

Práctica 3: Vectores matrices y programación estructurada

Joaquim Lloberas Valls

**Programación y Cálculo Matricial.
Curso Julio 2011**

**Métodos Numéricos para Cálculo y Diseño en
Ingeniería**

Índice

Resumen del trabajo.....	3
---------------------------------	----------

Ejercicio 1

Planteamiento y resultados.....	3
Presentación de resultados.....	3
Conclusiones.....	5

Ejercicio 2

Planteamiento y resultados.....	6
Presentación de resultados.....	6
Conclusiones.....	8

Ejercicio 3

Planteamiento y resultados.....	9
Presentación de resultados.....	9
Conclusiones.....	12

Ejercicio 4

Planteamiento y resultados.....	13
Presentación de resultados.....	13
Conclusiones.....	13

Annexos

Ejercicio 1.....	14
Ejercicio 2.....	15
Ejercicio 3.....	16
Ejercicio 4.....	16

Resumen:

En este trabajo se pretende abordar la programación y resolución de sistemas lineales de ecuaciones mediante FORTRAN.

Para ello, se analiza las distintas formas de almacenamiento de matrices, ya sea en filas o columnas tal como la eliminación Gaussiana para la resolución de sistemas lineales de ecuaciones.

También se estudian los tipos de salidas y entradas desde ficheros para leer y almacenar los resultados.

Práctica 3: Vectores matrices y programación estructurada

- 1) Realizar un programa en Fortran que calcule el producto de una matriz A por un vector b, según las implementaciones (filas y columnas). El programa estará formado por tres módulos que realizarán las siguientes funciones:
 - a. Módulo de lectura: Este módulo leerá el archivo de datos apartado1.dat con el formato indicado (ver nota(i)). Dicho archivo se puede obtener en la web de la asignatura.
 - b. Módulos de cálculo. Serán dos: el primer módulo realizará el producto de la matriz por el vector según una implementación por columnas. El vector resultado se guardará en un vector c diferente del utilizado para almacenar b.
 - c. Módulo de escritura: Este módulo escribirá los resultados en un archivo, con el formato indicado.

Planteamiento del problema:

Utilizaremos la definición pertinente de matrices en Fortran. Para dar más flexibilidad a su programación y hacer más general su desarrollo utilizaremos un parámetro (maxrank).

Mediante una secuencia de lectura leeremos la matriz A y el vector b.

Luego, escribiremos los vectores fila y columna a partir de la matriz leída. Llegamos luego a la multiplicación entre el vector y la matriz, para luego escribir los resultados obtenidos.

```
program apartat1

    INTEGER:: i, j, k, n, A1, A2, B1, B2
    parameter(maxrank = 10)
    dimension A(maxrank, maxrank)
    dimension b(maxrank)
    dimension col(maxrank*maxrank, 1)
    dimension fil(1, maxrank*maxrank)
    dimension c1(maxrank, 1)
    dimension c2(1, maxrank)
    open(UNIT=10, FILE='c:apartado1.dat', STATUS='OLD')
    read(10, *) n
    ! read A
    do i=1, n
        read(10, *) (A(i, j), j=1, n)
    end do
    ! read b
    do i=1, n
        read(10, *) b(i)
    end do
    close(10)
    k=0
    do i=1, 10
        do j=1, 10
            k=k+1
            fil(1, k) = A(i, j)
            col(k, 1) = A(j, i)
        end do
    end do
```

```

        enddo
    enddo
    do i=0,maxrank-1
        c1(i+1,1)=0
        c2(i+1,1)=0
        do j=0,maxrank-1
            c1(i+1,1)=c1(i+1,1)+fil(1,1+j+10*i)*b(i)
            c2(i+1,1)=c2(i+1,1)+col(i+1+10*j,1)*b(i)
        enddo
    enddo
    close(10)
    ! write A

open(UNIT=11,FILE='c:apartado1_resultados.dat',STATUS='UNKNOWN')
do i=1,n
    write(11,*) (A(i,j),j=1,n)
end do
! write b
write(11,*) (b(i),i=1,n)
!write c1,c2
write(11,*) (c1(i,1),i=1,n)
write(11,*) (c2(i,1),i=1,n)
close(11)
end

