TE 1-B : Dérivation numérique

1 Objectif de l'étude

1.1 Dérivation par différences finies

On s'intéresse à la dérivation numérique d'une fonction f **en un point** x_0 **fixé** via différents schémas aux différences finies de pas h et au **choix du pas** h permettant de minimiser les erreurs.
Plus précisément, on étudie des schémas centrés à deux, puis à quatre termes pour la dérivée première :

— Estimation centrée à deux termes

$$f'(x_0) \approx f_1(x_0) = \frac{f(x_0+h) - f(x_0-h)}{2h} \tag{1}$$

— Estimation centrée à quatre termes

$$f'(x_0) \approx f_{1b}(x_0) = \frac{-f(x_0+2h) + 8f(x_0+h) - 8f(x_0-h) + f(x_0-2h)}{12h} \tag{2}$$

1.2 Erreurs associées aux estimateurs

Deux types d'erreurs indépendantes interviennent dans ces approximations et s'ajoutent en valeur absolue :

— L'**erreur de troncature** déterministe, liée au nombre limité de termes du schéma aux différences finies. On peut l'estimer à partir d'un développement en série de Taylor avec reste : $h^2|f^{(3)}(x_0)|/6$ pour l'estimateur (1) à 2 termes et $h^4|f^{(5)}(x_0)|/30$ pour l'estimateur (2) à 4 termes.

— L'**erreur d'arrondi** pseudo-aléatoire, qui dépend de la précision relative ε des flottants choisis pour le calcul. Pour la dérivée première, sa borne supérieure décroît avec le pas en $1/h$: $|f(x_0)|\varepsilon/h$ pour l'estimateur (1) et $3|f(x_0)|\varepsilon/2h$ pour l'estimateur (2).

1.3 Recherche graphique du pas minimisant l'erreur totale

Les valeurs absolues de ces erreurs dépendent du pas selon des lois en puissance h^p : $p < 0$ pour l'arrondi et $p > 0$ pour la troncature. L'erreur totale est donc dominée par l'erreur d'arrondi à pas très faible et par l'erreur de troncature à grand pas. Le tracé de l'erreur totale en fonction du pas permet de mettre en évidence ces domaines respectifs et de vérifier graphiquement les lois en puissance.
À cet effet, on choisit une représentation en échelles log-log ; pour répartir régulièrement les points sur l'axe des pas, les valeurs du pas sont choisies selon une **progression géométrique** de raison r : si n est le nombre de valeurs du pas, r est donné par $r^{n-1} = h_{\max}/h_{\min}$.
Pour tracer en échelles log sur les deux axes, on utilisera : la fonction `loglog` de `matplotlib.pyplot` à la place de `plot`.
L'objectif final est d'estimer graphiquement la valeur du pas minimisant l'erreur totale d'estimation.

1.4 Fonctions testées

Estimer, en simple précision, la dérivée de $f(x) = \sin(kx)$ en $x_0 = \pi/4$, pour, par exemple, $k = 1$ avec $10^{-4} \leqslant h \leqslant 1$. Dans la mesure du temps disponible, reprendre le calcul avec $f(x) = \exp(kx)$ en $x_0 = 0$.

2 Partie calcul en fortran ou C

2.1 Organisation des codes

La structure générale doit permettre la compilation séparée des quatre fichiers via `make` :

1. Un programme principal (fichier `main.f90` ou `main.c`) auquel est confié le dialogue avec l'utilisateur pour choisir le nombre `n` de valeurs du pas, le plus petit pas `hmin` et le plus grand pas `hmax`, l'abscisse `x_0` du point où seront calculées les dérivées et enfin la fonction à étudier. Ce programme calcule la dérivée $f'(x_0)$ avec l'expression analytique. Il alloue ensuite le tableau `tabh` des pas et celui `delta` des erreurs. Pour tabuler l'erreur en fonction du pas, il lance alors une boucle sur les pas dans laquelle :
   — il calcule le pas et le stocke dans `tabh`
   — il appelle l'estimateur centré `deriv1` ou `deriv1b`
   — il calcule la valeur absolue de l'écart avec la valeur de l'expression analytique $f'(x_0)$ de la dérivée et la stocke dans le tableau des erreurs `delta`.

