Chapitre 1

Analyse d'erreurs et arithmétique de l'ordinateur

Sommaire

À la différence des méthodes de résolution classiques, par exemple les techniques d'intégration et de résolution d'équations algébriques ou différentielles, les méthodes d'analyse numérique proposent des techniques d'approximations qui résultent en des algorithmes. Ces algorithmes dépendent sur des paramètres dont leur choix permet, après un nombre fini d'opérations élémentaires, d'avoir une solution approchée du problème avec une précision donnée.
Il est ainsi fondamental de comprendre et savoir analyser les erreurs produites par un algorithme (ou un procédé) d'approximation. Une partie importante de l'analyse numérique consiste donc à contenir les effets des erreurs ainsi introduites, qui sont principalement de trois sources :

- les erreurs de modélisation : Ces erreurs sont en dehors du cadre du cours et proviennent de l'étape de mathématisation du phénomène physique auquel on s'intéresse (Ex : Voir le mouvement du pendule [a]).
- les erreurs de représentation sur ordinateur : Un nombre occupe un espace physique dans un ordinateur, avec une limite sur les ressources disponibles pour le représenter par arrondi ou troncature.
- les erreurs de troncature : Ces erreurs proviennent principalement de l'utilisation du développement de Taylor, qui permet par exemple de remplacer une équation différentielle par une équation algébrique.

a. André Fortin : Analyse numérique pour ingénieurs, Presses Internationales Polytechnique, Montréal (2016), pages 4-7.

1 Erreur absolue et erreur relative

Définition 1.1

Soit x un nombre et x^* une approximation de x. L'erreur absolue, notée $E_a(x^*)$, est la quantité définie par

$$E_a(x^*) = |x - x^*|. \tag{1.1}$$

Définition 1.2

Soit x un nombre et x^* une approximation de x. L'erreur relative, notée $E_r(x^*)$, est la quantité définie par

$$E_r(x^*) = \frac{|x - x^*|}{|x|} = \frac{E_a(x^*)}{|x|}, \ x \neq 0. \tag{1.2}$$

Remarque(s) :

- L'erreur absolue $E_a(x^*)$ donne une mesure quantitative de l'erreur commise en estimant la valeur exacte x, à partir de x^*.
- L'erreur relative $E_r(x^*)$ mesure l'importance (une mesure qualitative) de l'erreur commise en estimant x, à partir de x^*. Elle est souvent exprimée en pourcentage en la multipliant par 100, pour la rendre plus significative.

Exemple 1

On approxime le nombre $x = \pi$ par la quantité $x^* = \frac{22}{7} = 3,142857\cdots$. Déterminons l'erreur absolue ainsi que celle relative, générées par cette approximation.

- erreur absolue :
$$E_a(x^*) = |\pi - \frac{22}{7}| = 0,001264489\cdots \simeq 0,1264489 \times 10^{-2}.$$
- erreur relative :
$$E_r(x^*) = \frac{|\pi - \frac{22}{7}|}{|\pi|} = 0,000402499\cdots \simeq 0,402499 \times 10^{-3}.$$

En pratique, il est difficile d'évaluer les erreurs absolue et relative car le plus souvent, on ne connaît pas la valeur exacte de x ; on a que la valeur approximée x^*. Cependant, par le même procédé donnant x^*, on dispose souvent d'une *borne supérieure* de l'erreur absolue, qu'on note Δx. La borne Δx est quand même considérée comme étant l'erreur absolue, alors qu'en fait on a

$$|x - x^*| \leq \Delta x.$$

L'erreur relative $E_r(x^*)$ est ainsi remplacée par le rapport $\frac{\Delta x}{|x^*|}$

$$E_r(x^*) \simeq \frac{\Delta x}{|x^*|}.$$

Note: On a les équivalences suivantes :

$$|x - x^*| \leq \Delta x \iff -\Delta x \leq x - x^* \leq \Delta x \iff -\Delta x + x^* \leq x \leq \Delta x + x^*.$$

On note parfois $x = x^* \pm \Delta x$ et on interpréte ce résultat en disant que l'on a estimé la valeur exacte x à partir de x^* avec une incertitude de Δx de part et d'autre.