```

Presentación de resultados

Los resultados están incluidos en el archivo de texto apartado1_resultados.dat
Aquí presentamos solo los resultados obtenidos (c1,c2)

C1=65498853 2.98160005 0.706999898 4.96589994 1.18000007 -3.16079998
2.03009987 0.801900148 2.41359997 0.503499985

C2= 5498853 2.98160005 0.706999898 4.96589994 1.18000007 -3.16079998
2.03009987 0.801900148 2.41359997 0.503499985

Conclusiones

Este ejercicio analiza como trabajar matrices en filas y columnas y como trabajar con ellos.

- 2) Escribir un programa en Fortran que calcule el producto de dos matrices A y B dadas según las dos implementaciones expuestas. El programa estará constituido por tres módulos que realizarán las siguientes funciones.
 - a. Módulo de lectura: Este módulo leerá el archivo de datos apartado2.dat con el formato indicado. Dicho archivo se puede obtener en la web de la asignatura.
 - b. Módulos de cálculo. Serán dos: el primer módulo realizará el producto de la matriz por la matriz según una implementación por filas y por columnas.
 - c. Módulo de escritura: Este módulo escribirá los resultados en un archivo, con el formato indicado.

Planteamiento del problema:

En este problema se plantea la multiplicación de dos matrices según una implementación por filas y una implementación por columnas.

