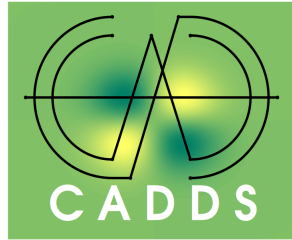


Manual de Laboratorio:
FACTORIZACIONES MATRICIALES LU Y DE CHOLESKY



Prof. Dr. Fredy Vides
Scientific Computing Innovation Center, UNAH &
Centre for Analysis of Data-Driven Systems
E-mail: fredy.vides@unah.edu.hn

ÍNDICE

Objetivos	1
1. Cómputo de Factorizaciones Matriciales LU y de Cholesky	1
1.1. Ejercicio de Laboratorio 1	3
1.2. Ejercicio de Laboratorio 2	4
Referencias	5

OBJETIVOS

1. Aplicar GNU Octave para el cálculo de factorizaciones matriciales LU y de Cholesky.

1. CÓMPUTO DE FACTORIZACIONES MATRICIALES LU Y DE CHOLESKY

Considerando una matriz $A_{h,N} \in \mathbb{R}^{N \times N}$ de la forma:

$$A_{h,N} = -\frac{1}{h^2} \begin{bmatrix} -2 & 1 & 0 & 0 & 0 & \cdots & 0 \\ 1 & -2 & 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & -2 & 1 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & -2 & 1 & 0 \\ 0 & \cdots & 0 & 0 & 1 & -2 & 1 \\ 0 & \cdots & 0 & 0 & 0 & 1 & -2 \end{bmatrix}$$

para $h > 0$. Es posible generar la matriz $A_{h,N}$ utilizando la siguiente secuencia de comandos en Octave:

```
>> A=@(h,N) (-1/(h^2))*spdiags(ones(N,1)*[1 -2 1],[-1:1,N,N]);
```

En particular, para $h = 1/10$ y $N = 9$, la matriz $A_{1/10,9} \in \mathbb{R}^{9 \times 9}$ puede calcularse utilizando la función A recién definida, utilizando la siguiente secuencia de comandos:

```
>> A9=A(.1,10)
```

```
A9 =
```

```
Compressed Column Sparse (rows = 10, cols = 10, nnz = 28 [28%])
```

```
(1, 1) -> 200.00
(2, 1) -> -100.00
(1, 2) -> -100.00
(2, 2) -> 200.00
(3, 2) -> -100.00
(2, 3) -> -100.00
(3, 3) -> 200.00
(4, 3) -> -100.00
(3, 4) -> -100.00
(4, 4) -> 200.00
(5, 4) -> -100.00
(4, 5) -> -100.00
(5, 5) -> 200.00
(6, 5) -> -100.00
(5, 6) -> -100.00
(6, 6) -> 200.00
(7, 6) -> -100.00
(6, 7) -> -100.00
(7, 7) -> 200.00
(8, 7) -> -100.00
(7, 8) -> -100.00
(8, 8) -> 200.00
(9, 8) -> -100.00
(8, 9) -> -100.00
(9, 9) -> 200.00
(10, 9) -> -100.00
(9, 10) -> -100.00
(10, 10) -> 200.00
```

La matriz A9 está en formato esparcido (los coeficientes iguales a cero no se almacenan en memoria para ahorrar espacio en memoria), para visualizar la matriz A9 en formato denso, puede escribirse (aunque para ahorrar memoria no se recomienda, especialmente para matrices de gran tamaño) la siguiente secuencia de comandos:

```
>> full(A9)
```

```
ans =
```

```
Columns 1 through 6:
```

```
200.00000  -100.00000   0.00000   0.00000   0.00000   0.00000
-100.00000  200.00000 -100.00000   0.00000   0.00000   0.00000
  0.00000  -100.00000  200.00000 -100.00000   0.00000   0.00000
  0.00000   0.00000 -100.00000  200.00000 -100.00000   0.00000
  0.00000   0.00000   0.00000 -100.00000  200.00000 -100.00000
```

```

0.00000    0.00000    0.00000    0.00000   -100.00000    200.00000
0.00000    0.00000    0.00000    0.00000    0.00000   -100.00000
0.00000    0.00000    0.00000    0.00000    0.00000    0.00000
0.00000    0.00000    0.00000    0.00000    0.00000    0.00000
0.00000    0.00000    0.00000    0.00000    0.00000    0.00000

```

Columns 7 through 10:

```

0.00000    0.00000    0.00000    0.00000
0.00000    0.00000    0.00000    0.00000
0.00000    0.00000    0.00000    0.00000
0.00000    0.00000    0.00000    0.00000
0.00000    0.00000    0.00000    0.00000
-100.00000    0.00000    0.00000    0.00000
200.00000   -100.00000    0.00000    0.00000
-100.00000    200.00000   -100.00000    0.00000
0.00000   -100.00000    200.00000   -100.00000
0.00000    0.00000   -100.00000    200.00000

```

Las matrices de la forma $A_{h,N}$ tienen aplicaciones en la solución numérica de ecuaciones diferenciales parciales y ordinarias aplicadas en ciencias e ingeniería.