Définition 1.3

Si l'erreur absolue vérifie $\Delta x \leq 0,5 \times 10^m$, alors le chiffre dans x^* correspondant à la m^e puissance de 10 est dit *significatif* et tous ceux à sa gauche, correspondant aux puissances de 10 supérieures à m, le sont aussi. On arrête le compte au dernier chiffre non nul.

Exemple 2

À l'Exemple 1 on a l'erreur absolue, dans l'approximation de $x = \pi$ par $x^* = \frac{22}{7} = 3,142857\cdots$, qui est donnée par

$$E_a(x^*) = |\pi - \frac{22}{7}| = 0,001264489\cdots \simeq 0,1264489 \times 10^{-2}.$$

Puisque l'erreur absolue est plus petite que $0,5 \times 10^{-2}$, alors le chiffre des centièmes est significatif et on a en tout 3 chiffres significatifs (en l'occurrence $3,14$).
Si l'on retient que $x^* = 3,1416$ comme approximation de $x = \pi$, alors on a l'erreur absolue

$$E_a(x^*) = |\pi - 3,1416| \simeq 0,73 \times 10^{-5}.$$

Ainsi, on a l'estimation suivante de l'erreur absolue

$$E_a(x^*) \simeq 0,73 \times 10^{-5} = 0,073 \times 10^{-4} \leq 0,5 \times 10^{-4}.$$

Donc le chiffre (sur l'approximation $x^* = 3,1416$) correspondant à la $4^{ième}$ position après la virgule (ici 6), est significatif ainsi que tous les chiffres situés à sa gauche. L'approximation $x^* = 3,1416$ possède donc 5 chiffres significatifs.

Note: Il est à noter que le chiffre 6 dans $x^* = 3,1416$ est significatif même si la quatrième décimale de π est un 5 ($\pi = 3,14159\cdots$). Significatif ne veut pas dire exact, mais dont on contrôle l'erreur, qui ont du sens.

Définition 1.4

Inversement, si une approximation x^* est donnée avec un nombre n de chiffres significatifs, on commence à compter à partir du premier chiffre non nul à gauche et l'erreur absolue est inférieure à $0,5$ fois la puissance de 10 correspondant au dernier chiffre significatif.

Exemple 3

On a mesuré le poids d'une personne à l'aide d'un appareil jugé suffisamment précis et trouvé $90,567\ kg$ où tous les chiffres sont supposés significatifs. Puisque le dernier chiffre significatif correspond aux millièmes (milligrammes), cela signifie que

$$\Delta x \leq 0,5 \times 10^{-3} kg.$$

En pratique, on considère que $\Delta x = 0,5 \times 10^{-3} kg$.

Note: Il existe une exception à la règle. Si le chiffre correspondant à la m^e puissance de 10 est nul ainsi que tous ceux à sa gauche, on dit qu'il n'y a aucun chiffre significatif.

Remarque(s) : En pratique, on cherchera à déterminer une borne pour Δx aussi petite que possible et donc la valeur de m la plus petite possible.

Exercice du manuel (fortement suggéré) 1: Tous les chiffres des nombres suivants sont significatifs. Donner une borne supérieure de l'erreur absolue et estimer l'erreur relative.

$a)$	$0,1234$	$b)$	$8,760$	$c)$	$3,14156$
$d)$	$0,11235 \times 10^{-3}$	$e)$	$8,000$	$f)$	$0,22356 \times 10^{8}$

2 Représentation des nombres sur ordinateur

Un ordinateur ne peut traiter les nombres de la même manière que l'être humain. Il doit d'abord les représenter dans un système qui permet l'exécution efficace des diverses opérations. Cela peut entraîner des erreurs de représentation sur ordinateur, qui sont inévitables et dont il est souhaitable de comprendre l'origine afin de mieux en maîtriser les effets.