Para ello se definen las matrices en filas y columnas y se hacen las operaciones pertinentes para obtener los resultados.

```
program multiplicacio_matrius
```

```
INTEGER:: i, j, k, n, A1, A2, B1, B2  
parameter(maxrank = 10)  
dimension A(maxrank, maxrank)  
dimension B(maxrank, maxrank)  
dimension col(maxrank*maxrank, 1)  
dimension filA(1, 20), colA(1, 20)  
dimension filB(1, 12), colB(1, 12)  
dimension c1(1, 15), c2(1, 15)
```

```
!Multiplicación por matrices almacenadas en filas y columnas
```

```
open(UNIT=12, FILE='c:apartado2.dat', STATUS='OLD')
```

```
! read A1 A2 B1 B2
```

```
read(12, *) A1, A2, B1, B2
```

```
! read A
```

```
do i=1, A1
```

```
read(12, *) (A(i, j), j=1, A2)
```

```
end do
```

```
! read B
```

```
do i=1, B1
```

```
read(12, *) (B(i, j), j=1, B2)
```

```
end do
```

```
close(12)
```

```
! write A
```

```
do i=1, A1
```

```
write(*, *) (A(i, j), j=1, A2)
```

```
end do
```

```
!write B
```

```
do i=1, B1
```

```
write(*, *) (B(i, j), j=1, B2)
```

```
end do
```

```
close(12)
```

```
k=0
```

```
do i=1, A1
```

```
do j=1, A2
```

```
k=k+1
```

```
filA(1, k) = A(i, j)
```

```
enddo
```

```
enddo
```

```
k=0
```

```
do i=1, A2
```

```
do j=1, A1
```

```
k=k+1
```

```
colA(1, k) = A(j, i)
```

```
enddo
```

```
enddo
```

```
k=0
```

```
do i=1, B1
```

```
do j=1, B2
```

```
k=k+1
```

```
filB(1, k) = B(i, j)
```

```
enddo
```

```
enddo
```

```
k=0
```

```
do i=1, B2
```

```
do j=1, B1
```

```
k=k+1
```

```
colB(1, k) = B(j, i)
```

```
enddo
```

```
enddo
```

```

write(*,*) (colA(1,k),k=1,20)
write(*,*) (colB(1,k),k=1,12)
do i=1,15
c1(1,i)=0
enddo
k=0
do i=1,4
c1(1,1)=c1(1,1)+filA(1,i)*filB(1,i+2*k)
c1(1,2)=c1(1,2)+filA(1,i)*filB(1,1+i+2*k)
c1(1,3)=c1(1,3)+filA(1,i)*filB(1,2+i+2*k)
c1(1,4)=c1(1,4)+filA(1,i+4)*filB(1,i+2*k)
c1(1,5)=c1(1,5)+filA(1,i+4)*filB(1,1+i+2*k)
c1(1,6)=c1(1,6)+filA(1,i+4)*filB(1,2+i+2*k)
c1(1,7)=c1(1,7)+filA(1,i+8)*filB(1,i+2*k)
c1(1,8)=c1(1,8)+filA(1,i+8)*filB(1,1+i+2*k)
c1(1,9)=c1(1,9)+filA(1,i+8)*filB(1,2+i+2*k)
c1(1,10)=c1(1,10)+filA(1,i+12)*filB(1,i+2*k)
c1(1,11)=c1(1,11)+filA(1,i+12)*filB(1,1+i+2*k)
c1(1,12)=c1(1,12)+filA(1,i+12)*filB(1,2+i+2*k)
c1(1,13)=c1(1,13)+filA(1,i+16)*filB(1,i+2*k)
c1(1,14)=c1(1,14)+filA(1,i+16)*filB(1,1+i+2*k)
c1(1,15)=c1(1,15)+filA(1,i+16)*filB(1,2+i+2*k)
k=k+1
enddo
do i=1,15
c2(1,i)=0
enddo
k=0
do i=0,3
c2(1,1)=c2(1,1)+colA(1,i*5+1)*colB(1,i+1)
c2(1,2)=c2(1,2)+colA(1,i*5+2)*colB(1,i+1)
c2(1,3)=c2(1,3)+colA(1,i*5+3)*colB(1,i+1)
c2(1,4)=c2(1,4)+colA(1,i*5+4)*colB(1,i+1)
c2(1,5)=c2(1,5)+colA(1,i*5+5)*colB(1,i+1)

c2(1,6)=c2(1,6)+colA(1,i*5+1)*colB(1,i+5)
c2(1,7)=c2(1,7)+colA(1,i*5+2)*colB(1,i+5)
c2(1,8)=c2(1,8)+colA(1,i*5+3)*colB(1,i+5)
c2(1,9)=c2(1,9)+colA(1,i*5+4)*colB(1,i+5)
c2(1,10)=c2(1,10)+colA(1,i*5+5)*colB(1,i+5)

c2(1,11)=c2(1,11)+colA(1,i*5+1)*colB(1,i+9)
c2(1,12)=c2(1,12)+colA(1,i*5+2)*colB(1,i+9)
c2(1,13)=c2(1,13)+colA(1,i*5+3)*colB(1,i+9)
c2(1,14)=c2(1,14)+colA(1,i*5+4)*colB(1,i+9)
c2(1,15)=c2(1,15)+colA(1,i*5+5)*colB(1,i+9)
k=k+1
enddo

do i=1,15
write(*,*) c1(1,i)
enddo
write(*,*) 'hola'
do i=1,15
write(*,*) c2(1,i)
enddo
open(UNIT=12,FILE='c:apartado2_resultados.dat',STATUS='OLD')
do i=1,15
write(*,*) c1(1,i)
enddo

```

```

write(*,*) ' _____ '
do i=1,15
write(*,*) c2(1,i)
enddo
close(12)
end

```

Presentación de resultados

Los resultados obtenidos dados en la multiplicación de las matrices es la siguiente:

C1=. -64. -66. -142. -152. -162. -222. -240. -258. -302. -328. -354. -382.
-416. -450.

Aun así están disponibles en el archivos de resultados apartado2_resultados.dat

Conclusiones

Se ha aprendido a trabajar con los índices de filas y columnas para poder realizar operaciones entre ellas, concretamente una multiplicación de dos matrices de rangos distintos.