1.1. Ejercicio de Laboratorio 1. Calcular la matriz $A_{\frac{1}{1000},999} \in \mathbb{R}^{999 \times 999}$.

1.1.1. *Factorización LU.* Aplicando el programa `LU.m` definido por el código Octave.

```

function [L,U]=LU(A)
% Descomposicion LU
% A : Matriz
% L : Matrix triangular inferior t.q. A = LL'
% Programador: Fredy Vides
n=size(A,1);
L=eye(n);
U=A(1,:);
for k=1:n,
    U(k,k)=A(k,k)-L(k,1:(k-1))*U(1:(k-1),k);
    for j=(k+1):n,
        U(k,j)=(A(k,j)-L(k,1:(k-1))*U(1:(k-1),j));
    end
    for i=(k+1):n
        L(i,k)=(A(i,k)-L(i,1:(k-1))*U(1:(k-1),k))/U(k,k);
    end
end
end

```

Calcular:

1. La factorización $LU = A_{\frac{1}{1000},999}$
2. El error normado $\|LU - A_{\frac{1}{1000},999}\|_{\infty}$ correspondiente.

Solución. Para calcular la factorización LU de A en Octave es posible aplicar el programa `LU.m` utilizando la siguiente secuencia de comandos:

1. Generar la matriz de prueba $A_{h,N}$ para el algoritmo:

```
>> A=@(h,N)(-1/(h^2))*spdiags(ones(N,1)*[1 -2 1],[-1:1,N,N]);
>> AhN=A(1/1000,999);
```

2. Calcular la factorización LU de la matriz $A_{h,N}$ correspondiente, estimando el tiempo de ejecución del programa con los comandos `tic-toc` de Octave:

```
>> tic,[L,U]=LU(AhN);toc
Elapsed time is 23.2845 seconds.
```

3. Estimar el error absoluto de aproximación de la factorización LU aproximada de $A_{h,N}$ producida por el programa, con respecto a la norma $\|\cdot\|_\infty$, utilizando la siguiente secuencia de comandos:

```
>> norm(AhN-L*U,inf)
ans = 0.00000000011642
```

4. También es posible estimar el error relativo de aproximación de la factorización LU aproximada de $A_{h,N}$ producida por el programa, con respecto a la norma $\|\cdot\|_\infty$, utilizando la siguiente secuencia de comandos:

```
>> norm(AhN-L*U,inf)/norm(AhN,inf)
ans = 2.9104e-17
```

1.1.2. *Factorización de Cholesky.* Aplicando el programa `Chole.m` definido por el código Octave.

```
function L=Chole(A)
% L = Chole (A)
% Descomposicion elemental de Cholesky
% A : Matriz SPD
% L : Matrix triangular inferior t.q. A = LL'
% Programador: Fredy Vides
n=size(A,1);
L(1,1)=sqrt(A(1,1));
L(2:n,1)=A(2:n,1)/L(1,1);
for k=2:n,
    L(k,k)=sqrt(A(k,k)-L(k,1:(k-1))*L(k,1:(k-1))');
    for i=(k+1):n
        L(i,k)=(A(i,k)-L(i,1:(k-1))*L(k,1:(k-1))')/L(k,k);
    end
end
```

Calcular:

1. La factorización $LL^T = A_{\frac{1}{1000},999}$
2. El error normado $\|LL^T - A_{\frac{1}{1000},999}\|_\infty$ correspondiente.

1.2. Ejercicio de Laboratorio 2. Modificar los programas `LU.m` y `Chole.m` definidos anteriormente para realizar un cómputo más eficiente (en el sentido del número de operaciones realizadas, uso de la memoria y tiempo de cómputo) en el caso especial de matrices tridiagonales como las matrices $A_{h,N}$ definidas anteriormente.

Resolver nuevamente le Ejercicio de Laboratorio 1 con las nuevas versiones de `LU.m` y `Chole.m`, justificando de forma teórica o computacional que las nuevas versiones de los programas propuestas por usted, son en efecto más eficientes.

REFERENCIAS

- [1] J. W. Eaton (2020). GNU Octave (version 5.2.0) Documentation. <https://octave.org/doc/v5.2.0/>
- [2] R. L. Burden, D. J. Faires, A. M. Burden. (2017). Análisis Numérico. 10a Ed. Cengage Learning Editores.
- [3] Quarteroni A., Saleri F., Gervasio P. (2014). Scientific computing with MATLAB and Octave (Textbook).
- [4] D. G. Luenberger, Y. Ye. (2016). Linear and Nonlinear Programming. 4a Ed. Springer International Publishing Switzerland.