Cette section présente les principaux éléments d'un modèle de représentation des nombres sur ordinateur. La structure interne de la plupart des ordinateurs s'appuie sur le système binaire qui prend que les valeurs 0 ou 1 (appelées bit). Évidemment, très peu d'information peut être accumulée au moyen d'un seul bit. Ainsi, les bits sont regroupés en mots de longueur variable dont les plus courants sont ceux de longueurs 8, 16, 32 ou 64 bits.

• 2.1 - Conversion en valeur binaire

Puisque le système binaire est à la base de la représentation des nombres dans la vaste majorité des ordinateurs, nous rappelons brièvement comment convertir des entiers positifs et des fractions décimales en notation binaire.

• 2.1.1 - Représentation des entiers positifs en binaire

Pour transformer un entier positif N dans sa représentation binaire habituelle, il faut déterminer les a_i tels que :

$$(N)_{10} = (a_{n-1}a_{n-2}a_{n-3}\cdots a_2a_1a_0)_2$$

où l'indice inférieur indique la base utilisée. Une autre représentation est

$$N = a_{n-1} \times 2^{n-1} + a_{n-2} \times 2^{n-2} + a_{n-3} \times 2^{n-3} + \cdots + a_2 \times 2^2 + a_1 \times 2^1 + a_0 \times 2^0$$

Question: Comment déterminer les bits a_i permettant la représentation binaire de l'entier N ?

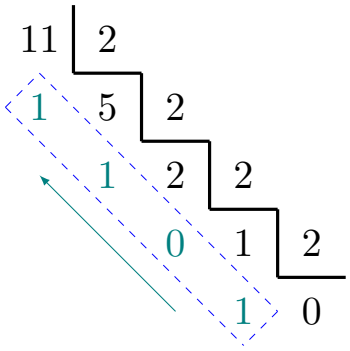
Méthode : On obtient la valeur des a_i par la démarche suivante :

- on divise N par 2 pour obtenir a_0 (le reste de la division) plus un entier ;
- on refait le même raisonnement avec la partie entière de $\frac{N}{2}$ (en négligeant la partie fractionnaire ou reste) pour obtenir a_1 ;

- on continue ainsi jusqu'à ce que la partie entière soit nulle.

Exemple 4

Pour convertir le nombre décimal $N = (11)_{10}$ en binaire, on procède comme suit :



Ainsi, l'entier décimal 11 s'écrit 1011 en binaire ($(11)_{10} = (1011)_2$).

- **2.1.2 - Conversion d'une fraction décimale en valeur binaire**

Soit f, une fraction décimale comprise entre 0 et 1. Sa conversion en binaire consiste à trouver les bits d_i tels que :

$$(f)_{10} = (0, d_1 d_2 d_3 \cdots)_2 = d_1 \times 2^{-1} + d_2 \times 2^{-2} + d_3 \times 2^{-3} + \cdots$$

La suite $d_1 d_2 d_3 \cdots$ sera appelée la mantisse et sera détaillée à la Sous-section (2.3).

Question: Comment déterminer les bits d_i afin d'obtenir la représentation binaire de la fraction décimale f ?

Méthode : Pour obtenir les bits d_i, on procède comme suit :

- on multiplie f par 2, pour obtenir d_1 plus une fraction ;
- on applique le même raisonnement à la fraction $(2f - d_1)$ obtenue, on obtient d_2 plus une autre fraction ;
- on poursuit ainsi jusqu'à ce que la partie fractionnaire soit nulle ou que l'on ait atteint le nombre maximal de chiffres prévu à la mantisse.

