



Informatique 3

TP 1 : Présentation d'un environnement de programmation scientifique

1. Le But de la matière Informatique 3.

Cette matière a pour objectif d'apprendre à l'étudiant la programmation en utilisant des logiciels faciles d'accès (Essentiellement : Matlab, Scilab, Maple, Fortran ...). Cette matière sera un outil pour la réalisation des TP de méthodes numériques en S4.

Pour cela on a adopté le langage Fortran comme environnement de programmation numérique.

Pour cela on va présenter quelques notes qui constituent le document du cours intitulé "Fortran 90 et programmation numérique" dispensé aux étudiants de 2^{ème} Année St option hydraulique de l'université de batna2 (même pour les années 3 LMD et master1 et 2).

Ce cours ne prétend pas présenter aux étudiants l'ensemble des possibilités offertes par le Fortran 90, mais se concentre sur les aspects les plus intéressants pour la programmation numérique.

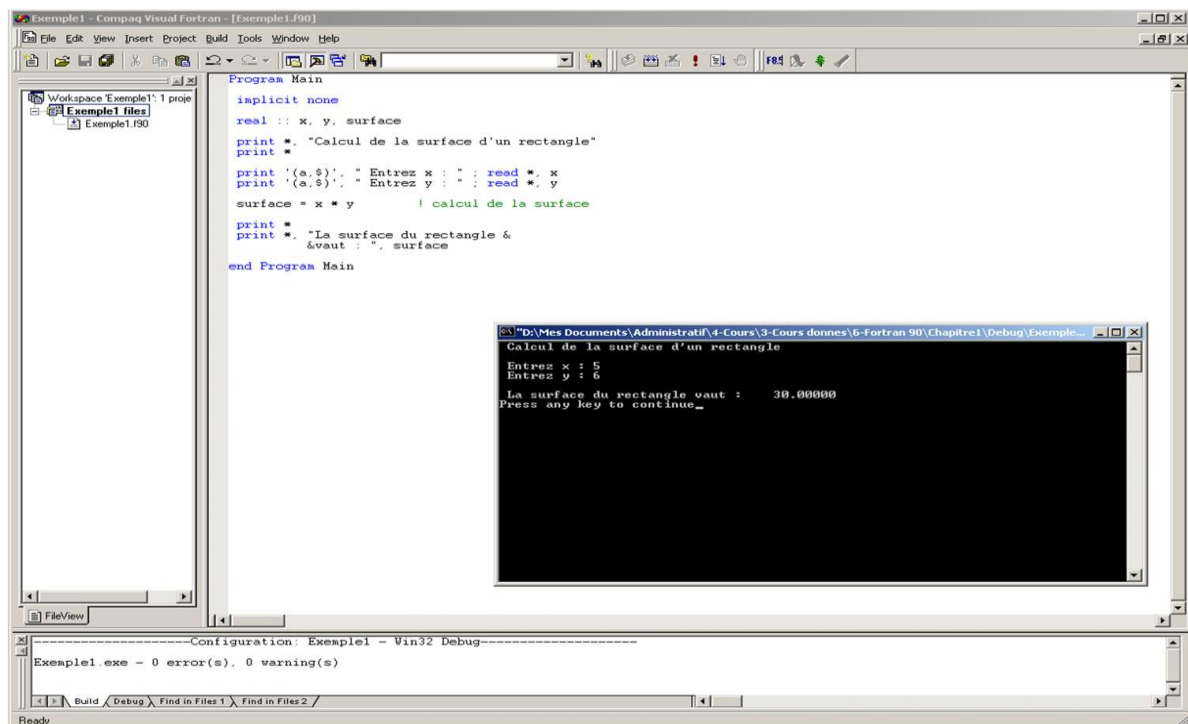
Les séances de Travaux Pratiques qui accompagnent ce cours permettent aux étudiants d'implémenter des algorithmes élémentaires d'analyse numérique en S4, tels que les résolutions de systèmes linéaires par des méthodes directes et itératives, la méthode de Newton-Raphson pour la résolution de problèmes non linéaires, les méthodes de recherche de valeurs propres, d'orthogonalisation, ...etc.

2. Présentation de l'environnement de programmation scientifique Fortran

Fortran (de l'anglais FORmula TRANslation), est né au milieu des années 1950. C'était le premier langage de niveau élevé.

- 1954 : John Backus, d'IBM, est à l'origine du projet de développement du premier langage symbolique.
- 1957 : Sortie des premiers compilateurs Fortran.
- 1958 : Sortie de Fortran II : apparition de la "compilation séparée".
Au début des années 1960, on s'occupe du développement de compilateurs pour les autres constructeurs de machines.
Il naît aussi une commission chargée de développer "une norme Fortran".
- 1966 : Sortie de Fortran IV (ou Fortran 66).
- 1977 : Sortie de Fortran V (ou Fortran 77).
- 1994 : Sortie des compilateurs Fortran 90.
- 1999 : Sortie des compilateurs Fortran 95.
- ????: Sortie des compilateurs Fortran 2000.
- ????: Sortie des compilateurs Fortran ????

En **Fortran 90**, les instructions sont écrites dans un fichier dont l'extension est "f90", (par exemple : toto.f90).



