

Cap 10. Tablouri

- *Tablouri și elemente de tablou*
- *Declararea dimensiunilor și a tipului*

Tablouri și elemente de tablou

Adesea, în practica programării, întâlnim problema prelucrării unei mulțimi ordonate de date care se întind pe mai multe dimensiuni pentru a reprezenta coloane, linii, plane, etc.

În limbajul Fortran o mulțime ordonată de date scalare cu 1-7 dimensiuni, toate de același tip și cu aceeași parametri de tip (kind și lungime) se poate organiza ca o entitate numită *tablou* (array). Tabloul se identifică printr-un *nume* iar oricare din elementele scalare ale tabloului se numește *element de tablou*.

Definiție

element de tablou este *nume* (*listă_de_indici*)

listă_de_indici este *indice* [, *indice*]

Tablouri și elemente de tablou

Aici *nume* este numele tabloului iar *indice* este o expresie scalară întreagă. Prin urmare, pentru a referi în program un element de tablou, se scrie numele tabloului urmat de o listă *de indici* (subscript list) închisă între paranteze. Pentru fiecare dimensiune există un indice a cărui valoare indică poziția în acea dimensiune. Numărul de elemente într-o dimensiune se numește *întinderea* (extent) în acea dimensiune. *Rangul tabloului* (array rank) este numărul de dimensiuni ale tabloului. *Forma unui tablou* (array shape) este determinată de rangul său și de întinderile sale. Pentru a specifica forma unui tablou de rang n folosim notația (d_1, d_2, \dots, d_n) , unde d_i este întinderea în dimensiunea i .

Numărul total de elemente ale tabloului, egal cu produsul întinderilor, se numește *mărimea tabloului* (array size).

Tablouri și elemente de tablou

Exemplul 1. Vectorul viteză \vec{v} ce are componentele $\vec{v} = (2, -3, 5)$ se poate organiza ca un tablou real unidimensional cu forma (3), tablou ce are numele viteza. Cele 3 elemente de tablou se notează cu viteza(1), viteza(2) și viteza(3) și au valorile 2, -3 și 5.

Exemplul 2. Matricea dreptunghiulară a cu 3 linii și 4 coloane

$$a = \begin{pmatrix} 3 & 2 & 5 & -4 \\ -1 & 7 & 25 & 8 \\ 4 & 9 & -6 & 1 \end{pmatrix}$$

se poate organiza ca un tablou real bidimensional, cu forma (4,3) și mărimea $4 \times 3 = 12$. Elementele tabloului sunt $a(1, 1)$, $a(2,1)$, $a(3,1)$, $a(4,1)$, $a(1,2)$, $a(2,2)$, $a(3,2)$, $a(4,2)$, $a(1,3)$, $a(2,3)$, $a(3,3)$, $a(4,3)$, $a(1, 4)$, $a(2,4)$, $a(3,4)$, $a(4,4)$. Spre deosebire de notația matricială, aici primul indice numără coloanele iar al doilea numără liniile.

Tablouri și elemente de tablou

Exemplul 3. Lista cu numele studenților dintr-o grupă:

