Faculté de science et technologie,
Université Djilali Bounaama, khemis Miliana, DBKM

Docteur : S. KHERROUBI

Notion d'unité de programme

I Introduction:

Un programme source Fortran est compose de parties indépendantes appelées unites de programme (**scoping unit**). Chaque partie est compilée de façon indépendante. Chacune admet son propre environnement. Il sera cependant possible que ces parties communiquent entre elles.

Les différentes unit´es de programme sont :

- ☞ Le programme principal,
- ☞ Les sous-programmes :
  - ➳ De type subroutine,
  - ➳ De type function,
- ☞ Les modules,
- ☞ les blocks data.

Chaque unité comprend une partie d´déclarative (déclaration des variables locales, ...) suivie d'une partie comportant des instructions exécutables ; l'instruction STOP interrompt le programme.

Des qu'on écrit des programmes de taille un peu importante, on a envie de d´écoper les programmes en différentes unités logiques. Ce choix rend les programmes plus lisibles et permet par ailleurs de repartir le développement d'un projet entre plusieurs d´enveloppeurs, dont chacun est responsable du d´développement d'une unit´e. Par ailleurs, un morceau de programme pourra être utilisé plusieurs fois `a l'intérieur d'un programme donne, ou aussi être mutualise dans différents programmes. Les sous-programmes seront de deux types : procédures (subroutine) ou fonctions (function).

Fonctions. Une fonction fortran correspond exactement à l'objet mathématique que nous connaissons bien. Elle est définie par le mot clé **function**. Ainsi la fonction

f: R 3 $\longrightarrow$ R , (a, b, c)

$7\longrightarrow a + bc$

peut être programmée comme suit : function f(a,b,c) ! on met ici toutes les variables

d'entr´ees de la fonction implicit none real :: a,b,c,f ! On définit les variables d'entrées ET ! la variable de sortie ! on met ici toutes les instructions nécessaires au calcul de f(a,b,c) f= a+b*c

! On termine par affecter la valeur de sortie. end function f La fonction *f* pourra alors être appelée dans le programme par une instruction x=f(a,b,c) .

Type function nom_de_fonction (<arg1>,..,<argn>) result(res)
implicit none
déclaration des arguments <arg1>,..,<argn>
// Déclaration des variables
// Suite des instructions
end function nom_de_fonction

Exemple :

```
PROGRAM Produit
INTEGER :: n, m, t
WRITE(*,*) "Entrez la valeur de n :"
READ(*,*) n
WRITE(*,*) "Entrez la valeur de m :"
READ(*,*) m
WRITE(*,*) "Entrez la valeur de t :"
READ(*,*) t
WRITE(*,*) "f = ", Fact(n) * Fact(m) * Fact(t)
CONTAINS
FUNCTION Fact(x)
INTEGER :: x, i, result
result = 1
DO i = 2, x
result = result * i
ENDDO
Fact = result
END FUNCTION Fact
END PROGRAM Produit
```