Fundação Getulio Vargas

Escola de Matemática Aplicada (EMAp)

Nome: Lucas Machado Moschen

Data: 21 de março de 2019

Professor Antônio Branco

Aula Prática 1

Exercício 1: Inicia-se com dois testes de sistemas lineares, em forma de matrizes. Para esse exercício e para os próximos, testa-se a validade do resultado com a inversa.

Scilab 6.0.2 Console

```
--> // Exercicio 1

--> A = [3,2;1,2]
 A  =

   3.   2.
   1.   2.


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

   5.
  -1.


--> [x,C] = Gaussian_Elimination_1(A,b)
 C  =

   3.          2.
   0.3333333   1.3333333

 x  =

   3.
  -2.


--> Resposta_certa = inv(A)*b
 Resposta_certa  =

   3.
  -2.
```

```
Scilab 6.0.2 Console
--> A = [-1,4,1,0;-1,0,1,5;1,0,1,4;-1,1,0,2]
 A  =

  -1.   4.   1.   0.
  -1.   0.   1.   5.
   1.   0.   1.   4.
  -1.   1.   0.   2.


--> b = [4;0.42;0.6;10]
 b  =

   4.
   0.42
   0.6
   10.


--> [x] = Gaussian_Elimination_1(A,b)
 x  =

   1.7618182
   5.0745455
  -14.536364
   3.3436364


--> Resposta_certa = inv(A)*b
 Resposta_certa  =

   1.7618182
   5.0745455
  -14.536364
   3.3436364
```

Exercício 2: Para a primeira função, parte-se da suposição de que os pivôs, durante o escalonamento para ter a decomposição LU, são diferentes de zero. Isso não ocorre nesse exemplo e, portanto, a função não funciona corretamente e retorna como resultado “Not a number”, pois há uma divisão por zero.

```
Scilab 6.0.2 Console
--> // Exercicio 2

--> A1=[1 -2 5 0; 2 -4 1 3; -1 1 0 2; 0 3 3 1]
 A1  =

   1.  -2.   5.   0.
   2.  -4.   1.   3.
  -1.   1.   0.   2.
   0.   3.   3.   1.


--> b1=[1;0;0;0]
 b1  =

   1.
   0.
   0.
   0.


--> [x,C] = Gaussian_Elimination_1(A1,b1)
 C  =

   1.  -2.    5.    0.
   2.   0.   -9.    3.
  -1.  -Inf  -Inf   Inf
   0.   Inf   Nan   Nan

 x  =

   Nan
   Nan
   Nan
   Nan
```

Exercício 3: Trocam-se linhas no momento que um dos pivôs é zero. Isso já ajuda a corrigir o problema visto anteriormente. Entretanto, no segundo caso, é observado que uma pequena alteração na matriz, para um valor muito próximo de zero, o resultado é alterado drasticamente do correto.

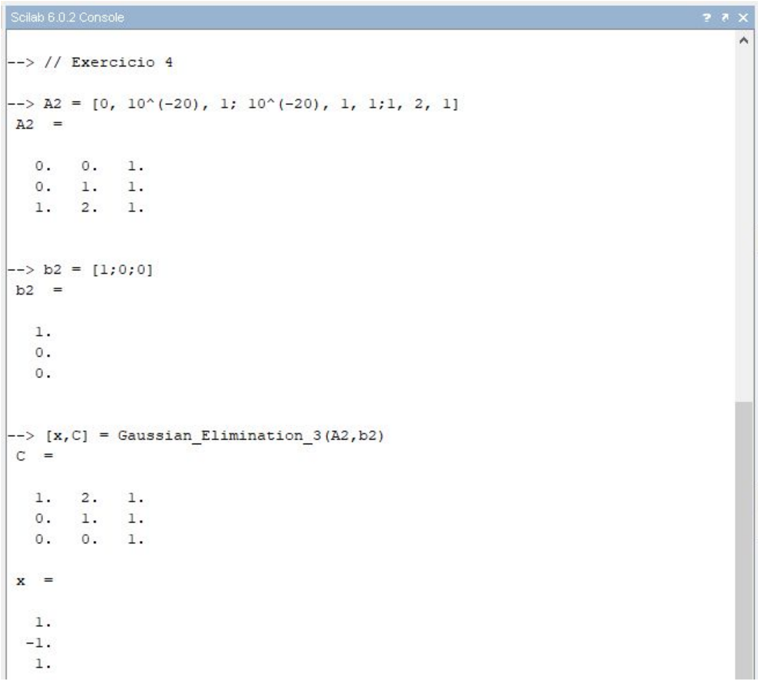
```
Scilab 6.0.2 Console
--> // Exercicio 3

--> A1=[1 -2 5 0; 2 -4 1 3; -1 1 0 2; 0 3 3 1]
 A1  =

   1.  -2.   5.   0.
   2.  -4.   1.   3.
  -1.   1.   0.   2.
   0.   3.   3.   1.


--> b1=[1;0;0;0]
 b1  =

   1.
   0.
   0.
   0.


--> [x,C] = Gaussian_Elimination_2(A1,b1)
 C  =

   1.  -2.   5.   0.
  -1.  -1.   5.   2.
   2.   0.  -9.   3.
   0.  -3.  -2.   13.

 x  =

  -0.3247863
  -0.1709402
   0.1965812
  -0.0769231
```

```
--> Resposta_certa = inv(A1)*b1
 Resposta_certa  =

  -0.3247863
  -0.1709402
   0.1965812
  -0.0769231
```

