Salvador Macías Hernández
Rodolfo Gómez Vega

# FACTORIZACIÓN LU. PROGRAMA

7/9/2003 Tarea No.4

INTRODUCCIÓN. Supongamos que A se puede factorizar como el producto de una matriz triangular inferior L con una matriz triangular superior U:

$$A = LU$$

En este caso, el sistema de ecuaciones dado por podría representarse en la forma:

$$Lux=b$$

Si denominamos z a la matriz columna de n filas resultado del producto de las matrices Ux, tenemos que la ecuación anterior se puede rescribir del siguiente modo:

$$Lz=b$$

A partir de las ecuaciones anteriores, es posible plantear un algoritmo para resolver el sistema de ecuaciones empleando dos etapas:

T Primero obtenemos z aplicando el algoritmo de sustitución progresiva en la última ecuación.

T Posteriormente obtenemos los valores de x aplicando el algoritmo de sustitución regresiva a la ecuación

$$Ux = z$$

El análisis anterior nos muestra lo fácil que es resolver estos dos sistemas de ecuaciones triangulares y lo útil que resultaría disponer de un método que nos permitiera llevar a cabo la factorización A=LU. Si disponemos de una matriz A de nxn , estamos interesados en encontrar aquellas matrices:

$$L = \begin{pmatrix} l_{11} & 0 & 0 & \cdots & 0 \\ l_{21} & l_{22} & 0 & \cdots & 0 \\ l_{31} & l_{32} & l_{33} & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ l_{n1} & l_{n2} & l_{n3} & \cdots & l_{nn} \end{pmatrix} \quad U = \begin{pmatrix} u_{11} & u_{12} & u_{13} & \cdots & u_{14} \\ 0 & u_{22} & u_{23} & \cdots & u_{2n} \\ 0 & 0 & u_{33} & \cdots & u_{3n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & u_{nn} \end{pmatrix}$$

tales que cumplan la ecuación A=LU. Cuando esto es posible, decimos que A tiene una descomposición LU. Se puede ver que las ecuación anterior no determina de forma única a Ly a U. De hecho, para cada i podemos asignar un valor distinto de cero a lii o uii (aunque no ambos).

Por ejemplo, una elección simple es fijar lii para i=1,2,3 ... n haciendo de este modo que L sea una matriz triangular inferior unitaria. Otra elección es hacer U una matriz triangular superior unitaria (tomando uii=1 para cada i).
Para deducir un algoritmo que nos permita la factorización LU de A partiremos de la fórmula para la multiplicación de matrices:

$$a_{ij} = \sum_{s=1}^{n} l_{is} u_{sj} = \sum_{s=1}^{\text{mín}(i,j)} l_{is} u_{sj}$$

en donde nos hemos valido del hecho de que lis=0 para s >i y usj=0 para s>j.
En este proceso, cada paso determina una nueva fila de U y una nueva columna de L. En el paso k, podemos suponer que ya se calcularon las filas de U, al igual que las columnas de L. Haciendo i=j=k en la ecuación anterior obtenemos:

$$a_{kk} = l_{kk} u_{kk} \sum_{s=1}^{k-1} l_{ks} u_{sk}$$

Si especificamos un valor para lkk (o para ukk), a partir de la ecuación es posible determinar un valor para el otro término. Conocidas ukk y lkk y a partir de la ecuación anterior a esta podemos escribir las expresiones para la k-ésima fila (i=k) y para la k-ésima columna (j=k), respectivamente:

$$a_{kj} = l_{kk} u_{kj} + \sum_{s=1}^{k-1} l_{ks} u_{sj}$$

$$a_{ik} = l_{ik} u_{kk} + \sum_{s=1}^{k-1} l_{is} u_{sk}$$

Es decir, las ecuaciones se pueden emplear para encontrar los elementos ukj y lik. El algoritmo basado en el análisis anterior se denomina factorización de Doolittle cuando se toman los términos lii = 1 para (L triangular inferior unitaria) y factorización de Crout cuando se toman los términos uii=1 (U triangular superior unitaria).

