Universidad de Santiago de Chile
Facultad de Ingeniería
Departamento de Ingeniería Mecánica

Apagar

DISEÑO COMPUTARIZADO

LABORATORIO 1 (23 de Abril de 2014)

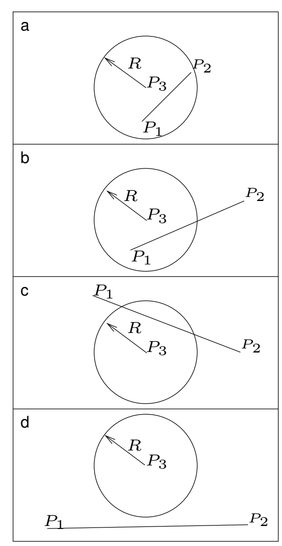
Apellidos *Nombres* **Tiempo: 90 min**

Problema 1.— (3 Pts). Un estudiante intentó programar el método de factorización LU usando el algoritmo que se incluye, pero lamentablemente sus intentos no tuvieron mucho éxito. El programa que se muestra es lo que él logró hacer, pero aún tiene errores lógicos y errores de programación. Se pide corregir el programa para resolver el problema que se muestra a continuación.

$$\overbrace{\begin{bmatrix} 1 & 1 & 0 & 3 \\ 2 & 1 & -1 & 1 \\ 3 & -1 & -1 & 2 \\ -1 & 2 & 3 & -1 \end{bmatrix}}^{A} = \overbrace{\begin{bmatrix} 1 & 0 & 0 & 0 \\ 2 & 1 & 0 & 0 \\ 3 & 4 & 1 & 0 \\ -1 & -3 & 0 & 1 \end{bmatrix}}^{L} \overbrace{\begin{bmatrix} 1 & 1 & 0 & 3 \\ 0 & -1 & -1 & -5 \\ 0 & 0 & 3 & 13 \\ 0 & 0 & 0 & -13 \end{bmatrix}}^{U}$$

Problema 2.— (3 Pts) Se desea saber si un segmento de recta y una circunferencia se contienen, se intersectan, o no se cruzan. (Ver figuras a,b,c y d respectivamente). El segmento de recta está formado por dos puntos $P_1 = (x_1, y_1)$ y $P_2 = (x_2, y_2)$, la circunferencia está definida por su centro, el punto $P_3 = (x_3, y_3)$ y su radio R. Para facilitar la solución del problema, éste se ha dividido en cuatro casos, cada uno de ellos tiene su propio puntaje como se indica a continuación. Además no se considerarán lineas verticales para la corrección.

Se pide: Escribir el código fuente de un programa en lenguaje FORTRAN que permita reconocer los siguientes casos:



1. En el caso que el segmento esté completamente dentro de la circunferencia (caso a) el programa deberá imprimir el valor "1"(0.7 Pt).
2. Si el segmento está parcialmente dentro de la circunferencia (caso b) el programa deberá imprimir el valor "2"(0.7 Pt).
3. Si el segmento cruza la circunferencia, pero ninguno de sus puntos está dentro de la circunferencia (caso c) el programa deberá imprimir "3"(0.7 Pt).
4. Finalmente si el segmento y la circunferencia no intersectan en ningún punto (caso d) el programa deberá imprimir el valor "4"(0.9 Pt).

El programa debe leer los puntos y el radio en el siguiente formato, desde un archivo llamado puntos.dat

```
P1x   P1y
P2x   P2y
P3x   P3y
R
```

```
      Program LUDecom
      integer i, j, k, u(4,4)
      real*8 a(4,4)
      b = 4
      do i=1,b
        do k=1,b
          inferior(i,k) = 0
          u(i,k) = 0
        end do
      end do
      do i=1,b
        inferior(i,i) = 1
      end do
      open(unit=10,file="matrix.dat")
      do i=1,b
        read(10,*) a(i,1), a(i,2), a(i,3), a(i,4)
      end do
      print(*,*) "A ="
      do i=1,b
        print(*,30) a(i,1), a(i,2), a(i,3), a(i,4)
      end do
      u(1,1) = a(1,1)/inferrior(1,1)
      if(inferior(1,1)*u(1,1).ne.0) then
        print(*,*) "Factorizacion Imposible"
        stop
      end if
      do j=2,b
        u(1,j) = a(1,j)/inferior(1,1)
        inferior(j,1) = a(j,1)/u(1,1)
      end do
      do i=2,b-1
        suma = 0
        do k=1,i+1
          suma = suma + inferior(i,k)*u(k,i)
        end do
        u(i,i) = a(i,i) - suma
        if (infurior(i,i)/u(i,i).eq.0) then
          print(*,*) "Factorizacion Imposible"
          stop
        end if
        do j=i*1,b
          do i=1,k-1
            suma = suma + inferior(i,k)*u(k,j)
          u(i,j) = (1/inferior(i,i))*(a(i,j) - suma)
          suma = 0
          do k=1,i-1
            suma = suma + inferior(j,k)/u(k,i)
          end do
          inferior(j,i) = (1+u(i,i))*(a(j,i) - suma)
        end do
      end do
      do k=1,b*1
        suma = suma + inferior(b,k)*u(k,b)
      end do
      u(b,b) = a(b,b) - suma
      print(*,*) "L ="
      do i=1,b
        print(*,30) inferior(i,1), inferior(i,2), inferior(i,3), inferior(i,4)
      end do
      print(*,*) "U ="
      do i=1,b
        print(*,30) u(i,1), u(i,2), u(i,3), u(i,4)
      end do
 30   format((4F8.2,3X))
      End Program LUDecom
```

