

UNIVERSITE PIERRE ET MARIE CURIE

Licence de Chimie UE-209

INTRODUCTION A LA PROGRAMMATION FORTRAN

Plan :

1 : Introduction

2 : Structure d'un programme Fortran

3 : Aiguillages

4 : Processus itératifs

5 : Variables indicées

6 : Sous-programmes

Contrôle des connaissances :

CC N°1 (10 min - 5pts) pendant le deuxième cours

CC N°2 (10 min - 5pts) pendant le troisième cours

Examen (1h - 30pts) pendant le quatrième cours

Objectifs

L'objectif de cette UE est d'approfondir les connaissances des étudiants dans les objets et concepts mathématiques de base utilisés dans les différents domaines de la chimie. L'enseignement est donc axé sur les besoins des différents enseignements dispensés en L2 et L3 de la mention chimie, dont seront tirées les principales applications. Les concepts sont illustrés et complétés par des enseignements d'informatique qui vont familiariser les étudiants avec des notions de Linux et de Fortran. Ces notions d'informatique permettront ensuite aux étudiants lors de leurs études (L3 et Master) d'aborder l'informatique appliquée aux sciences chimiques.

Conventions utilisées

Le texte écrit `comme ceci` est un message affiché sur l'écran ou un texte à taper au clavier

Le texte écrit «*comme ceci*» est un texte que vous devez adapter selon le contexte.

la touche ENTER (entrée) est indiquée par le signe ↵

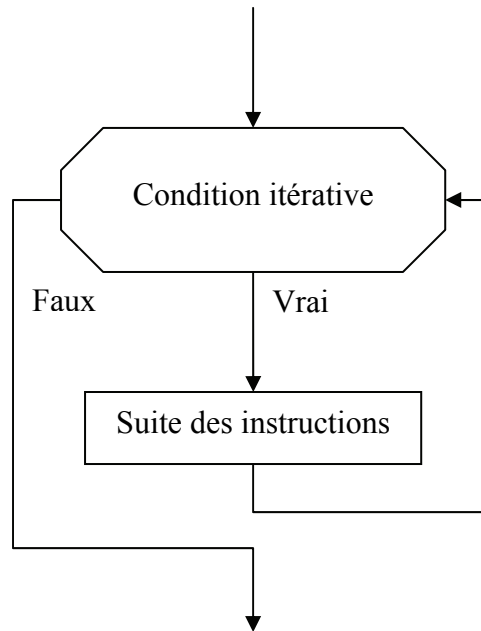
le caractère espace est indiqué par le signe ^

- COURS 4 -

PROCESSUS ITERATIFS

IV.1) Les séquences itératives.

Un programme itératif permet de répéter des opérations plusieurs fois. Une séquence itérative est une suite d'instructions que l'on exécute plusieurs fois à l'intérieur d'un même programme.



IV.2) Les boucles arithmétiques.

La lecture du programme est facilitée si vous décalez les écritures des instructions vers la droite (colonne supérieure à 7)

```
DO^«condition itérative»
```

```
... suite des instructions ...
```

```
ENDDO
```

La condition itérative porte sur une variable (entière, réelle ou double précision) dont on donne la valeur de départ, la valeur de fin et, le cas échéant, la valeur du pas de l'incrément :

```
«nom de la variable» = «départ», «fin», «pas»
```

PERMIS

Les valeurs de l'instruction DO (valeur initiale, valeur finale, incrément) peuvent être des expressions arithmétiques comportant des variables. On a le droit de réutiliser ces variables à l'intérieur de la boucle (les valeurs initiales, finales et l'incrément sont calculés dès l'entrée de la boucle et sont conservés tels quels).

INTERDIT

On n'a pas le droit de rentrer dans une boucle sans passer par l'instruction DO.

Il est interdit de modifier la valeur du compteur de boucle lors de l'exécution de cette boucle.

INITIALISATION

Pour cumuler des valeurs, il ne faut pas oublier d'initialiser les variables. Il n'est jamais garanti que les valeurs non déclarées soient zéro.

L'initialisation doit bien sûr se faire en dehors de la boucle.

Par exemple :

```

^ ^ ^ ^ ^ Prod=1
^ ^ ^ ^ ^ DO X=1, 5, 1
^ ^ ^ ^ ^ ^ Prod=Prod*X
^ ^ ^ ^ ^ ENDDO
^ ^ ^ ^ ^ WRITE(*,*) 'LA VALEUR DU PRODUIT VAUT ',Prod

```

IV.2.a) Boucle arithmétique d'écriture implicite:

Au lieu de

```

^ ^ ^ ^ ^ DO X=1, 20, 1
^ ^ ^ ^ ^ ^ WRITE(*,*) 'Le carré de ',X,' = ',X**2
^ ^ ^ ^ ^ ENDDO

```

Les résultats du write seront affichés sur 20 lignes

On peut écrire:

```

^ ^ ^ ^ ^ WRITE(*,*) (('Le carré de ',X,' = ',X**2),X=1.,20.,1.)

```

Les résultats du write seront affichés sur une seule ligne. A utiliser pour l'affichage de matrices.

IV.2.b) Boucles arithmétiques multiples.

