Libro: Quarteroni, “Scientific Computing with MATLAB and Octave”

Cap5: Sistemas Lineales

Regla de Cramer

Table 5.1. Time required to solve a linear system of dimension n by the Cramer rule. “o.r.” stands for “out of reach”

	Flops				
n	10^9 (Giga)	10^{10}	10^{11}	10^{12} (Tera)	10^{15} (Peta)
10	10^{-1}sec	10^{-2}sec	10^{-3}sec	10^{-4}sec	negligible
15	17 hours	1.74 hours	10.46 min	1 min	0.6 10^{-1} sec
20	4860 years	486 years	48.6 years	4.86 years	1.7 day
25	o.r.	o.r.	o.r.	o.r.	38365 years

Eliminación Gaussiana

$$
\begin{aligned}
&\text{for } k = 1, \ldots, n-1\\
&\qquad \text{for } i = k+1, \ldots, n\\
&\qquad\qquad l_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}},\\
&\qquad\qquad \text{for } j = k+1, \ldots, n\\
&\qquad\qquad\qquad a_{ij}^{(k+1)} = a_{ij}^{(k)} - l_{ik} a_{kj}^{(k)}
\end{aligned}
$$

$$
\begin{bmatrix}
a_{11}^{(1)} & a_{12}^{(1)} & \dots & & \dots\dots & a_{1n}^{(1)}\\
l_{21} & a_{22}^{(2)} & & & & a_{2n}^{(2)}\\
\vdots & \ddots & \ddots & & & \vdots\\
l_{k1} & \dots & l_{k,k-1} & \boxed{a_{kk}^{(k)}} & \dots & \boxed{a_{kn}^{(k)}}\\
\vdots & & \vdots & \vdots & & \vdots\\
l_{n1} & \dots & l_{n,k-1} & \boxed{a_{nk}^{(k)}} & \dots & \boxed{a_{nn}^{(k)}}
\end{bmatrix}
$$

$$
\begin{aligned}
&\text{for } k = 1, \dots, n-1 \\
&\qquad \text{for } i = k+1, \dots, n \\
&\qquad\qquad l_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}}, \\
&\qquad\qquad \text{for } j = k+1, \dots, n \\
&\qquad\qquad\qquad a_{ij}^{(k+1)} = a_{ij}^{(k)} - l_{ik} a_{kj}^{(k)} \\
&\qquad\qquad b_i^{(k+1)} = b_i^{(k)} - l_{ik} b_k^{(k)}
\end{aligned}
$$

Programa lugauss.m:

```
function A=lugauss (A)
%LUGAUSS LU factorization without pivoting .
% A = LUGAUSS(A) stores an upper triangular matrix in
% the upper triangular part of A and a lower triangular
% matrix in the strictly lower part of A (the diagonal
% elements of L are 1).
[n,m]= size (A);
if n ~= m; error('A is not a square matrix'); else
for k = 1:n-1
    for i = k+1:n
    A(i,k) = A(i,k)/A(k,k);
    if A(k,k) == 0, error('Null diagonal element'); end
    j = [k+1:n]; A(i,j) = A(i,j) - A(i,k)*A(k,j);
    end
end
end
```

EJEMPLO 5.5:

A =

-0.3700	0.0500	0.0500	0.0700
0.0500	-0.1160	0	0.0500
0.0500	0	-0.1160	0.0500
0.0700	0.0500	0.0500	-0.2020

b =

-2

0

0

0

>> A= lugauss(A); y(1)= b(1);

```
>> for i=2:4; y=[y; b(i)-A(i,1:i-1)* y(1:i -1)]; end
>> x(4)= y(4)/ A(4 ,4);
>> for i=3:-1:1
x(i)=(y(i)-A(i,i+1:4)*x(i+1:4)')/A(i,i); end
>> x
x =
   8.1172    5.9893    5.9893    5.7779
```