Adam

Barbu

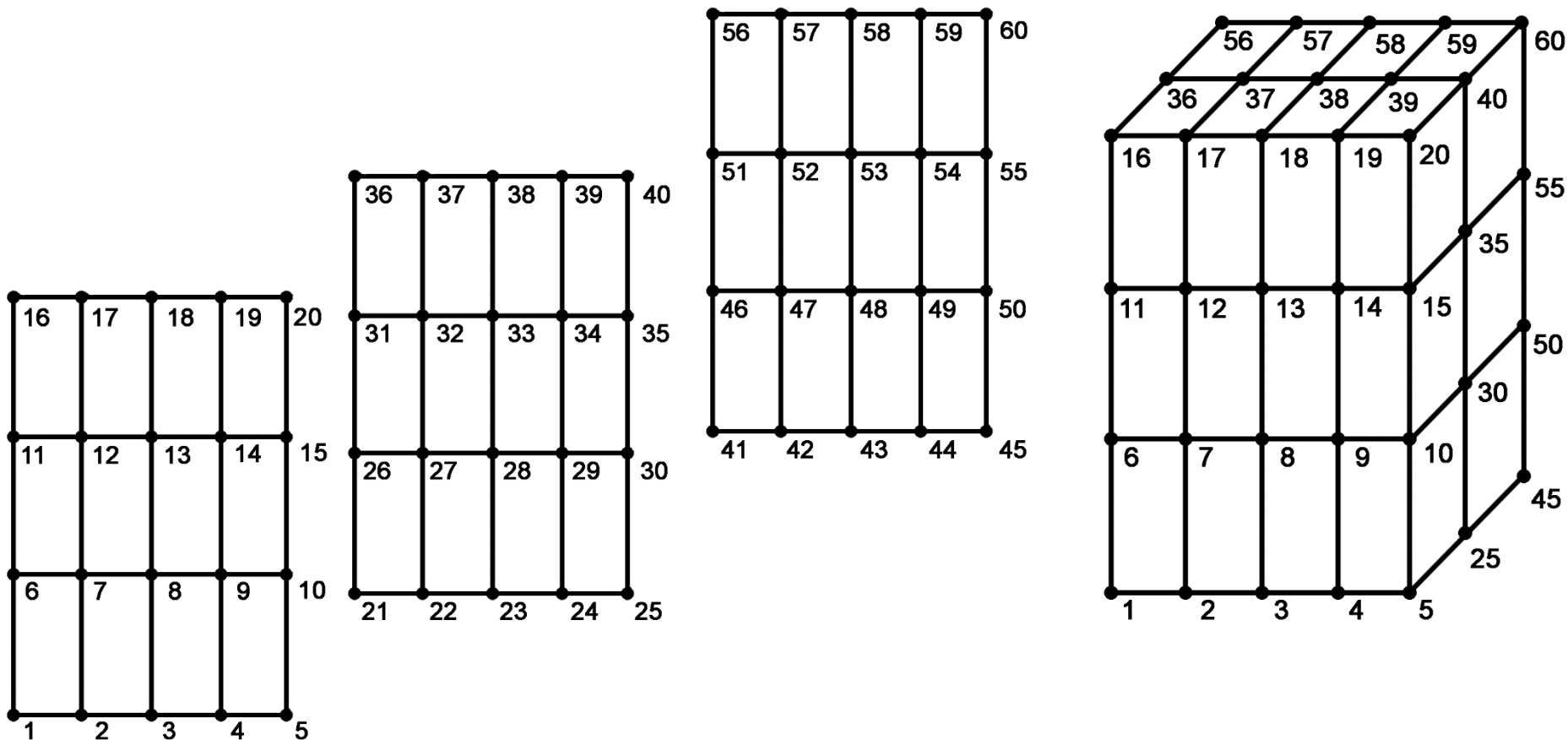
Constantin

Diaconu

se poate organiza ca un tablou unidimensional, de tip caracter, de mărime 4 și formă (4), tablou numit grupă. Elementele tabloului sunt grupa(1), grupa(2), grupa(3), grupa(4) toate având lungimea egală cu 10.

Tablouri și elemente de tablou

Exemplul 4. Un corp omogen de forma unui paralelipiped dreptunghic este discretizat în elemente finite cu ajutorul unei grile 3D, prezentate mai jos.



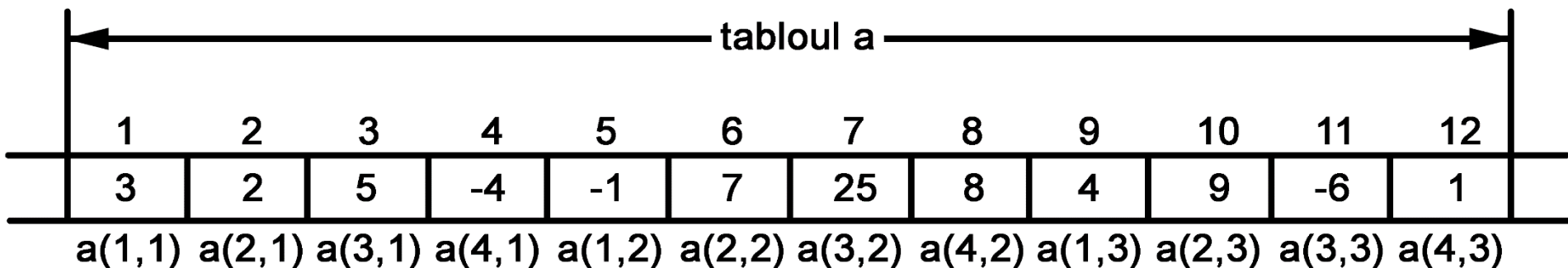
Tablouri și elemente de tablou

Mulțimea nodurilor grilei, numerotate de la 1 la 60, mulțime ce alcătuiește un patern ordonat cu linii, coloane și planuri se poate organiza ca un tablou tridimensional numit nod. Elementele tabloului se notează $\text{nod}(1,1,1)$, $\text{nod}(2,1,1)$, $\text{nod}(3,1,1)$, ... , $\text{nod}(5,4,3)$. Aici primul indice numără coloanele, al doilea numără liniile și al treilea numără planele. De exemplu, $\text{nod}(2,3,1)=12$, $\text{nod}(4,3,2)=34$. Tabloul nod are forma (5,4,3).

