MS211 - Cálculo Numérico - Turma J - Atividade 3

Luiz Henrique Costa Freitas RA: 202403
Gabriel de Freitas Garcia RA: 216179
Vanessa Vitória de Arruda Pachalki RA: 244956

27 de outubro de 2020

Enunciado Atividade 3

Considere $n = 1000$, $A = rand(n, n)$ e a matriz de Hilbert H de ordem n: $h_{ij} = \frac{1}{i+j-1}$. O Octave fornece a função $hilb(n)$ para criar essa matriz.

Considere os sistemas lineares $Ax = b$ e $Hx = c$, onde x é o vetor de *uns*.

1. Utilize a função $lu()$ do Octave para calcular a decomposição LU de A e de H.
2. Resolva os sistemas lineares utilizando L e U das decomposições calculadas.
3. Verifique os reultados obtidos calculando as seguintes normas: $\frac{||b-Ay||}{||y||}$, $\frac{||c-Hy||}{||y||}$, $\frac{||x-y||}{||y||}$, onde y é a solução aproximada encontrada por cada um dos métodos. Também compare os resultados com o obtido pelo comando do *Octave* $x = A\backslash b$.
4. Calcule o determinante de A e H usando as respectivas matrizes U. Compare com a função $det()$.
5. Faça $n = 100$. Calcule B, a inversa da nova matriz A usando sua decomposição LU. Compare com a inversa obtida pelo comando $inv()$ calculando $norm(B - inv(A))$.

1 Respostas

Para resolução do item 1, desenvolvemos o código a seguir.

```
# Sou um lab feliz
# Jesus, nosso Senhor, te ama.
# Exercício 1
n = 1000;
A = rand(n,n);
H = hilb(n);
[La,Ua,Pa] = lu(A);
[Lh,Uh,Ph] = lu(H);

disp(norm(Pa*A - La*Ua));
disp(norm(Ph*H - Lh*Uh));

#GO POCKETS!!!
```

Em seguida, com o intuito de resolver os sistemas lineares utilizando as matrizes L e U sendo L uma matriz triângular inferior e U uma matriz triângular superior, fizemos algumas funções de acordo com o algoritmo passado em aula. Consideramos que, dado um sistema $Ax = b$ então $PA = LU$. Com isso foi possível resolver as expressões triângulares a seguir.

$$\begin{cases} Ly = Pb \\ Ux = y \end{cases} \tag{1}$$

No código desenvolvido, a função *TriangInf* é responsável por calcular o valor de y do sistema acima utilizando a primeira equação e a função *TriangSup* calcula o valor de x a partir da segunda equação. Para a resolução do item 4 da atividade, criamos a função *dTriang* para calcular o determinante de uma matriz triângular. O código dessas funções está descrito a seguir.

```
#Definição da função triangInf
function Y = triangInf(L,P,B,n)
  Y = zeros(n,1);
  b = P*B;
```

```
  Y(1,1) = b(1,1)/L(1,1);
  for k = 2:n
    soma = b(k,1);
    j = k-1;
   while (j > 0)
     soma = soma - L(k,j)*Y(j,1);
     j = j -1;
   endwhile
   Y(k,1) = soma/L(k,k);
  endfor
endfunction

#Definição da função triangSup
function X = triangSup(U,y,n)
  X = zeros(n,1);
  X(n,1) = y(n,1)/U(n,n);
  k = (n-1);
  while (k > 0)
    soma = y(k,1);
   for j = k+1:n
     soma = soma - U(k,j)*X(j,1);
   endfor
   X(k,1) = soma/U(k,k);
   k = k - 1;
  endwhile
endfunction

#Definição da função dTriang
function dete = dTriang(A,n)
  dete = 1;
  for i = 1:n
    dete = dete * A(i,i);
  endfor
endfunction
```

Resolvemos os sistemas lineares para a matriz A e para a matriz H aplicando o código mostrado abaixo. Vale ressaltar que para cada execução do programa, uma matriz A diferente é gerada, visto que ela é definida utilizando valores aleatórios.

```
# Exercício 2 3 e 4 - A
n = 1000;
A = rand(n,n);
# x = ones(n,1);
b = A*ones(n,1);
[La,Ua,Pa] = lu(A);
ya = triangInf(La,Pa,b,n);
xa = triangSup(Ua,ya,n);
dete = dTriang(Ua,n);
dete2 = det(A);

disp(norm(b-(A*xa))/norm(xa));
disp(norm(ones(n,1)-xa)/norm(xa));
disp(dete);
disp(dete2);

#Exercício 2 3 e 4 - H
n = 1000;
H = hilb(n);
# x = ones(n,1);
c = H*ones(n,1);
[Lh,Uh,Ph] = lu(H);
yh = triangInf(Lh,Ph,c,n);
xh = triangSup(Uh,yh,n);
dete = dTriang(Uh,n);
dete2 = det(H);

disp(norm(c-(H*xh))/norm(xh));
disp(norm(ones(n,1)-xh)/norm(xh));
disp(dete);
disp(dete2);
```

