Travaux pratiques - Intégration Numérique

Contents

1 Objectifs du TP

Les codes fournis en ligne permettent d'illustrer les différentes notions vues en cours : **pointeurs**, **allocation dynamique**, **Makefile**. Ils ont été écrits uniquement dans un but pédagogique et ils sont d'être optimaux. Ils contiennent les programmes principaux `essai01` à `essai04` et une documentation partielle au format html.

1.1 Lecture et compréhension rapide de codes

1.1.1 essai01.c

Générer le code avec la commande `make essai01`. Exécuter le code et, en s'aidant de la "documentation", décrire ce qu'il calcule. On peut noter que $\int_0^1 \sqrt{x(1-x)}dx = \frac{\pi}{8}$.

Exécuter la commande `make clean` suivi de `make DEBUG=1 VERBOSE=1 essai01`. Executer le code et comprendre son nouveau fonctionnement.

1.1.2 essai02.c

Générer le code avec la commande `make essai02`. Exécuter le code et, en s'aidant de la "documentation", décrire ce qu'il calcule.

Exécuter la commande `make clean` suivi de `make DEBUG=1 VERBOSE=1 essai02`. Executer le code et comprendre son nouveau fonctionnement.

1.1.3 essai03.c

Sur le même principe que les deux premiers programmes, décrire ce qui est sauvegardé dans le fichier *essai03.out*.

1.1.4 essai04.c

Générer le code avec la commande `make essai04`. Exécuter successivement les commandes

```
./essai04 help
./essai04
./essai04 titi.out
./essai04 toto.out 500 500 10000
```

puis après avoir configurer la variable/macro `MATLAB` du fichier `Makefile`, exécuter la commande `make test04`. Villarceau

1.2 Modification de codes existants

Dans cette partie, il vous est demandé d'ajouter les méthodes composites des Trapèzes, de Simpson et de Villarceau pour le calcul d'intégrales. Il faudra, sur le même modèle que le code `essai04`, écrire le fichier `essai05.c` contenant un programme permettant de calculer les erreurs commises par chacune des méthodes puis de sauvegarder l'ensemble des résultats dans un fichier. On modifiera, entre autres, le `makefile` pour permettre la compilation par la commande `make essai05`. La commande `make test05` permettra quand à elle d'obtenir une représentation graphique des différents ordres des méthodes (voir par exemple la Figure 1).

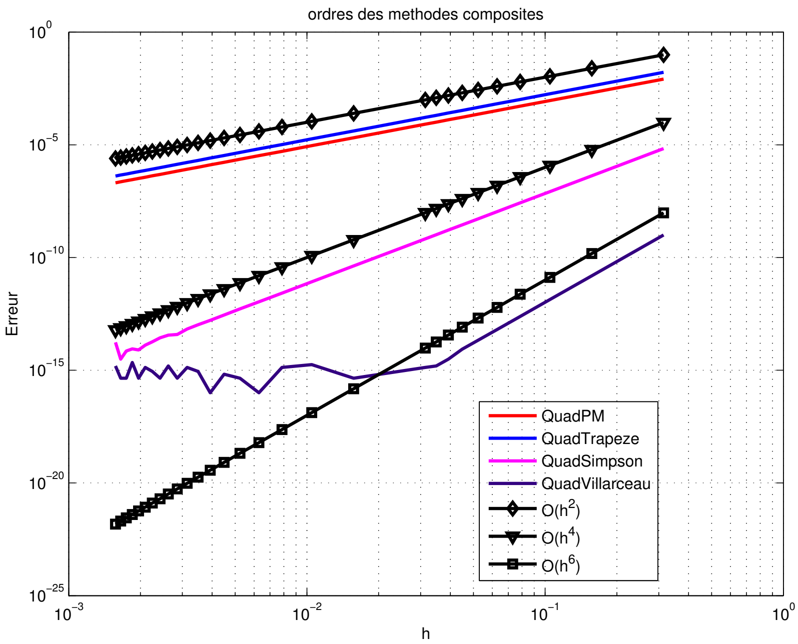

Figure 1: Ordre de l'erreur des méthodes composites

1.3 Amélioration de codes existants : utilisation de structures

typedef struct int dim; double *v; char name[80]; array1D;

2 Méthodes composites pour le calcul de $\int_A^B f(x)dx$

On rappelle ici brièvement le principe de base des méthodes composites pour l'approximation numérique d'intégrale du type $\int_A^B f(x)dx$. Une présentation plus détaillée, correspondant à un cours donné en 2012-2013 aux ingénieurs Energétiques 1ère année (apprentissage), est disponible en ligne : AlgoNum_EnerApp.pdf (page 26)

Les méthodes composites sont basées sur la relation de Chasles.