- 3) Realizar un programa en Fortran que resuelva mediante el método de Gauss, un sistema lineal de ecuaciones del tipo $A \cdot x = b$. El programa estará formado por tres módulos:
 - a. Módulo de lectura: Este módulo leerá el archivo de datos apartado3.dat con el formato indicado. Dicho archivo se puede obtener en la web de la asignatura.
 - b. Módulos de cálculo. Resolverá el sistema lineal de ecuaciones, guardando el resultado en un vector x diferente del utilizado para b.
 - c. Módulo de escritura: Este módulo escribirá los resultados en un archivo, con el formato indicado.

Planteamiento del problema:

En este ejercicio se trata de resolver un sistema lineal. SE procederá siguiendo una eliminación Gaussiana hasta llegar a la solución directamente.

```

PROGRAM EX43

INTEGER:: I,J,K,N
parameter(maxrank = 10)
dimension A(maxrank,maxrank)
dimension B(maxrank)
dimension X(maxrank)
dimension INDX(maxrank)
open(UNIT=12,FILE='c:apartado3.dat',STATUS='OLD')
! read n
read(12,*) N

! read A
do I=1,N
read(12,*) (A(I,J),J=1,N)
end do
! read B
read(12,*) (B(J),J=1,N)

close(12)

C
CALL LEGS (A,N,B,X,INDX)

```

C

```
WRITE (6, 999) (X(I), I=1,N)

999 FORMAT (F16.8)

open(UNIT=14,FILE='c:apartado3_resultados.dat',STATUS='UNKNOWN')
do I=1,N
write(*,*) (A(I,J),J=1,N)
write(14,*) (A(I,J),J=1,N)
end do
! write c
write(*,*) '_____ '
write(14,*) '_____ '
write(*,*) (B(I),I=1,N)
write(14,*) (B(I),I=1,N)
close(14)
END

SUBROUTINE LEGS(A,N,B,X,INDX)

DIMENSION A(N,N),B(N),X(N),INDX(N)

CALL ELGS(A,N,INDX)

DO 100 I = 1, N-1
DO 90 J = I+1, N
B(INDX(J)) = B(INDX(J))
* -A(INDX(J),I)*B(INDX(I))
90 CONTINUE
100 CONTINUE

X(N) = B(INDX(N))/A(INDX(N),N)
DO 200 I = N-1, 1, -1
X(I) = B(INDX(I))
DO 190 J = I+1, N
X(I) = X(I)-A(INDX(I),J)*X(J)
190 CONTINUE
X(I) = X(I)/A(INDX(I),I)
200 CONTINUE

RETURN
END

SUBROUTINE ELGS(A,N,INDX)

DIMENSION A(N,N),INDX(N),C(N)

DO 50 I = 1, N
INDX(I) = I
50 CONTINUE

DO 100 I = 1, N
C1= 0.0
DO 90 J = 1, N
C1 = AMAX1(C1,ABS(A(I,J)))
90 CONTINUE
C(I) = C1
100 CONTINUE

DO 200 J = 1, N-1
```



```

    PI1 = 0.0
    DO 150 I = J, N
        PI = ABS(A(INDX(I),J))/C(INDX(I))
        IF (PI.GT.PI1) THEN
            PI1 = PI
            K = I
        ELSE
            ENDIF
150    CONTINUE

    ITMP = INDX(J)
    INDX(J) = INDX(K)
    INDX(K) = ITMP
    DO 170 I = J+1, N
        PJ = A(INDX(I),J)/A(INDX(J),J)

        A(INDX(I),J) = PJ

        DO 160 K = J+1, N
            A(INDX(I),K) = A(INDX(I),K) - PJ*A(INDX(J),K)
160    CONTINUE
170    CONTINUE
200 CONTINUE

    RETURN
    END

```

Presentación de resultados

El vector solución es

```

1.00000083 1.00000095 0.99999994 0.999999523 0.999999762 1. 0.999999285
1.00000048 1.00000024 1.00000107

```

El archivo de resultados se adjunta como aparatdo3_resultados.dat

Conclusiones

Este algoritmo contiene algunos métodos para la eliminación Gaussiana hasta una matriz diagonal. Se han utilizado distintos formatos.