

TP d'informatique
Première année apprentissage, mécanique des fluides
Lecture de paramètres par fichier d'entrée, allocation dynamique,
discrétisation

Travail à mener sur une seule séance.

1 Objectif du TP

- Apprendre à utiliser un fichier de paramètres
- Comprendre des modules ou procédures déjà écrites
- Assembler dans un programme principale les différentes fonctions pour répondre aux fonctions du programme.
- Apprendre à déclarer des tableaux dynamiques
- Comprendre comment calculer numériquement une dérivée de fonction.

2 Introduction

Pour éviter la tâche répétitive d'entrer les paramètres ou options d'un programme au le clavier il est recommandé de toujours écrire un fichier de paramètres qu'on ouvre et modifie en cas de besoin.

Lire ce fichier est très souvent la première tâche d'un programme.

Dans ce TP, on devoir uniquement écrire le programme principal en comprenant ce qu'il doit faire et en utilisant les sources fournies.

3 Objet du programme

Le programme a pour objectif de calculer numériquement les dérivées de fonctions en utilisant un schéma de discrétisation par différences finies d'ordre 2. Ce genre de schémas numériques sont obtenus par des développements limités en série de Taylor.

Le schéma est le suivant pour une maillage régulier, de taille n , de pas constant h :

$$f'_{num}(x_i) \approx \frac{f(x_{i+1}) - f(x_{i-1}))}{2h}, \quad \text{pour } i = 2..n - 1$$

$$f'_{num}(x_1) \approx \frac{-3f(x_1) + 4f(x_2) - f(x_3))}{2h}$$

$$f'_{num}(x_n) \approx \frac{3f(x_n) - 4f(x_{n-1}) + f(x_{n-2}))}{2h}$$

Les trois fonctions testées sont :

1. $y_1(x) = \sin^2(x) \cos(x)$
2. $y_2(x) = 2x + 0.5x^2$
3. $y_3(x) = 2 \exp(-0.05x) \cos(3x)$

pour $0 \leq x \leq 4$.

On validera le code en comparant la solution analytique à la solution trouvée numériquement sur la base d'une erreur relative définie ainsi :

$$\varepsilon(x_i) = \left| 1 - \frac{y'_{num.}(x_i)}{y'_{analy.}(x_i)} \right|, \quad \text{si } y'_{analy.}(x_i) \neq 0$$

et $\varepsilon_{\log}(x_i) = \log 10(\varepsilon(x_i))$

4 Déroulement du TP

5 Tâches à implémenter dans le programme

La programmation a lieu dans le dossier `Fortran`. Mais on lance le programme dans `RUN`.

1. Charger l'archive `TP3.tar` sur moodle et l'ouvrir avec par exemple `tar -xvf TP3.tar` dans un terminal. On travaillera dans le dossier `tp3`
2. L'archive comprend trois dossiers, celui contenant les sources fortran, celui où sera lancé le code, et un dossier à `Latex` contenant l'énoncé du TP.
3. Le programme doit effectuer différentes tâches pour fonctionner et être validé :
 - (a) Lire les paramètres
 - (b) Allouer les variables (allocation dynamique)
 - (c) Calculer la solution $y(x)$ et sa dérivée analytique
 - (d) Calculer la dérivée de la fonction numériquement.
 - (e) Calculer créer une fonction d'erreur
 - (f) On cherchera le maximum de l'erreur, ainsi que sa position
 - (g) Dans une fichier de sortie `tp3.out`, écrire les valeurs de la x , de la fonction, des dérivées analytiques et numériques, de l'erreur et du `LOG10` de l'erreur (attention aux valeurs !)
4. Un travail préliminaire a été effectué. Un module et des procédures ou fonctions ont été créées. Ce sont les sources du code.

Le fichier des paramètres `tp3.in` se trouve lui dans le dossier `RUN`. Vérifier si le fichier présent contient les éléments suivants :

- (a) `option` qui permet de choisir une fonction (au choix, 3 fonctions : une contenant des cosinus et sinus, une avec un polynôme et une avec des exponentielles)
 - (b) x_1 et x_2 qui définissent les intervalles d'études
 - (c) `npt` qui définit le nombre de points du maillage (constant)
5. Ouvrir le module, analyser son contenu. Il contient des paramètres ou variables globales et des tableaux dynamiques. Vérifier la conformité avec les objectifs du problème.
 6. Ouvrir le fichier contenant l'ensemble des fonctions ou procédures, et les analyser une par une pour comprendre les points suivants :
 - Les entrées
 - Les sorties
 - Les variables locales

— Les tâches effectuées

7. Ouvrir le `Makefile` et vérifier sa configuration.
8. Seul le fichier `tp3.f90` sera complété, les autres fichiers ne doivent pas être modifiés. A partir des codes sources et de leur analyse, il faut écrire le code principal pour remplir les tâches décrites plus haut.
Il faut procéder très progressivement en compilant souvent pour vérifier de l'exactitude du code entré.
9. Dans le dossier `RUN` tester le code et vérifier que les objectifs sont atteints, notamment en dessinant à l'écran les différentes courbes, fonctions, dérivées, erreurs relatives pour différentes fonctions et jeux de paramètres (`gnuplot`, `xmgrace`).
Faire une étude paramétrique en fonction de la fonction sélectionnée et du nombre de points de discrétisation (101, 201, 1001, 5001)
10. Tirer des conclusions qualitatives et quantitatives notamment sur les erreurs de discrétisation.