Factorización LU (A=LU)

$$\boxed{\begin{aligned} y_1 &= \frac{1}{l_{11}} b_1, \\ y_i &= \frac{1}{l_{ii}} \left(b_i - \sum_{j=1}^{i-1} l_{ij} y_j \right), \; i = 2, \ldots, n \end{aligned}}$$

$$\boxed{\begin{aligned} x_n &= \frac{1}{u_{nn}} y_n, \\ x_i &= \frac{1}{u_{ii}} \left(y_i - \sum_{j=i+1}^{n} u_{ij} x_j \right), \; i = n-1, \ldots, 1 \end{aligned}}$$

EJEMPLO 5.6:

Para e=1

```
A =
     1     0     3
     2     2     2
     3     6     4
```

Usando la función lugauss, obtenemos L y U:

```
L =

    1     0     0
    2     1     0
    3     3     1

U =

    1     0     3
    0     2    -4
    0     0     7
```

Ahora, para e=0

```
A =

    1     1     3
    2     2     2
    3     6     4

>> A=lugauss(A)
Error using lugauss (line 12)
Null diagonal element
```

EJEMPLO 5.8:

```
A =

   1.000000000000000   1.000000000000000   3.000000000000000
   2.000000000000000   2.000000000000000  20.000000000000000
   3.000000000000000   6.000000000000000   4.000000000000000

>> A(1,2)-A(1,1)

ans =

    4.440892098500626e-16
```

Usando la función lugauss, obtenemos L y U:

```
L =

   1.0e+15 *

   0.000000000000001                   0                   0
   0.000000000000002   0.000000000000001                   0
   0.000000000000003  -3.377699720527871   0.000000000000001
```

U =

1.0e+16 *

```
  0.000000000000000   0.000000000000000   0.000000000000000
                  0  -0.000000000000000   0.000000000000001
                  0                   0   4.728779608739018
```

Calculamos A-LU:

>> A-L*U

ans =

```
     0     0     0
     0     0     0
     0     0     4
```

PIVOTEO

Algoritmo para descomposición LU con pivoteo parcial:

for $k = 1, \ldots, n-1$,

find $\bar{r}$ such that $|a_{\bar{r}k}^{(k)}| = \max_{r=k,\ldots,n} |a_{rk}^{(k)}|$,

exchange row k with row $\bar{r}$
in both A and P,

for $i = k+1, \ldots, n$

$$l_{ik} = \frac{a_{ik}^{(k)}}{a_{kk}^{(k)}},$$

for $j = k+1, \ldots, n$

$$a_{ij}^{(k+1)} = a_{ij}^{(k)} - l_{ik} a_{kj}^{(k)}$$

Dos formas de usar el programa lu del MATLAB:

FORMA 1:

```
>> [L,U]=lu(A)

L =

  0.333333333333333   0.500000000000000   1.000000000000000
  0.666666666666667   1.000000000000000                   0
  1.000000000000000                   0                   0

U =

  3.000000000000000   6.000000000000000   4.000000000000000
                  0  -2.000000000000000  17.333333333333332
                  0                   0  -6.999999999999996

>> A-L*U

ans =

  1.0e-15 *

                  0                   0  -0.888178419700125
                  0                   0                   0
                  0                   0                   0
```

FORMA 2:

```
>> [L,U,P]=lu(A)

L =

  1.000000000000000                   0                   0
  0.666666666666667   1.000000000000000                   0
  0.333333333333333   0.500000000000000   1.000000000000000

U =

  3.000000000000000   6.000000000000000   4.000000000000000
                  0  -2.000000000000000  17.333333333333332
                  0                   0  -6.999999999999996

P =

   0   0   1
   0   1   0
   1   0   0
```

```
>> P*A-L*U

ans =

  1.0e-15 *

              0              0              0
              0              0              0
              0              0  -0.888178419700125
```

Factorización de Cholesky (A=R'R)

$$r_{ij} = \frac{1}{r_{ii}}\left(a_{ij} - \sum_{k=1}^{i-1} r_{ki} r_{kj}\right),\ i = 1, \ldots, j-1$$

$$r_{jj} = \sqrt{a_{jj} - \sum_{k=1}^{j-1} r_{kj}^2}$$