Soit $(x_k)_{k\in[\![0,n]\!]}$ une discrétisation régulière de l'intervalle $[A,B]$: $x_k = A + kh$ avec $h = (B-A)/n$. On a alors

$$\int_A^B f(x)dx = \sum_{k=1}^{n} \int_{x_{k-1}}^{x_k} f(x)dx. \tag{0.1}$$

2.1 Description de différentes méthodes

2.1.1 Méthodes composites des points milieux

On note $m_k = \frac{x_{k-1}+x_k}{2}$ et on approche chacune des intégrales par la formule du point milieux :

$$\int_{x_{k-1}}^{x_k} f(x)dx \approx hf(m_k).$$

Théorème 1 *Soit f une fonction définie et suffisament régulière sur l'intervalle $[A, B]$. On a alors*

$$\int_A^B f(x)dx = h\sum_{k=1}^{n} f(m_k) + \mathcal{O}(h^2).$$

Cette formule de quadrature est d'ordre 1: elle est exacte pour les polynômes de degré inférieur ou égal à 1. L'erreur commise est en $\mathcal{O}(h^2)$: elle est donc d'ordre 2. ■

2.1.2 Méthodes composites des trapèzes

On approche chacune des intégrales de (0.1) par la formule des trapèzes

$$\int_{x_{k-1}}^{x_k} f(x)dx \approx h\frac{f(x_{k-1} + f(x_k)}{2}.$$

Théorème 2 *Soit f une fonction définie et suffisament régulière sur l'intervalle $[a, b]$. On a alors*

$$\int_A^B f(x)dx = h\sum_{k=1}^{n} \frac{f(x_{k-1}) + f(x_k)}{2} + \mathcal{O}(h^2).$$

Cette formule de quadrature est d'ordre 1: elle est exacte pour les polynômes de degré inférieur ou égal à 1. L'erreur commise est en $\mathcal{O}(h^2)$: elle est donc d'ordre 2. ■

2.1.3 Méthodes composites de Simpson

On approche chacune des intégrales de (0.1) par la formule de Simpson

$$\int_{x_{k-1}}^{x_k} f(x)dx \approx \frac{h}{6}(f(x_{k-1}) + 4f(m_k) + f(x_k)).$$

Théorème 3 *Soit f une fonction définie et suffisament régulière sur l'intervalle $[a, b]$. On a alors*

$$\int_A^B f(x)dx = \frac{h}{6}\sum_{k=1}^{n}(f(x_{k-1}) + 4f(m_k) + f(x_k)) + \mathcal{O}(h^4).$$

Cette formule de quadrature est d'ordre 3: elle est exacte pour les polynômes de degré inférieur ou égal à 3. L'erreur commise est en $\mathcal{O}(h^4)$: elle est donc d'ordre 4. ■

2.1.4 Méthode composite de Villarceau

On rappelle que les formules de Newton-Cotes génériques sont données par

$$\int_a^b f(x)dx \approx \sum_{i=0}^{m} \alpha_i f(x_i).$$

Avec $x_i = a + ih$, $\forall i \in [\![0, m]\!]$ et $h = (b-a)/m$. En posant $\alpha_i = hAw_i$, on a

m	A	w_0	w_1	w_2	w_3	w_4	nom	ordre
1	1/2	1	1				trapèzes	1
2	1/3	1	4	1			Simpson	3
3	3/8	1	3	3	1		Simpson (3/8)	3
4	2/45	7	32	12	32	7	Villarceau	5

On va donc approcher chacune des intégrales de (0.1) par la formule de Villarceau.

Théorème 4 *Soit f une fonction définie et suffisament régulière sur l'intervalle $[A, B]$. La méthode* ***composite*** *de Villarceau est d'ordre 5: elle est exacte pour les polynômes de degré inférieur ou égal à 5. L'erreur commise est en $\mathcal{O}(h^6)$: elle est donc d'ordre 6.* ▪