Trabalho Prático 2

João Pedro de Abreu Marciano

Questão 1 e 2

Os códigos estão em anexo. **Jacobi** é o método de Jacobi. **GaussSeidel_inv** é o método de Gauss-Seidel usando a função inversa. **GaussSeidel_sist** é o método de Gauss-Seidel usando uma função criada no próprio código chamada **sist** para resolver o sistema com a matriz triangular inferior.

Questão 3

Testando sistema dado esperamos encontrar a solução $x = (\frac{1}{4}, -\frac{1}{4}, \frac{3}{8})$.

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

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


-->x0 = [0;0;0]
 x0  =

   0.
   0.
   0.


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

   2.
   1.
   1.


-->[x, d, k, r] = Jacobi(A, b, x0, 10^(-8), 1000000, 2)
at line    17 of function Jacobi ( /home/joao/Documentos/Algebra-Linear-Numerica/AulaPratica2/Jacob

norm: Wrong value for argument #1: Must not contain NaN or Inf.
```

Nesse caso, o motivo do erro ter ocorrido é que a diagonal de A não é estritamente dominante, ou seja, não temos a condição de convergência do método e ele diverge. E o mesmo ocorreu para o método de Gauss-Seidel.

```
-->[x, d, k, r] = GaussSeidel_inv(A, b, x0, 10^(-8), 1000000, 2)
at line    15 of function GaussSeidel_inv ( /home/joao/Documentos/

norm: Wrong value for argument #1: Must not contain NaN or Inf.
```

Alterando a matriz A para que ela possua diagonal estritamente dominante.

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

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


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

   1.
   2.
   1.
```

```
-->[x, d, k, r] = Jacobi(A, b, x0, 10^(-8), 1000000, 2)
 r  =

   1.344D-08

 k  =

   26.

 d  =

   4.834D-09

 x  =

   0.25
  -0.25
   0.375
```

```
-->[x, d, k, r] = GaussSeidel_inv(A, b, x0, 10^(-8), 1000000, 2)
 r  =

   3.881D-09

 k  =

   14.

 d  =

   4.339D-09

 x  =

   0.25
  -0.25
   0.375
```

```
-->[x, d, k, r] = GaussSeidel_sist(A, b, x0, 10^(-8), 1000000, 2)
 r  =

   3.881D-09

 k  =

   14.

 d  =

   4.339D-09

 x  =

   0.25
  -0.25
   0.375
```

Encontramos o resultado desejado em todas as funções. O método de Gauss-Seidel conseguiu a tolerancia desejada em 14 iterações e o método de Jacobi em 26.

Questão 4

Item A

Fazendo 25 iterações do método de Jacobi no sistema obtemos:

```
-->[x, d, k, r] = Jacobi(A, b, x0, 0, 25, 2)
 r  =

   120.54837

 k  =

   25.

 d  =

   55.650375

 x  =

   11.913936
   45.655746
  -11.913936
```

Podemos ver que o valor encontrado nessas iterações está muito longe da solução esperada, com $r = 120,55$ e $d = 55,65$. Um dos motivos para que isso possa ter acontecido é que novamente a diagonal de A não é estritamente dominante, então não há garantia de convergência para o método.

Item B

Utilizando o méto de Gauss-Seidel no mesmo sistema obtemos:

```
-->[x, d, k, r] = GaussSeidel_inv(A, b, x0, 10^(-5), 1000000, %inf)
 r  =

   0.0000069

 k  =

   22.

 d  =

   0.0000073

 x  =

   1.0000023
   1.9999975
  -1.0000001
```

Nesse caso conseguimos uma aproximação muito boa com apenas 22 iterações. Isso pode ter se dado à princípio pelo mudança de norma ou pelo mudança de método. Vale observar que a condição de convergência também não é satisfeita para o método de Gauss-Seidel, porém não é obrigatório que a condição seja satisfeita para que haja convergência.

Desenvolvendo as hipóteses, pode ser que haja convergncia em uma norma e divergência em outra. O mesmo vale para os métodos. Porém, com um teste, vemos que o método de Jacobi também não aproxima bem mesmo com norma infinito. Logo, o motivo da convergência foi a mudança de métodos.

```
-->[x, d, k, r] = Jacobi(A, b, x0, 10^(-5), 100, %inf)
 r  =

   420389.54

 k  =

   100.

 d  =

   168155.82

 x  =

  -105096.38
   2.
  -105098.38
```

Questão 5

Item a

```
-->[x, d, k, r] = GaussSeidel_inv(A, b, x0, 10^(-2), 300, %inf)
 r  =

   0.0040412

 k  =

   12.

 d  =

   0.0064659

 x  =

   0.8975131
  -0.8018652
   0.7015543
```

Atingiu a tolerância especificada com 12 iterações. Em outras normas os resultados foram bem similares.

Item B

```
-->[x, d, k, r] = GaussSeidel_inv(A, b, x0, 10^(-2), 300, 2)
 r  =

   7.098D+41

 k  =

   300.

 d  =

   6.992D+41

 x  =

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

Nesse caso, vemos que o método não convergiu. Novamente vale observar que a matriz não possui a diagonal estritamente dominante, então o sistema não cumpria a condição de convergência.

Questão 6

Rodando o código **questao6.sce** em anexo obtemos os seguintes tempos:

```
-->exec('/home/joao/Documentos/Algebra-Linear-Numerica/AulaPratica2/questao6.sce',-1)

   0.000177   0.001454   1.826055   14.242667
   0.000299   0.002322   0.064194   0.245397
```

Onde a primeira linha são os tempos do **GaussSeidel_inv** e a segunda **GaussSeidel_sist**.

Podemos perceber que para um número maior de elementos a função *inv* demora bem mais que resolver o sistema.