Exem

exemple de programme fortran 90

```

PROGRAM EXEMPLE1
!
  INTEGER :: A, B, C
!
  PRINT*, ' TAPER AU CLAVIER LA VALEUR DE A' READ*, A
  PRINT*, ' TAPER AU CLAVIER LA VALEUR DE B' READ*, B
  C=A+B
  PRINT*, ' LA VALEUR DE C=A+B EST' , C C=A-B
  PRINT*, ' LA VALEUR DE C=A-B EST' , C C=A*B
  PRINT*, ' LA VALEUR DE C=A*B EST' , C
!
END PROGRAM EXEMPLE1

```

Ce programme élémentaire permet déjà de comprendre les points suivants :

- Le programme commence par l'instruction : PROGRAM EXEMPLE1 et s'achève par END PROGRAM EXEMPLE1. Ceci est indispensable. Naturellement, on peut modifier le nom du programme (ici EXEMPLE1), mais la dernière instruction doit reprendre le même nom que la première.
- Juste après l'instruction PROGRAM EXEMPLE1, on "déclare" les différentes variables qui vont être utilisées dans le code.
 INTEGER :: A,B,C indique que l'on va utiliser trois variables, nommées A,B et C, qui seront toutes les trois de type "INTEGER", c'est à dire de type entier.
- Les instructions suivantes sont quasiment écrites en anglais.
 PRINT*, demande l'affichage à l'écran de données (*ici une chaîne de caractères et l'entier C un peu plus loin dans le code*),
 READ*, demande la lecture au clavier d'une donnée. La variable qui suit cette instruction prendra la valeur qui sera saisie au clavier lors de l'exécution.
- Suivent trois instructions correspondant à des calculs et affectations, ainsi que trois affichages à l'écran.

L'instruction $C=A+B$ demande à la machine d'évaluer la somme des variables entières A et B, et de stocker le résultat dans la variable C.

Ensuite, l'instruction $C=A-B$ permettra d'évaluer la différence A-B et modifiera la valeur de la variable C lui affecter la valeur du résultat de ce calcul.

De même pour l'instruction $C=A*B$...

Exemple d'exécution :

```
TAPER AU CLAVIER LA VALEUR DE A
8
TAPER AU CLAVIER LA VALEUR DE B
3
LA VALEUR DE C=A+B EST 11
LA VALEUR DE C=A-B EST 5 LA
VALEUR DE C=A*B EST 24
```

3. Les apports de la norme 90

- Le "format libre", identificateurs, déclarations, !, &, ;
- Précision des nombres. KIND, destiné à assurer une meilleure portabilité
- Objets de type dérivés
- DO ... END DO, SELECT CASE, WHERE
- Extensions sur les tableaux : profil, manipulation, fonctions prédéfinies
- Allocation dynamique de mémoire (ALLOCATABLE)
- Pointeurs
- Procédures et fonctions récursives
- Arguments de fonctions et de procédures : OPTIONAL, INTENT, PRESENT.

Passage par mot-clé

- Bloc interface, interface générique, surcharge d'opérateurs
- Les entrées / sorties
- De nouvelles fonctions intrinsèques

4. formats du Fortran

4.1 Les caractéristiques du format fixe Fortran 77 :

1. Une ligne de code a la structure suivante :
 - Colonnes 1 à 5 : zone étiquette. Ces colonnes peuvent contenir des espaces ou une étiquette (nombre entier compris entre 1 et 99999) permettant de se référer en cours de programme à l'instruction qui débute sur la même ligne.
 - Colonne 6 : colonne de continuation. Lorsqu'une instruction ne peut être placée sur une seule ligne, cette colonne contient un caractère différent de "0". Les caractères les plus couramment utilisés sont "+", "\$". Le nombre de lignes de continuation est limité à 19.
 - Colonnes 7 à 72 : zone instruction. A partir de la 73^{ème} colonne, les caractères sont ignorés.
2. Toute ligne commençant par l'un des caractères "C", "c" ou "*" est une ligne de commentaire.
3. Le caractère blanc n'est pas significatif.

4.2 Les Caractéristiques du format fixe Fortran 90 :

Il s'agit du format fixe Fortran 77 avec 2 extensions :

1. Plusieurs instructions peuvent être codées sur une même ligne en les séparant avec le caractère ";"
2. Le caractère "!" après une ou des instructions indique le début d'un commentaire. Toute ligne dont le premier caractère non blanc commence par le caractère "!" en une colonne différente de la 6ème est une ligne de commentaire.

Ce format est principalement destiné à assurer la compatibilité entre le Fortran 77 et le Fortran 90. Les fichiers au "format fixe" porteront l'un des suffixes .f (le plus courant), .for ou .FOR pour le compilateur fortran 90 DEC-Unix.

4.3 Les caractéristiques du format libre Fortran 90 :

1. Les colonnes 1 à 6 ne sont plus réservées. Les étiquettes peuvent être placées à partir de n'importe quelle colonne.
2. Plusieurs instructions peuvent être codées sur une même ligne en les séparant avec le caractère ";" La première instruction peut être placée à partir de n'importe quelle colonne.
3. Un commentaire peut être indiqué n'importe où sur une ligne à condition qu'il soit précédé du caractère "!".

Attention. C-----, c-----, *----- ne sont pas des commentaires Fortran 90 en "format libre" et génèrent des erreurs de compilation.

Exemple:

i=0! Initialisation

4. Les lignes peuvent être de longueur quelconque à concurrence de 132 caractères.
5. Les blancs sont significatifs. Les commandes ENDIF, ENDDO et GOTO peuvent également être écrites sous la forme END IF, END DO, GO TO (mais EN DIF est interdit alors que c'est autorisé en "format fixe").
6. Une ligne se terminant par le caractère "&" indique que la ligne suivante est une ligne de continuation. Si la ligne de continuation commence par le caractère "&", seuls les caractères situés juste après le "&" font partie de la ligne de continuation (pratique pour écrire des chaînes de caractères sur plusieurs lignes). Sans le caractère "&" au début de la ligne de continuation, tous les caractères, y compris les blancs et la tabulation, font partie de la ligne de continuation.

Exemple :

```
print *, 'Ceci est une phrase sur &  
&sur deux lignes'
```

Les fichiers au format libre porteront l'extension .f90.