În memorie, elementele unui tablou unidimensional, se stochează unul după altul iar elementele tablourilor multidimensionale sunt stocate tot ca un șir liniar. Un tablou se stochează în memorie astfel ca cel mai din stânga indice să varieze cel mai rapid, iar cel mai din dreapta să varieze cel mai lent.

Tablouri și elemente de tablou

Exemplul În figura de mai jos arătăm modul de stocare în memorie a tabloului a, care este un tablou 2D cu 12 elemente.



Declaraarea tipului

Într-un program Fortran entitățile folosite (variabile, tablouri, proceduri) posedă diferite *atribute*. Atributele determină modul în care entitatea va fi folosită în program.

Atributele pot fi specificate, fie cu o instrucțiune de declarare a tipului, fie cu instrucțiuni de declarare a unui anumit atribut (instrucțiunile DIMENSION, ALLOCATABLE, TARGET, DATA, PARAMETER, PUBLIC, PRIVATE, INTENT, OPTIONAL, SAVE, EXTERNAL, INTRINSIC). Asta înseamnă că în limbajul Fortran, specificațiile atributelor se pot face în două moduri: i) *orientate către entitate și orientate către atribut*.

Exemplu. În instrucțiunea de declarare a tipului

```
REAL, PARAMETER :: A = 2.
```

am colectat într-o singură instrucțiune toate atributele entității A. În acest exemplu avem de-a face cu o specificație orientată către entitate.

Declararea tipului

Putem face specificații orientate către atribut dacă în locul instrucțiunii anterioare scriem declarațiile:

REAL A

PARAMETER (A = 2.)

Declaraarea dimensiunilor

În limbajul Fortran orice tablou posedă *atributul dimensiune*. Declaraarea dimensiunilor unui tablou este obligatorie și se face cu ajutorul unor instrucțiuni ce conține o specificație *de tablou*. Există mai multe metode pentru a declara un tablou însă vom utiliza doar *declaraarea tablourilor cu dimensiuni fixe*.

Șabloanele recomandate sunt:

```
type, DIMENSION (spec_de_tablou) :: nume_de_tablou
```

```
type nume_de_tablou (spec_de_tablou)
```

Primul șablon îl folosim atunci când dorim să declarăm că mai multe tablouri au același tip și aceleași dimensiuni. Al doilea șablon îl folosim atunci când dorim să declarăm că tablouri diferite, de dimensiuni diferite, au toate același tip.

Declaraarea dimensiunilor

Instrucțiunea DIMENSION are forma:

```
DIMENSION [::] nume_de_tablou (spec_de_tablou)
```

Exemplu. Tabloul întreg 1D numit indice are 25 elemente. Pentru a-i declara dimensiunile putem folosi o declarare orientată către entitate:

```
INTEGER, DIMENSION (25) :: INDICE
```

sau putem folosi declararea orientată către atribut:

```
INTEGER INDICE
```

```
DIMENSION INDICE (10)
```

Declararea dimensiunilor

Exemplu. Tablourile 2D, cu forma (3,4), **alfa**, **beta**, **gama**, **delta**, **epsilon** au fiecare 12 elemente. Declararea dimensiunilor o putem face cu instrucțiunea:

```
REAL, DIMENSION (3,4) :: alfa, beta, gama, delta, epsilon
```

sau cu instrucțiunea:

```
REAL alfa(3,4), beta(3,4), gama(3,4), delta(3,4), epsilon(3,4)
```

sau cu grupul de instrucțiuni:

```
REAL alfa, beta, gama, delta, epsilon
```

```
DIMENSION alfa(3,4), beta(3,4), gama(3,4), delta(3,4),  
epsilon(3,4)
```