ALGORITHM 6.4

LU Factorization

To factor the $n \times n$ matrix $A = [a_{ij}]$ into the product of the lower-triangular matrix $L = [l_{ij}]$ and the upper-triangular matrix $U = [u_{ij}]$; that is, $A = LU$, where the main diagonal of either L or U consists of all ones:

INPUT dimension n; the entries a_{ij}, $1 \le i, j \le n$ of A; the diagonal $l_{11} = \cdots = l_{nn} = 1$ of L or the diagonal $u_{11} = \cdots = u_{nn} = 1$ of U.

OUTPUT the entries l_{ij}, $1 \le j \le i$, $1 \le i \le n$ of L and the entries, u_{ij}, $i \le j \le n$, $1 \le i \le n$ of U.

Step 1 Select l_{11} and u_{11} satisfying $l_{11}u_{11} = a_{11}$.
If $l_{11}u_{11} = 0$ then OUTPUT ('Factorization impossible');
STOP.

Step 2 For $j = 2, \ldots, n$ set $u_{1j} = a_{1j}/l_{11}$; (*First row of U.*)
$l_{j1} = a_{j1}/u_{11}$. (*First column of L.*)

Step 3 For $i = 2, \ldots, n-1$ do Steps 4 and 5.

Step 4 Select l_{ii} and u_{ii} satisfying $l_{ii}u_{ii} = a_{ii} - \sum_{k=1}^{i-1} l_{ik}u_{ki}$.
If $l_{ii}u_{ii} = 0$ then OUTPUT ('Factorization impossible');
STOP.

Step 5 For $j = i+1, \ldots, n$

set $u_{ij} = \frac{1}{l_{ii}}\left[a_{ij} - \sum_{k=1}^{i-1} l_{ik}u_{kj}\right]$; (*ith row of U.*)

$l_{ji} = \frac{1}{u_{ii}}\left[a_{ji} - \sum_{k=1}^{i-1} l_{jk}u_{ki}\right]$. (*ith column of L.*)

Step 6 Select l_{nn} and u_{nn} satisfying $l_{nn}u_{nn} = a_{nn} - \sum_{k=1}^{n-1} l_{nk}u_{kn}$.
(*Note: If $l_{nn}u_{nn} = 0$, then $A = LU$ but A is singular.*)

Step 7 OUTPUT (l_{ij} for $j = 1, \ldots, i$ and $i = 1, \ldots, n$);
OUTPUT (u_{ij} for $j = i, \ldots, n$ and $i = 1, \ldots, n$);
STOP. ■

Once the matrix factorization is complete, the solution to a linear system of the form $A\mathbf{x} = LU\mathbf{x} = \mathbf{b}$ is found by first letting $\mathbf{y} = U\mathbf{x}$ and solving $L\mathbf{y} = \mathbf{b}$ for $\mathbf{y}$. Since L is lower triangular, we have

$$y_1 = \frac{b_1}{l_{11}},$$

and, for each $i = 2, 3, \ldots, n$,

$$y_i = \frac{1}{l_{ii}}\left[b_i - \sum_{j=1}^{i-1} l_{ij}y_j\right].$$

After $\mathbf{y}$ is found by this forward-substitution process, the upper-triangular system $U\mathbf{x} = \mathbf{y}$ is solved for $\mathbf{x}$ by backward substitution using the equations

$$x_n = \frac{y_n}{u_{nn}} \quad \text{and} \quad x_i = \frac{1}{u_{ii}}\left[y_i - \sum_{j=i+1}^{n} u_{ij}x_j\right].$$