EJEMPLO Ejemplo:

Encuentre las factorizaciones de Doolittle y Crout de la matriz:

$$A = \begin{pmatrix} 60 & 30 & 20 \\ 30 & 20 & 15 \\ 20 & 15 & 12 \end{pmatrix}$$

La factorización de Doolittle es, a partir del algoritmo:

$$A = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ \frac{1}{3} & 1 & 1 \end{pmatrix} \begin{pmatrix} 60 & 30 & 20 \\ 0 & 5 & 5 \\ 0 & 0 & \frac{1}{3} \end{pmatrix} = LU$$

En vez de calcular la factorización de Crout directamente, la podemos obtener a partir de la factorización de Doolittle que acabamos de ver. Efectivamente, si tenemos en cuenta que la matriz A es simétrica, es posible comprobar que se cumple la relación:

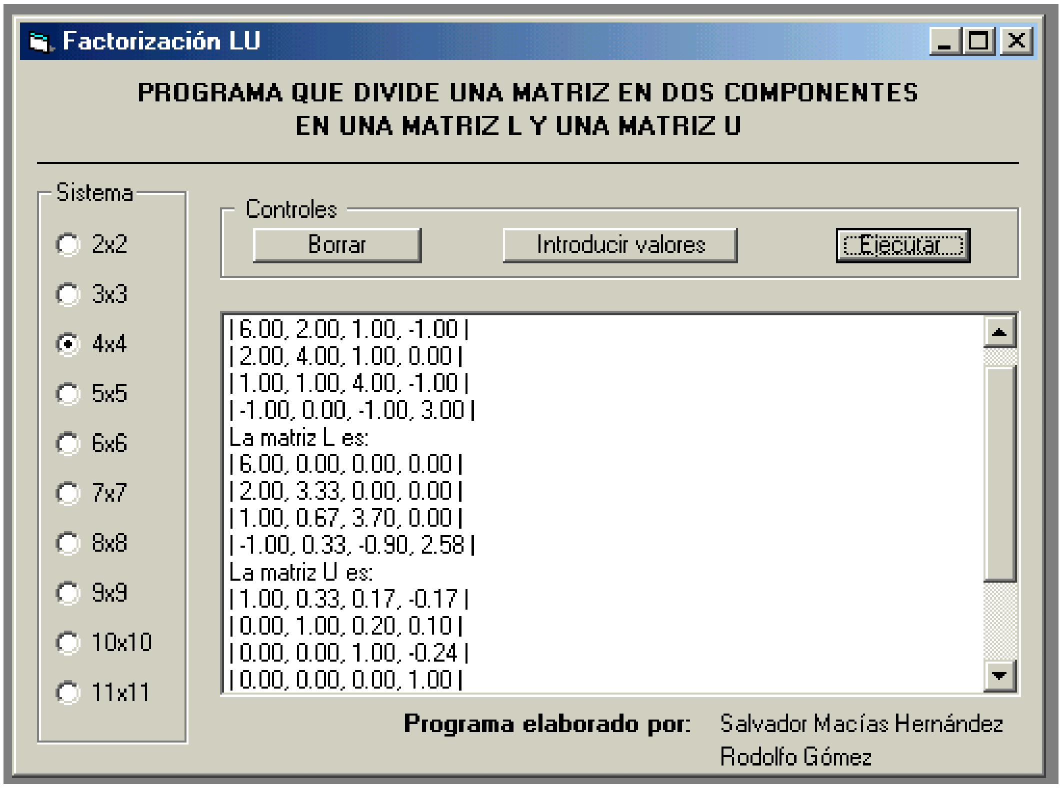
$$A = LU = U^T L^T$$

por lo que la factorización de Crout resulta ser:

$$A = \begin{pmatrix} 60 & 0 & 0 \\ 30 & 5 & 0 \\ 20 & 5 & \frac{1}{3} \end{pmatrix} \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{pmatrix} = U^T L^T$$

**PROGRAMA** El programa que hicimos para la tarea número tres se encarga de separar una matriz A en sos dos componentes L y U por el método de Crout. Cuenta con un ambiente amigable para el usuario y en entorno windows. Gracias a que está hecho en Visual Basic.