Quando resolvemos o sistema da a matriz A para o item 2, obtivemos como resultado o vetor de *uns*, conforme esperado. Já para a matriz de *Hilbert*, os valores encontrados divergiram do aguardado, apresentando valores em módulo variando de dezenas de milhares até valores menores que um. A fim de verificar os resultados, conforme pede o item 3, calculamos a norma e as mostramos nas tabelas 1 e 2

Norma	Valor
$\frac{\|\|b-Axa\|\|}{\|\|xa\|\|}$	$9.9612e-13$
$\frac{\|\|x-xa\|\|}{\|\|xa\|\|}$	$9.4282e-13$

Tabela 1: Verificação dos resultados para o sistema $Ax = b$

Norma	Valor
$\frac{\|\|c-Hxh\|\|}{\|\|xh\|\|}$	$7.6032e-18$
$\frac{\|\|x-xh\|\|}{\|\|xh\|\|}$	1.00000

Tabela 2: Verificação dos resultados para o sistema $Hx = c$

Nas tabelas, a primeira linha representa o erro relativo ao cálculo de Ax e o vetor b e c, respectivamente. A segunda linha das tabelas representa o erro relativo entre os vetores obtidos e os vetores de *uns* originalmente usados.

Os valores apresentados na Tabela 1 corroboram com o nosso resultado apresentado para a resolução do sistema, uma vez que as normas calculadas são muito próximas de zero. Contudo, pela Tabela 2 podemos perceber que a primeira norma calculada foi próxima de zero, entretanto a segunda não. Isso ocorre principalmente pelo fato de a matriz de *Hilbert* ser uma matriz mal condicionada, ou seja, ao realizar operações numéricas com essa matriz, pequenas imprecisões nas entradas geram erros significativos na saída, podendo passar de 1000%. Logo, como realizamos diversas operações com números em precisão finita e utilizamos uma matriz de *Hilbert* muito grande, a cada etapa o erro crescia, produzindo ao final um resultado totalmente distoante do esperado.

A diferença entre os dois resultados da Tabela 2 se dá porque na primeira fórmula são realizadas novas operações utilizando a matriz de *Hilbert*, reproduzindo os mesmos erros. Já na segunda, a comparação é feita utilizando o vetor de *uns* original, sem nenhum erro, revelando assim a discrepância.

Por fim, utilizamos a operação $x = A\backslash b$ e obtivemos o vetor de *uns*. Quando realizamos a operação $x = H\backslash c$ obtivemos um vetor totalmente diferente do vetor de *uns*, reafirmando os resultados já discutidos.

Para o item 4, computamos o determinante das matrizes A e H utilizando as matrizes Ua e Uh através da função $dTriang$ descrita anteriormente. Para uma matriz $M = LU$ temos que $det(L) = 1$, então $det(M) = det(U)$. A partir disso e da função nativa $det()$ do *Octave* obtivemos os resultados apresentados na Tabela 3.

Método	Matriz A	Matriz H
Função do Octave	∞	0
Função $dTriang$	∞	0

Tabela 3: Valores obtidos para os determinantes da matriz A e H.

Esses resultados não significam, no entanto, que o determinantes seja zero ou infinito. Considerando que essas matrizes são da ordem $1000x1000$, os cálculos produzem números tão grandes (no caso da matriz A) ou tão pequenos (no caso da matriz H) que o computador não é capaz de representá-los. Por exemplo, uma matriz triângular em que todos os elementos da diagonal sejam iguais a 1.1 possuirá o determinante $det = 2.47x10^{41}$ e quando utilizamos os elementos iguais a 0.99, obtivemos o valor do determinante $det = 4.3x10^{-5}$, porém para elementos da diagonal iguais a 0.9, o resultado obtido foi nulo. Esses exemplos demonstram o que aconteceu para os resultados das matrizes A e H.

Para o item 5 desenvolvemos o código a seguir.

```
# Exercício 5
n = 100;
A = rand(n,n);
I = eye(n);
[L,U,P] = lu(A);
B = (L*U\P*I);
D = inv(A);
disp(norm(B-D));
```

Este código se baseia na seguintes operações matemáticas: Seja um sistema $Ax = b$, se $b = I$ e $X = x$ uma matriz, então $AX = I$ e $X = A^{-1}$. Porém $PA = LU$, desta forma temos que $P^{-1}LUX = I$ e $X = (LU)^{-1}PI$, que é o valor da inversa de A.

Executando este código e comparando os resultados com a inversa obtida pelo *Octave* através da norma da diferença das duas, obtivemos que $||B - A^{-1}|| = 7.6568e - 13$, que é suficientemente próximo de zero para considerarmos as duas matrizes aproximadamente iguais, demonstrando assim a corretude do nosso algoritmo.