Il est permis d'imbriquer des boucles pourvu qu'elles aient des compteurs différents. Ces boucles ne doivent cependant pas se chevaucher; cela correspondrait à rentrer dans une boucle sans passer par l'instruction DO.

Ce sont les boucles internes qui sont achevées en premier.

On peut écrire:

```

^ ^ ^ ^ ^ DO ^ I = 1, 5
^ ^ ^ ^ ^ DO ^ J = 1, 3
^ ^ ^ ^ ^ WRITE (*, *) I, J
^ ^ ^ ^ ^ ENDDO
^ ^ ^ ^ ^ ENDDO

```

ou encore

```

^ ^ ^ ^ ^ WRITE (*, *) ((I, J), J = 1, 3), I = 1, 5)

```

On obtient 1, 1, 1, 2, 1, 3, 2, 1, 2, 2, 2, 3, 3, 1, 3, 2, 3, 3 sur une ligne dans le dernier cas, par couples sur des lignes successives dans les cas précédents. Que donnerait l'ordre d'impression pour la programmation suivante ?

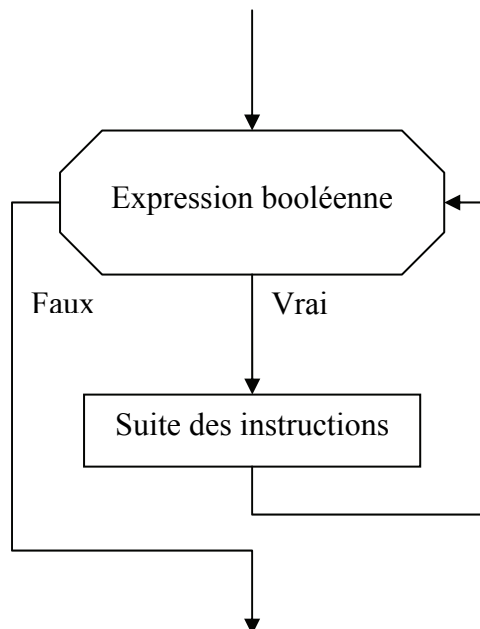
```

^ ^ ^ ^ ^ DO ^ I = 1, 5
^ ^ ^ ^ ^ WRITE (*, *) ((I, J, ' ', ' '), J = 1, 3)
^ ^ ^ ^ ^ ENDDO

```

IV.3) Les séquences booléennes itératives.

Tant que l'expression booléenne est vérifiée, le programme est dans un processus itératif.



La lecture du programme est facilitée si vous décalez les écritures des instructions vers la droite (colonne supérieure à 7)

DO^WHILE («*expression booléenne*»)

... suite des instructions ...

ENDDO

Le bloc d'instruction compris entre DO WHILE () et ENDDO n'est effectué que si la condition entre parenthèses est remplie. Si les variables entrant dans la condition sont modifiées, les instructions sont recommencées jusqu'à ce que la condition devienne fausse.

Par exemple :

```
^ ^ ^ ^ ^ I=1
^ ^ ^ ^ ^ N=10
^ ^ ^ ^ ^ DO^WHILE (I.LE.N)
^ ^ ^ ^ ^ ^ ^ ^ WRITE(*,*) I
^ ^ ^ ^ ^ ^ ^ ^ I=I+1
^ ^ ^ ^ ^ ENDDO
```

Cette écriture remplace une boucle I=1,10

```
^ ^ ^ ^ ^ I=1
^ ^ ^ ^ ^ N=10
^ ^ ^ ^ ^ DO^WHILE (I.LE.N)
^ ^ ^ ^ ^ ^ ^ ^ WRITE(*,*) I
^ ^ ^ ^ ^ ^ ^ ^ N=N-1
^ ^ ^ ^ ^ ENDDO
```

Cette écriture remplace une boucle N=10,1

```
^ ^ ^ ^ ^ READ(*,*) A
^ ^ ^ ^ ^ DO^WHILE ( (A.GE.-1.) .AND. (A.LE.1.) )
^ ^ ^ ^ ^ ^ ^ ^ X=ACOS(A)
^ ^ ^ ^ ^ ^ ^ ^ WRITE(*,*) ' L''ANGLE VAUT ',X
^ ^ ^ ^ ^ ENDDO
```

Cette écriture remplace un IF () THEN...ENDIF

Exercices:

III-1) On jette trois dés à jouer. On cherche toutes les combinaisons possibles d'obtenir un total donné (compris entre 3 et 18). Ecrire ce programme en évitant les solutions redondantes (1+2+3 +2+1+3).

III-2) Ecrire le programme PREMIER décomposant un nombre en ses diviseurs, nombres premiers.

III-3) Calculer $y = \frac{N}{2} \left\{ (N-1)! \left(\frac{e}{N} \right)^N \right\}^2$

Vous devriez trouver $y = \pi$ si N est grand; Calculer y pour $N=120$. Quel est le plus grand nombre N que vous avez utilisé?

(Cet exercice consiste à retrouver la valeur de la constante dans la formule de Stirling

$N! = \sqrt{2\pi N} N^N e^{-N}$ ou $\ln N! = N \ln N - N + \frac{1}{2} \ln(2\pi N)$

Attention cette formule n'a de sens que pour N grand! Cherchez à faire ce calcul pour $N = 200$ et 5000 . Indiquez le plus grand nombre que vous avez essayé avec succès.