Al programa se le corrió el ejercicio de una matriz de 4x4 que se hizo en clase para corroborar su buen funcionamiento. Y los resultados se pueden observar en la siguiente imágen:




**CÓDIGO** El código del programa es el siguiente:

```
Dim MATRIZ() As Double        'Matriz Actual A
Dim MATRIZL() As Double       'Matriz L
Dim MATRIZU() As Double       'Matriz U
Dim MATRIZB() As Double       'Matriz de resultados
Dim MATRIZZ() As Double       'Matriz Z
Dim MATRIZX() As Double       'Matriz X
Dim DIMENSION As Integer      'Dimensión de la matriz
Dim UNO As String             'Comodín de texto
Dim POLAH As Integer          'Comodín entero
Dim ACC As Double             'Comodín Doble
```

```
Private Sub cmdBorrar_Click()  'Subrutina que borra la pantalla
lstPantalla.Clear
End Sub

Private Sub cmdEjecutar_Click()  'Ejecutando el programa
lstPantalla.Clear
lstPantalla.AddItem ("La matriz A es de " & DIMENSION & " x " &
DIMENSION)
Call MUESTRA(MATRIZ())

ReDim MATRIZL(DIMENSION, DIMENSION) 'Definiendo matrices
ReDim MATRIZU(DIMENSION, DIMENSION)

For MI = 1 To DIMENSION            'Dando formato a la Matriz L
 MATRIZL(MI, 1) = MATRIZ(MI, 1)
Next

For MI = 1 To DIMENSION            'Dando formato a la Matriz U
 MATRIZU(MI, MI) = 1
Next
For MB = 2 To DIMENSION
 MATRIZU(1, MB) = MATRIZ(1, MB) / MATRIZ(1, 1)
Next
                                   'Calculando los elementos de las matrices
For MI = 2 To DIMENSION
 For MB = 2 To DIMENSION
  If MB <= MI Then
   ACC = 0
   For POLAH = 1 To MB - 1
    ACC = ACC + MATRIZL(MI, POLAH) * MATRIZU(POLAH, MB)
   Next
     MATRIZL(MI, MB) = (MATRIZ(MI, MB) - ACC)       'Calcula
elementos de la matriz L
  Else
   ACC = 0
   For POLAH = 1 To MI - 1
    ACC = ACC + MATRIZL(MI, POLAH) * MATRIZU(POLAH, MB)
   Next
   MATRIZU(MI, MB) = (MATRIZ(MI, MB) - ACC) / MATRIZL(MI,
MI)  'Calcula elementos de la matriz U
  End If
 Next
Next
```

```
lstPantalla.AddItem ("La matriz L es:")  'Desplegando resultados
Call MUESTRA(MATRIZL())
lstPantalla.AddItem ("La matriz U es:")
Call MUESTRA(MATRIZU())

lstPantalla.AddItem ("Haciendo el producto de matrices...")
MATRIZ() = MULTIPLICAMATRIZ(MATRIZL(), MATRIZU(),
DIMENSION)
Call MUESTRA(MATRIZ())


ReDim MATRIZZ(DIMENSION)              'Calculando el vector Z
For MI = 1 To DIMENSION
 ACC = 0
 For MB = 1 To MI
  ACC = ACC + MATRIZL(MI, MB) * MATRIZZ(MB)
 Next
 MATRIZZ(MI) = (MATRIZB(MI) - ACC) / MATRIZL(MI, MI)
Next
lstPantalla.AddItem ("EL vector Z es:")
Call MUESTRAVECTOR(MATRIZZ())

ReDim MATRIZX(DIMENSION)              'Calculando el vector X
For MI = 0 To DIMENSION - 1
 ACC = 0
