

Informática

Clase 4: Arrays

Mario Merino Martínez
mario.merino@upm.es

Escuela de Ingeniería Aeronáutica y del Espacio
Universidad Politécnica de Madrid

18 de octubre de 2011

Índice

- 1 Declaración de Arrays
- 2 Uso de Arrays y sus elementos

¿Qué es un Array?

- Los **arrays** son **secuencias ordenadas de datos**. Una variable array puede **almacenar** un **vector** o una **matriz**
- Cada **elemento del array** tiene unas **coordenadas** o **índices** para acceder a él:

Ejemplo

En un vector v , los elementos se denotan $v(1)$, $v(2)$, etc.

En una matriz A , los elementos tienen dos índices: $a(1, 3)$...

- Todos **los elementos de un array** son **del mismo tipo** (**real**, **integer**...)
- A las **variables simples** (sin varias componentes) se las suele llamar **escalares**

Declaración de Arrays

Hay que **especificar las dimensiones** de nuestros arrays **al declararlos**:

```
real*8 :: v(3), A(5,4) ! Vector de 3 elementos y matriz 5x4
```

También se puede usar el atributo **dimension**:

```
real*8, dimension(2,3) :: A, B ! Matrices 2x3
```

- Para las **matrices**, **el primer índice denota filas, el segundo columnas**. Ej., la matriz B es:

$$B = \begin{bmatrix} b(1,1) & b(1,2) & b(1,3) \\ b(2,1) & b(2,2) & b(2,3) \end{bmatrix}$$

Declaración de Arrays II

- Los **índices comienzan a contar en 1** si no se especifica lo contrario. Así, el vector declarado `real*8 :: v(3)` tiene los elementos $v(1)$, $v(2)$ y $v(3)$.
- Es posible **definir arrays que comiencen en cualquier número** con la **notación ":"**

```
real*8 :: A(0:2,-2:2) ! Una matriz 3x5 con índices  
                        !empezando en 0 y -2 resp.
```

- Por último, se pueden usar **enteros con atributo parameter** para **establecer las dimensiones**:

```
integer, parameter :: n = 3  
real*8 :: A(n,0:2*n)
```

Constructor de Vectores

La **notación** "`(/ ... /)`" (**constructor de vectores**) permite **crear un vector fila** para usarlo en **expresiones** o **guardarlo en una variable adecuada**.

```
real*8 :: v(3)
! ...
v = (/ 1.2d0, 2.3d0, 5.5d0 /)
```

Todos los elementos del vector han de ser **del mismo tipo**: **real**, **integer**... (si no, error)

Rodajas de Array: *Slicing*

Es **muy útil** **seleccionar** **filas** o **columnas** enteras de una matriz con la **notación** “ $n_i:n_f:p$ ” para **usarlas en expresiones**:

Ej., para la matriz: $A = \begin{bmatrix} a_{1,1} & a_{1,2} & a_{1,3} & a_{1,4} \\ a_{2,1} & a_{2,2} & a_{2,3} & a_{2,4} \\ a_{3,1} & a_{3,2} & a_{3,3} & a_{3,4} \end{bmatrix}$

- “:” selecciona **todos los elementos de esa dimensión**:
 $A(2, :) \rightarrow (a_{2,1}, a_{2,2}, a_{2,3}, a_{2,4})$
- “ $n_i:n_f$ ” indica **principio y fin**: $A(2, 2:3) \rightarrow (a_{2,2}, a_{2,3})$
- “ $n_i:n_f:p$ ” añade el **paso**: $A(2, 2:4:2) \rightarrow (a_{2,2}, a_{2,4})$

Esto puede hacerse **simultáneamente en varias dimensiones** [Ej. $A(:, 2:4)$]. Si n_i o n_f **se omiten**, se **sobreentiende el principio o el final** de esa dimensión

Asignación de valores

La **asignación de valores** a un array puede hacerse de **varias maneras**:

- **Elemento a elemento**, especificando los **índices adecuados**:

$A(2,3) = 1.088E-2$

$v(i) = 3.55$! *Siendo i una variable entera*

- **Fila a fila con *slicing*** (especialmente **útil para inicializar matrices**. También podría hacerse por columnas, etc.):

$A(1,:) = (/ 1.23, 2.41, 9.88 /)$

$A(2,:) = (/ 32.1, 1.55, 1.07 /)$

Asignación de valores II

- Con la función **reshape** (Muy poco visual, **más compleja**. Hay que dar la **matriz entera por columnas**, y un **vector entero** para las **dimensiones**. A veces es **necesario continuar línea con &)**:

```
A = reshape( (/ 1.,2.,3.,4.,5.,6. /), (/2,3/) )
```

$$\Rightarrow A = \begin{bmatrix} 1. & 3. & 5. \\ 2. & 4. & 6. \end{bmatrix}$$

Funciones para Arrays

Fortran tiene **multitud de funciones para el cálculo matricial**, etc., que iremos conociendo. Por ejemplo, existen las funciones para encontrar el **máximo y mínimo elemento de un array**:

- **maxval**(A), **minval**(A): Encuentran el **máximo/mínimo valor** de un array (vector o matriz) A
- **maxloc**(A), **minloc**(A): Devuelven las **coordenadas (localización)** del máximo/mínimo de A. **El resultado es un vector de enteros.**

*Imprescindibles

- Los **arrays de fortran** son nuestros **vectores, matrices, y tablas** con más dimensiones
- En la **declaración** tenemos que **indicar las dimensiones**. Puede hacerse con un **parameter**
- **Usar arrays bien es saber hacer slicing bien**. Con **slicing “:”** y el **constructor (/.../)** podemos seleccionar **filas/columnas/submatrices**, y **dar valores de forma rápida y visual**.
- Hay **infinidad de funciones para arrays** (podéis buscar en la ayuda de Plato)
- Para los curiosos: es posible no fijar las dimensiones de un array hasta más tarde, con variables **allocatable**.