Declaraarea dimensiunilor

Exemplu. Instrucțiunile

```
REAL a, b, c, d
```

```
DIMENSION a(3), b(-1 : 4), c(2,4), d(2,3,2)
```

declară că tabloul real cu numele a este unidimensional și are trei elemente ce se stochează în memorie în ordinea a(1), a(2), a(3); tabloul real numit b este unidimensional și are șase elemente ce se stochează în ordinea b(-1), b(0), b(1), b(2), b(3), b(4); tabloul real numit c este bidimensional și că are $2 \times 4 = 8$ elemente stocate în ordinea c(1,1), c(2,1), c(1,2), c(2,2), c(1,3), c(2,3), c(1,4), c(2,4) și tabloul real numit d are $2 \times 3 \times 2 = 12$ elemente stocate în ordinea d(1,1,1), d(2,1,1), d(1,2,1), d(2,2,1), d(1,3,1), d(2,3,1), d(1,1,2), d(2,1,2), d(1,2,2), d(2,2,2), d(1,3,2), d(2,3,2).

Declaraarea dimensiunilor

Exemplu. Instrucțiunile

```
INTEGER, DIMENSION(4):: int
```

```
REAL alfa (2,3), b(0:5)
```

```
LOGICAL, DIMENSION(3) :: gama
```

```
CHARACTER*20 nume (10)
```

declară că tabloul int de tip INTEGER are patru elemente; tabloul alfa de tip REAL este bidimensional și are $2 \times 3 = 6$ elemente iar b, tablou bidimensional, este de tip REAL și are 6 elemente; tabloul unidimensional gama are 3 elemente și este de tip LOGICAL, iar tabloul nume cu 10 elemente este de tip CHARACTER*20.

Utilizarea tablourilor

Exemplu. Pentru a citi element cu element tabloul real a cu 4 elemente putem folosi instrucțiunile:

```
READ*, a(1)
```

```
READ*, a(2)
```

```
READ*, a(3)
```

```
READ*, a(4)
```

Dacă tabloul a ar avea mai multe elemente, de exemplu 100, acest procedeu de citire este ineficient. Pentru citirea element cu element a tabloului putem folosi o secvență ce conține o iterație:

```
READ*,n
```

```
DO i=1,n
```

```
    READ*, a(i)
```

```
END DO
```


Utilizarea tablourilor

Aici n este numărul de elemente ale tabloului. Observăm că în corpul buclei DO figurează $a(i)$ unde indicele i este o variabilă întreagă.

Pentru scrierea element cu element a tabloului a putem folosi secvența:

```
DO i=1,n
    PRINT*, a(i)
END DO
```

Utilizarea tablourilor

Exemplu. Citirea unei matrici pe linii, element cu element

```
INTEGER i, j, n
PARAMETER (n = 5)
REAL a(n, n)
DO i=1,n
    DO j=1,n
        READ*, a(i, j)
    END DO
END DO
```

imbricate).

Exemplu. Citirea unei matrici pe coloane, element cu element

```
INTEGER i, j, n
PARAMETER (n= 5)
REAL a(n, n)
DO j=1,n
    DO i=1,n
        READ*, a( j, i)
    END DO
END DO
```

în aceste exemplele avem de-a face cu bucle incluse unele în altele (bucle imbricate – suprapuse ca țiglele).

Utilizarea tablourilor

Exemplu. Programul următor citește două matrici, află produsul lor și scrie rezultatul.

```
PROGRAM produs_matrici
IMPLICIT NONE
INTEGER i, j, k, n
PARAMETER (n = 3)
REAL a(n,n), b(n,n), c(n,n)
! Citirea pe linii a matricii a (element cu
element)
DO i=1,n
    DO j=1,n
        READ*, a(i,j)
    END DO
END DO
! Citirea pe linii a matricii b (element cu
element)
DO i=1,n
    DO j=1,n
        READ*, b(i,j)
    END DO
```

```
END DO
! Calculul matricii produs  $c = a * b$ 
DO i=1,n
    DO j=1,n
        c(i,j) = 0.
        DO k=1,n
            c(i,j) = c(i,j) + a(i,k) * b(k,j)
        END DO
    END DO
END DO
! Tipirea matricii c
J= 1
DO i=1,n
    PRINT*, c(i,j), c(i,j+1), c(i,j+2)
END DO
END PROGRAM produs_matrici
```

Bibliografie

- *Octavian PETRUȘ, Fortran 90/95, Limbaj și Tehnici de programare, Editura Universității Tehnice “Gheorghe Asachi” din Iași, 2001*
- Romeo CHELARIU, Sisteme de operare și limbaje de programare (Îndrumar de laborator), <http://www.sim.tuiasi.ro/wp-content/uploads/Chelariu-indrumar-solp.pdf>, 2004
- <https://ro.wikipedia.org>