3.061D+35

 k  =

   51.

 dif  =

   1.031D+35

 xk  =

   2.212D+34
   6.144D+34
   3.565D+34


-->
```

Apliquei uma matriz de P à matriz A e ao vetor b, a fim de a matriz A ser de diagonal estritamente dominante. Essa é uma condição suficiente para a convergência dos métodos, o que é facilmente percebido pelos valores de resíduo e o erro, menor que a tolerância.

```
Scilab 6.0.2 Console

--> P
 P  =

   0.   0.   1.
   1.   0.   0.
   0.   1.   0.


--> A = P*A
 A  =

   6.  -1.  -2.
   1.  -4.   2.
   0.   2.   4.


--> b = P*b
 b  =

   1.
   2.
   1.
```

```
Scilab 6.0.2 Console

--> b = P*b
 b  =

   1.
   2.
   1.


--> [xk, dif, k, residuo] = Jacobi_solver(A,b,x_0,E,M,1)

 ||x_k - x_k-1|| < E: a norma da diferença entre os vetores é menor do que
 a tolerância

 residuo  =

   0.0151307

 k  =

   8.

 dif  =

   0.0072097

 xk  =

   0.2512418
  -0.2482157
   0.3750362
```

```
Scilab 6.0.2 Console

--> [xk, dif, k, residuo] = GaussSeidel_solver(A,b,x_0,E,M,1)

 ||x_k - x_k-1|| < E: a norma da diferençça entre os vetores é menor do que
  a tolerância

 residuo  =

   0.004069

 k  =

   5.

 dif  =

   0.0061035

 xk  =

   0.25
  -0.2508138
   0.3754069
```

Exercício 4:

a) Vê-se pela imagem que após 25 iterações, o método de Jacobi não converge para esse sistema. Observe que a matriz A não tem diagonal estritamente dominante.

Scilab 6.0.2 Console

```
--> //Exercicio 4.a

--> A
 A  =

   2.  -1.   1.
   2.   2.   2.
  -1.  -1.   2.


--> b = [-1;4;-5]
 b  =

  -1.
   4.
  -5.


--> M = 25
 M  =

   25.
```

```
--> [xk, dif, k, residuo] = Jacobi_solver(A,b,[0;0;0],E,M,1)

 k>M: o número de iterações ultrapassou o máximo


 residuo  =

   196.45086

 k  =

   26.

 dif  =

   87.311491

 xk  =

   11.913936
   45.655746
  -11.913936
```

b) Para o mesmo sistema de equações, o método de Gauss-Seidel converge e com menos de 25 iterações, com erro menor do que a tolerância que exigimos.

Scilab 6.0.2 Console

```
--> //Exercicio 4.b//

--> A
 A  =

   2.  -1.   1.
   2.   2.   2.
  -1.  -1.   2.


--> b
 b  =

  -1.
   4.
  -5.
```

```
--> [xk, dif, k, residuo] = GaussSeidel_solver(A,b,[0;0;0],10^(-5),50,%inf)

 ||x_k - x_k-1|| < E: a norma da diferençça entre os vetores é menor do que
  a tolerância

 residuo  =

   0.0000069

 k  =

   23.

 dif  =

   0.0000073

 xk  =

   1.0000023
   1.9999975
  -1.0000001
```

5) **a)** Os resultados são encontrados na imagem abaixo.

```
Scilab 6.0.2 Console

--> //Exercicio 5.a//

--> A
 A  =

   1.     0.    -1.
  -0.5    1.    -0.25
   1.    -0.5    1.


--> b
 b  =

   0.2
  -1.425
   2.
```

```
--> [xk, dif, k, residuo] = GaussSeidel_solver(A,b,[0;0;0],10^(-2),300)

 ||x_k - x_k-1|| < E: a norma da diferençça entre os vetores é menor do que
  a tolerância

 residuo  =

   0.0041656

 k  =

   13.

 dif  =

   0.0090364

 xk  =

   0.8975131
  -0.8018652
   0.7015543
```

b) Quando o sistema é alterado, A(1,2) = -2, uma mudança muito sutil aos olhos de um leigo, os resultados são completamente diferentes. Em 300 iterações, o que antes foram necessárias 19 iterações, aqui foram necessárias 301 para ver que a solução não convergia.

Scilab 6.0.2 Console

```
--> A(1,3) = -2
 A  =

   1.     0.    -2.
  -0.5    1.    -0.25
   1.    -0.5    1.