 For MB = 0 To MI
  ACC = ACC + MATRIZU(DIMENSION - MI, DIMENSION - MB) *
MATRIZX(DIMENSION - MB)
 Next
 MATRIZX(DIMENSION - MI) = (MATRIZZ(DIMENSION - MI) -
ACC) / MATRIZU(DIMENSION - MI, DIMENSION - MI)
Next
lstPantalla.AddItem ("EL vector X es:")
Call MUESTRAVECTOR(MATRIZX())
End Sub
```

```
Private Sub cmdValores_Click()      'Subrutina que permite ingresar los valores
UNO = " "
ReDim MATRIZ(DIMENSION, DIMENSION)
ReDim MATRIZB(DIMENSION)
For MB = 1 To DIMENSION
 For MI = 1 To DIMENSION + 1
  If MI = DIMENSION + 1 Then
   MATRIZB(MB) = InputBox("El elemento No: (" & MB & "," & MI & ") es:")
  Else
   MATRIZ(MB, MI) = InputBox("El elemento No: (" & MB & "," & MI & ") es:")
  End If
 Next
Next
lstPantalla.AddItem ("La matriz A es:")
Call MUESTRA(MATRIZ)
lstPantalla.AddItem ("El vector B es:")
Call MUESTRAVECTOR(MATRIZB())
End Sub


Public Sub opt1_Click(Index As Integer)  'Definición de la dimensión de la matriz
DIMENSION = Index + 1
End Sub

Private Sub MUESTRA(MATRIZM() As Double)  'Subrutina que muestra la matriz
UNO = "| "
For MI = 1 To DIMENSION
 For MB = 1 To DIMENSION
  If MB = DIMENSION Then
   UNO = UNO & Format$(MATRIZM(MI, MB), "fixed") & " |"
  Else
   UNO = UNO & Format$(MATRIZM(MI, MB), "fixed") & ", "
  End If
 Next
  lstPantalla.AddItem (UNO)
  UNO = "| "
Next
End Sub
```

```
Private Sub MUESTRAVECTOR(MATRIZM() As Double)  'Subrutina
que muestra un vector
UNO = "| "

 For MB = 1 To DIMENSION
  If MB = DIMENSION Then
   UNO = UNO & Format$(MATRIZM(MB), "fixed") & " |"
  Else
   UNO = UNO & Format$(MATRIZM(MB), "fixed") & ", "
  End If
Next
lstPantalla.AddItem (UNO)
End Sub

Function MULTIPLICAMATRIZ(MATRIZA() As Double, MATRIZB()
As Double, DIMENSION) 'Subrutina que multiplica dos matrices
ACC = 0
ReDim PRODUCTO(DIMENSION, DIMENSION) As Double
For MI = 1 To DIMENSION
 For MB = 1 To DIMENSION
  For POLAH = 1 To DIMENSION
   ACC = ACC + MATRIZA(MI, POLAH) * MATRIZB(POLAH, MB)
  Next
  PRODUCTO(MI, MB) = ACC
  ACC = 0
 Next
Next
MULTIPLICAMATRIZ = PRODUCTO()
End Function
```

**CONCLUSIÓN** Después de haber entendido la factorización LU. Y después de los problemas encontrados al hacer el programa. Concluimos que es más fácil y rápido resolver un sistema de ecuaciones por el método de Gauss. Más sin embargo el método LU sigue siendo un método interesante.

**BIBLIOGRAFÍA**

Análisis Numérico
Richard L. Buden; J. Douglas Faire

Numerical methods for scientists and engineers
R.W. Hamming

Metodos Numericos y Programacion Fortran
Mccraken, Daniel D

Visual Basic 6, How to program
Deitel & Deitel T.R. Nieto

http://www.uv.es/~diaz/mn/node28.html