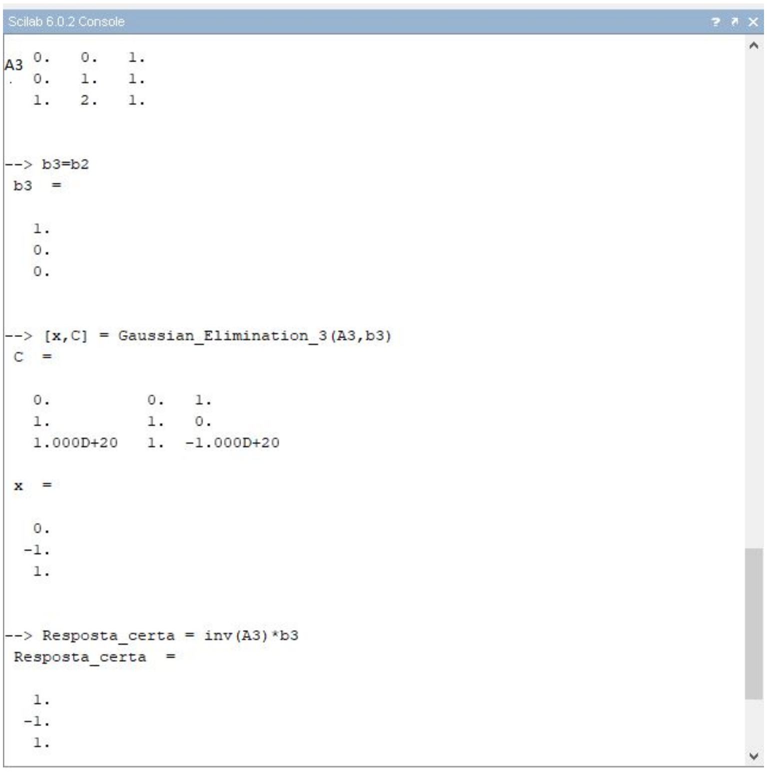
```
Scilab 6.0.2 Console

--> A2 = [0, 10^(-20), 1; 10^(-20), 1, 1;1, 2, 1]
 A2  =

   0.   0.   1.
   0.   1.   1.
   1.   2.   1.


--> b2 = [1;0;0]
 b2  =

   1.
   0.
   0.


--> [x,C] = Gaussian_Elimination_2(A2,b2)
 C  =

   0.          1.          1.
   0.          0.          1.
   1.000D+20  -1.000D+40   1.000D+40

 x  =

  -1.000D+20
   0.
   1.
```

```
--> Resposta_certa = inv(A2)*b2
 Resposta_certa  =

   1.
  -1.
   1.
```

Exercício 4: Nessa questão, tenta-se corrigir o que ocorreu de errado na questão anterior. Quando encontra-se um valor igual a 0 no pivô, toma-se o maior valor na coluna correspondente, abaixo do pivô em questão. O objetivo é aumentar a estabilidade do cálculo. O resultado é bem interessante para a matriz anterior, porém ainda pode ser melhorado, já que basta trocar a ordem das linhas na matriz inicial que o pequeno valor será o pivô.



```
Scilab 6.0.2 Console

--> // Exercicio 4

--> A2 = [0, 10^(-20), 1; 10^(-20), 1, 1;1, 2, 1]
 A2  =

   0.   0.   1.
   0.   1.   1.
   1.   2.   1.


--> b2 = [1;0;0]
 b2  =

   1.
   0.
   0.


--> [x,C] = Gaussian_Elimination_3(A2,b2)
 C  =

   1.   2.   1.
   0.   1.   1.
   0.   0.   1.

 x  =

   1.
  -1.
   1.
```



```
Scilab 6.0.2 Console
A3 0.   0.   1.
   0.   1.   1.
   1.   2.   1.


--> b3=b2
 b3  =

   1.
   0.
   0.


--> [x,C] = Gaussian_Elimination_3(A3,b3)
 C  =

   0.           0.   1.
   1.           1.   0.
   1.000D+20    1.  -1.000D+20

 x  =

   0.
  -1.
   1.


--> Resposta_certa = inv(A3)*b3
 Resposta_certa  =

   1.
  -1.
   1.
```