Exemple 5

Pour convertir la fraction décimale $f = 0,0625$, on procède comme suit :

$$\begin{aligned} 0,0625 \times 2 = 0,1250 &\longrightarrow d_1 = 0 \\ 0,1250 \times 2 = 0,2500 &\longrightarrow d_2 = 0 \\ 0,2500 \times 2 = 0,5000 &\longrightarrow d_3 = 0 \\ 0,5000 \times 2 = 1,0000 &\longrightarrow d_4 = 1 \end{aligned}$$

signifiant ainsi $(0,0625)_{10} = (0,0001)_2$.
Pour la fraction décimale $f = \frac{1}{3}$, on procède de la même façon :

$$\begin{aligned} \frac{1}{3} \times 2 = \frac{2}{3} + 0 &\longrightarrow d_1 = 0 \\ \frac{2}{3} \times 2 = \frac{1}{3} + 1 &\longrightarrow d_2 = 1 \\ \frac{1}{3} \times 2 = \frac{2}{3} + 0 &\longrightarrow d_3 = 0 \\ \frac{2}{3} \times 2 = \frac{1}{3} + 1 &\longrightarrow d_4 = 1 \\ \vdots & \quad \vdots \end{aligned}$$

On peut poursuivre la conversion à l'infini et obtenir

$$(\frac{1}{3})_{10} = (0,010101 \cdots)_2$$

Note: En pratique, puisque l'on n'utilise qu'un nombre fini de chiffres dans la mantisse, il faudra en un certain moment s'arrêter après n bits.

Remarque(s) : Une machine (ordinateur, calculatrice, $\cdots$) ne peut stocker qu'un nombre fini de chiffres (bits en binaire) pour représenter un nombre entier ou réel donné. Ainsi, un recours est fait à la troncature ou à l'arrondi :

- Si l'on travaille en notation décimale et si l'on souhaite utiliser la troncature avec 4 chiffres dans la mantisse (n dans le cas général), on coupe les décimales restantes à partir de la cinquième ($(n+1)^e$).
- Pour arrondir, on ajoute 5 unités au cinquième ($(n+1)^e$) chiffre de la mantisse et l'on tronque le résultat.

Note: Ces procédés s'appliquent de la même façon en binaire (ou base 2).

Exemple 6

- Par troncature à 4 chiffres, le nombre $0,1234672$ devient tout simplement $0,1234$.
- Alors que pour arrondir celui-ci à 4 chiffres, on ajoute 5 au cinquième chiffre de la mantisse qui devient alors $0,1235172$ et que l'on tronque pour obtenir $0,1235$.

2.2 - Représentation des entiers signés

Nous présenterons ici, les deux formes les plus courantes de représentation sur ordinateur d'un entier signé. Elles s'agissent de la représentation en complément à 2 et celle dite par excès.

2.2.1 - Représentation en complément à 2

Si l'on dispose de n bits à la mantisse pour exprimer l'entier N en binaire , alors celui-ci s'écrirait

$$N = -a_{n-1} \times 2^{n-1} + a_{n-2} \times 2^{n-2} + a_{n-3} \times 2^{n-3} + \cdots + a_2 \times 2^2 + a_1 \times 2^1 + a_0 \times 2^0,$$

où il est important de remarquer le signe négatif devant le terme a_{n-1}.

Méthode :

- Un entier positif est représenté par 0 suivi de son expression binaire habituelle en $(n-1)$ bits.
- Pour obtenir la représentation d'un nombre négatif N, il suffit de lui ajouter 2^{n-1} et de transformer le résultat en forme binaire. Le résultat final est donc 1 suivi de la représentation sur $(n-1)$ bits de $N + 2^{n-1}$.

Exemple 7

La représentation en complément à 2 sur 4 bits de 0101 vaut :

$$-0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0.$$

En forme décimale, cela correspond au nombre 5.
Par contre, La représentation en complément à 2 sur 4 bits de 1101 vaut :

$$-1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0.$$

Ce qui correspond en forme décimale au nombre $-8 + 5 = -3$.
Inversement, la représentation binaire de -6 sera 1 suivi de la représentation sur 3 bits du nombre

$$-6 + 2^3 = 2$$

qui est 010. On écrira donc $(-6)_{10} = (1010)_2$ dans la représentation en complément à 2.

