

TP d'informatique
Première année apprentissage, mécanique des fluides - 2018-2019
Fonctions, Makefile, utilisations de modules et de librairie
4 h.

1 Introduction

Il s'agit d'écrire un programme en utilisant différentes méthodes pour gérer les procédures (sous-programme) et fonctions, en passant les paramètres comme argument ou en utilisant des variables globales ou en créant une librairie.

Tout ne sera pas accompli dans les 4 h prévues, il faudra donc apprendre par l'exemple, téléchargeant la correction et en l'étudiant.

2 Tâches du programme : première séance

Le programme doit faire les tâches suivantes, dans l'ordre, mais avec un menu composé de 4 items :

1. Ecrire un sous-programme qui génère un fichier `entree.txt` de m lignes et n colonnes de nombres aléatoires entre 0 et 1.
2. Ecrire un sous-programme qui génère un fichier `sortie.dat`, la première colonne est $x \in [0, 4\pi]$, la seconde colonne est $\cos(x)$, la troisième $\sin(x)$, la quatrième $\exp(x)$. On prendra par exemple $npt = 201$ points (paramétrable).

Utiliser un format de sortie

3. Ecrire un sous-programme qui lit un fichier `entree.txt` contenant m lignes et n colonnes
4. Ecrire un sous-programme qui écrit un fichier de sortie binaire contenant la première colonne, et la somme des colonnes de 2 à n .

Des sorties se font à l'écran, d'autres dans des fichiers.

On se servira du fichier source sur Moodle qui contient déjà un squelette du programme ainsi que certaines fonctions déjà implémentées.

3 Les quatre méthodes à implémenter : seconde séance

Il faut télécharger l'archive qui se trouve sur moodle. Suivre les indications pour la décompresser. Vous venez de créer un dossier TP2 contenant 5 dossiers : `Simple`, `Subs`, `Module`, `Lib` et `RUN`.

Pour le développement on travaille dans les 4 premiers dossiers et pour **lancer les codes on travaillera uniquement dans le dernier dossier RUN**.

On utilise ici un programme principal, très court (autour de 40 lignes) qui contient principalement des appels à des procédures, qui elles mêmes feront appel éventuellement à d'autres procédures ou fonctions.

4 méthodes sont considérées ici pour compiler le programme :

1. **Méthode "simple"** : le programme principal et toutes ses dépendances se trouvent dans un fichier unique

2. **Méthode "Subs"** : il existe un fichier pour le programme principal et un fichier par procédures ou fonctions
3. **Méthode "Module"** : il existe un fichier pour le programme principal et toutes les dépendances sont dans un module
4. **Méthode "Lib"** : il existe un fichier pour le programme principal et toutes les dépendances sont incluses dans une librairie.

Ouvrir 2 terminaux, le premier étant ouvert dans le dossier `RUN`, le second varie avec les questions et correspond au dossier de compilation du code qui s'effectue à chaque fois avec l'instruction `make`.

1. Dans `Simple`, analyser le fichier `Makefile` ainsi que le programme `fortran`. Regarder en particulier comment est écrit le code par rapport à ce que vous avez écrit.
Tester le code (dossier `RUN`), tester les 4 réponses aux questions et visualiser les courbes (par exemple avec `gp` ou `gnuplot`)
2. Dans `Subs`, Est-ce que le fichier principal est modifié ? Faire la même chose que dans la question précédente.
3. Dans `Module` Regarder les modifications dans tous les fichiers. Passer du temps sur le fichier contenant le module et sur le fichier `Makefile`
4. Dans `Lib`. Regarder dans le dossier `Librairie` et voir si les fichiers sont modifiés ou pas par rapport aux méthodes précédentes ; Analyser les différents `Makefile`. Dans le `Makefile` servant à compiler le programme principal, on vérifie que l'appel à la librairie est inclus.

4 Visualisation

Il faut récupérer l'utilitaire `Shell` qui s'appelle `gp`. Sinon on peut aussi utiliser le logiciel `xmgrace`. Tracer les courbes sinus et cosinus à partir du fichier `sortie.dat` :

1. en utilisant `gnuplot` et ses options de commande en ligne.

```
> gnuplot
>> plot 'exo.out' u 1:2 w l
```
2. en utilisant le script shell `gp` :

```
> gp l fichier.dat 1 2
```