Il fournit enfin les tableaux de résultats à la procédure d'écriture sur fichier.

2. Un module (fichier `fcts.f90` ou `fcts.c`) définissant les différentes fonctions à tester ainsi que leurs dérivées analytiques.
3. Un module (fichier `diff.f90` ou `diff.c`) hébergeant les fonctions `deriv1`, `deriv1b` qui évaluent la dérivée de la fonction passée en argument à l'abscisse fournie par le programme appelant. Comme ces opérateurs évaluent la fonction test f en plusieurs points autour de x_0, il est nécessaire de passer à `deriv1` et `deriv1b` un argument de type fonction.
4. Une procédure `ecrit` chargée d'écrire les résultats dans un fichier formaté.

Module des fonctions à tester
Fichier `fcts.f90` ou `fcts.c`

Définition des différentes fonctions à tester et de leurs dérivées analytiques
Un seul argument pour chacune des fonctions ;
les paramètres éventuels sont des variables globales du module.

Module de la méthode à appliquer
Fichier `diff.f90` ou `diff.c`

Arguments des dérivateurs :

— la fonction à tester
— l'abscisse du point où on calcule les dérivées
— le pas

Programme principal
Fichier `main.f90` ou `main.c`

— Saisie des paramètres
— Construction des vecteurs
— Calcul de la dérivée analytique en x_0
— Boucle sur les pas avec remplissage du tableau des pas, appel de la méthode `deriv1` pour estimer les dérivées et remplissage du tableau des écarts à la dérivée analytique
— Appel de la procédure d'écriture des résultats

Module d'écriture sur fichier
Fichier `util.f90` ou `util.c`

— Ouverture du fichier
— Écriture de l'entête avec valeurs extrêmes (+)
— Boucle d'écriture des $h, \delta_1(h)$: à raison d'un couple par ligne
— Fermeture du fichier

2.2 Structure du fichier de résultats

Le fichier de résultats comportera deux lignes d'**entête (+)**indiquant respectivement :
— le nombre de valeurs du pas et les valeurs minimale et maximale du pas ;
— les bornes des valeurs absolues des erreurs (valeurs minimale et maximale [1]) ;
Cet entête ne sera implémenté qu'après un premier test et une visualisation.
Le **corps du fichier** comportera une ligne par valeur du pas, ligne constituée de la valeur du pas suivie des valeurs absolues des erreurs. On spécifiera un format compatible avec la dynamique des valeurs et assurant une précision relative de l'ordre de 10^{-7}.

2.3 Exemple d'interface des fonctions de dérivation

```
REAL FUNCTION deriv1(f, x0, h)
  REAL, INTENT(in) :: x0, h
  INTERFACE ! de la fonction générique
    REAL FUNCTION f(x1)
       REAL, INTENT(in) :: x1
    END FUNCTION f
  END INTERFACE
END FUNCTION deriv1
```

```
#ifndef DIFF_H
#define DIFF_H
float deriv1(float (*pf)(float),
             float x, float h);
float deriv1b(float (*pf)(float),
             float x, float h);
#endif /*  DIFF_H */
```

2.4 Influence de la précision des flottants

Dans la mesure du temps disponible, on reprendra l'étude avec des réels sur 64 bits (voir les outils de paramétrage rappelés en cours) pour évaluer l'influence de la précision ε sur l'erreur d'arrondi.

1. En langage C, on pourra définir des fonctions `maxabsfloat1d` et `minabsfloat1d` pour calculer les maximum et minimum de la valeur absolue d'un tableau 1D de float.