--> [xk, dif, k, residuo] = GaussSeidel_solver(A,b,[0;0;0],10^(-2),300)

 k>M: o número de iterações ultrapassou o máximo


 residuo  =

   7.098D+41

 k  =

   301.

 dif  =

   6.992D+41

 xk  =

  -2.966D+41
  -1.854D+41
   2.039D+41
```

6)

$$
\begin{bmatrix}
4 & -1 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 4 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 4 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -1 & 4 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & 4 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & -1 & 0 & 4 & 0 & -1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 4 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 4 & 0 & 0 & 0 & -1 \\
0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 4 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 4 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 4 & -1 \\
0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & -1 & 4
\end{bmatrix}
\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \\ w_5 \\ w_6 \\ w_7 \\ w_8 \\ w_9 \\ w_{10} \\ w_{11} \\ w_{12} \end{bmatrix}
=
\begin{bmatrix} 220 \\ 110 \\ 110 \\ 220 \\ 110 \\ 110 \\ 110 \\ 110 \\ 220 \\ 110 \\ 110 \\ 220 \end{bmatrix}.
$$

a. Esta matriz é estritamente diagonal dominante?

b. Resolva o sistema linear usando o método de Jacobi com $\mathbf{x}^{(0)} = \mathbf{0}$ e $TOL = 10^{-2}$.

c. Repita a parte (b) usando o método de Gauss-Seidel.

a) A matriz é estritamente diagonal dominante, pois em todas as linhas:

$$A(i,i) = 4 > |-1| + |-1| = 2$$

b) A solução para o sistema utilizando o método de Jacobi e **c)** o método de Gauss Seidel, com tolerância solicitada encontra-se nas imagens abaixo.

Scilab 6.0.2 Console

```
 A  =

   4.  -1.   0.   0.  -1.   0.   0.   0.   0.   0.   0.   0.
  -1.   4.  -1.   0.   0.   0.   0.   0.   0.   0.   0.   0.
   0.  -1.   4.  -1.   0.   0.   0.   0.   0.   0.   0.   0.
   0.   0.  -1.   4.   0.  -1.   0.   0.   0.   0.   0.   0.
  -1.   0.   0.   0.   4.   0.  -1.   0.   0.   0.   0.   0.
   0.   0.   0.  -1.   0.   4.   0.  -1.   0.   0.   0.   0.
   0.   0.   0.   0.  -1.   0.   4.   0.  -1.   0.   0.   0.
   0.   0.   0.   0.   0.  -1.   0.   4.   0.   0.   0.  -1.
   0.   0.   0.   0.   0.   0.  -1.   0.   4.  -1.   0.   0.
   0.   0.   0.   0.   0.   0.   0.   0.  -1.   4.  -1.   0.
   0.   0.   0.   0.   0.   0.   0.   0.   0.  -1.   4.  -1.
   0.   0.   0.   0.   0.   0.   0.  -1.   0.   0.  -1.   4.


--> b
 b  =

   220.
   110.
   110.
   220.
   110.
   110.
   110.
   110.
   220.
   110.
   110.
   220.


-->
```

Scilab 6.0.2 Console

```
--> [xk, dif, k, residuo] = Jacobi_solver(A,b,zeros(12,1),10^(-2),300)

 ||x_k - x_k-1|| < E: a norma da diferença entre os vetores é menor do que
 a tolerância

 residuo  =

   0.015505

 k  =

   15.

 dif  =

   0.0077525

 xk  =

   87.997762
   65.997762
   65.997762
   87.997762
   65.997762
   65.997762
   65.997762
   65.997762
   87.997762
   65.997762
   65.997762
   87.997762


-->
```

Scilab 6.0.2 Console

```
--> [xk, dif, k, residuo] = GaussSeidel_solver(A,b,zeros(12,1),10^(-2),300)

 ||x_k - x_k-1|| < E: a norma da diferençça entre os vetores é menor do que
  a tolerância

 residuo  =

   0.0042672

 k  =

   10.

 dif  =

   0.0048536

 xk  =

   87.99869
   65.999312
   65.999631
   87.999803
   65.999312
   65.999897
   65.999631
   65.999947
   87.999803
   65.999897
   65.999947
   87.999974


-->
```