Exercício 5: Por fim, tomamos como pivô sempre o maior valor em módulo. Desde Gaussian_Elimination_2, retorna-se a matriz P de permutação de linhas, que será utilizada no exercício posterior.

```
Scilab 6.0.2 Console

--> // Exercicio 5

--> A3
 A3  =

   0.   0.   1.
   0.   1.   1.
   1.   2.   1.


--> b3
 b3  =

   1.
   0.
   0.


--> [x,C] = Gaussian_Elimination_4(A3,b3)
 C  =

   1.   2.   1.
   0.   1.   1.
   0.   0.   1.

 x  =

   1.
  -1.
   1.
```

Exercício 6: Utilizou-se a função Gaussian_Elimination_4 para fazer a decomposição LU da matriz PA. P é a matriz de permutações, que pode ser a identidade de mesmo tamanho de A, caso a troca de linhas não seja necessária. Obtiveram-se os resultados e conferiu-se com a matriz inversa. Na função Resolve_com_LU, a ordem dos parâmetros é:

C: decomposição LU da matriz PA;

B: matriz em que cada coluna representa a combinação linear das colunas de A, correspondente a coluna da matriz X;

P: Matriz de permutação da decomposição LU. Se ela não for passada como parâmetro, ela é dita como a matriz identidade de tamanho nxn.

O resultado é atribuído a uma matriz X, onde cada coluna é a solução do sistema corresponde à coluna em B.

```
--> A1
 A1  =

   1.  -2.   5.   0.
   2.  -4.   1.   3.
  -1.   1.   0.   2.
   0.   3.   3.   1.


--> [x,C,P] = Gaussian_Elimination_4(A1,b1)
 P  =

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

 C  =

   2.   -4.          1.          3.
   0.    3.          3.          1.
   0.5   0.          4.5        -1.5
  -0.5  -0.3333333   0.3333333   4.3333333
```

```
   2.   4.  -1.   5.   3.
   0.   1.   0.   3.  -3.
   2.   2.  -1.   1.   0.
   0.   1.   1.   5.  -4.


--> X = Resolve_com_LU(C,P,B1)
 X  =

  -2.034188   -1.9316239   1.4529915   0.8119658  -3.6666667
  -0.6495726  -0.7008547   0.6068376   0.4273504  -1.6666667
   0.5470085   0.9059829  -0.2478632   1.008547    0.6666667
   0.3076923   0.3846154  -0.0769231   0.6923077  -1.


--> Resposta_certa = inv(A1)*B1
 Resposta_certa  =

  -2.034188   -1.9316239   1.4529915   0.8119658  -3.6666667
  -0.6495726  -0.7008547   0.6068376   0.4273504  -1.6666667
   0.5470085   0.9059829  -0.2478632   1.008547    0.6666667
   0.3076923   0.3846154  -0.0769231   0.6923077  -1.
```

Scilab 6.0.2 Console

```
--> A2
 A2  =

   0.   0.   1.
   0.   1.   1.
   1.   2.   1.


--> B2
 B2  =

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


--> [x,C,P] = Gaussian_Elimination_4(A2,b2)
 P  =

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

 C  =

   1.   2.   1.
   0.   1.   1.
   0.   0.   1.
```

```
--> X = Resolve_com_LU(C,P,B2)
 X  =

   0.   3.
   0.  -2.
   1.   1.


--> Resposta_certa = inv(A2)*B2
 Resposta_certa  =

   0.   3.
   0.  -2.
   1.   1.
```

Scilab 6.0.2 Console

```
--> A3
 A3  =

   0.   0.   1.
   0.   1.   1.
   1.   2.   1.


--> B3 = B2
 B3  =

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


--> [x,C,P] = Gaussian_Elimination_4(A3,b2)
 P  =

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

 C  =

   1.   2.   1.
   0.   1.   1.
   0.   0.   1.
```

```
--> X = Resolve_com_LU(C,P,B3)
 X  =

   0.   3.
   0.  -2.
   1.   1.


--> Resposta_certa = inv(A3)*B3
 Resposta_certa  =

   0.   3.
   0.  -2.
   1.   1.
```