2.2.2 - Représentation par excès

Méthode :

- Si l'on veut exprimer un entier décimal N avec un excès d, il suffit de lui ajouter l'excès et de représenter le résultat sous forme binaire.

- Inversement, si l'on a la représentation binaire par excès d'un entier, il suffit de calculer sa valeur en base 10 et de soustraire l'excès d pour obtenir l'entier recherché.

Note: En général, la valeur de l'excès d est 2^{n-1} ou $2^{n-1}-1$ (soit 2^3 ou 2^3-1 pour $n=4$), où n est le nombre de bits à la mantisse avec lequel la représentation est faite.

Exemple 8

Soit un mot de 8 bits et un excès $d = 2^7 - 1 = 127$. Pour représenter $(-100)_{10}$, il suffit de lui ajouter 127, ce qui donne 27, et d'exprimer le résultat sur 8 bits, soit 00011011.

2.3 - Représentation des nombres réels

Définition 2.1

Soit x un nombre réel. On appelle représentation en notation flottante ou notation en point flottant ou encore en virgule flottante du nombre x, s'il est écrit sous la forme

$$x = \pm m \times b^l, \tag{1.3}$$

où l est l'exposant, b est la base et m est la mantisse dont la forme générale est

$$m = 0, d_1 d_2 d_3 \cdots d_n \cdots$$

Cette écriture de la mantisse signifie

$$m = d_1 \times b^{-1} + d_2 \times b^{-2} + d_3 \times b^{-3} + \cdots + d_n \times b^{-n} + \cdots$$

où les d_i vérifient

$$1 \leq d_1 \leq (b-1) \quad \text{et} \quad 0 \leq d_i \leq (b-1), \text{ pour } i = 2, 3, \cdots, n, \cdots .$$

L'inégalité $d_1 \geq 1$ signifie que la mantisse est normalisée et garantit l'unicité de la représentation (1.3).

Note: La mantisse satisfait $\frac{1}{b} \leq m < 1$ et la représentation de 0 devient une exception puisque la mantisse ne s'annule jamais. Dans le système décimal, la représentation (1.3) devient $x = \pm m \times 10^l$.

Exemple 9

Pour une mantisse de longueur infinie, la troncature et l'arrondi sont également envisagés :

- Troncature à n chiffres : on coupe la mantisse m au-delà de la position n

$$m = d_1 \times b^{-1} + d_2 \times b^{-2} + d_3 \times b^{-3} + \cdots + d_n \times b^{-n}.$$

- Arrondi à n chiffres :
  1. On ajoute (5 en décimal, 1 en binaire) à la position $n+1$
  2. On fait une troncature à n chiffres

$$m = d_1 \times b^{-1} + d_2 \times b^{-2} + d_3 \times b^{-3} + \cdots + \tilde{d}_n \times b^{-n}.$$

Pour représenter le nombre $10,2$ en notation flottante à 4 chiffres dans la mantisse et en base $b = 10$, on a ces deux formes

$$0,1020 \times 10^2 \quad \text{et} \quad 0,0102 \times 10^3.$$

Grâce à la normalisation, la dernière expression n'est pas retenue ; ce qui évite toutes ambiguïtés.

Remarque(s) : Le choix de la base b est arbitraire, cependant avec le temps des préférences sont apparues :

- Humains : usuellement décimale ($b = 10$).

- Ordinateurs : usuellement binaire ($b = 2$) ou hexadécimale ($b = 16$) avec quelques exceptions (décimale, octale).

Note: Pour la représentation d'un nombre réel sur ordinateur, la base sera généralement $b = 2$ et elle consistera à représenter la mantisse (une fraction), l'exposant (un entier signé) et le signe de ce nombre.

2.4 - Erreurs dues à la représentation

Le fait d'utiliser un nombre limité de bits pour représenter un nombre réel a des conséquences importantes :

- D'une part, on introduit une erreur de représentation qui peut avoir des répercussions significatives sur la précision des résultats.
- D'autre part, quel que soit le nombre de bits utilisés, il existe un plus petit et un plus grand nombres positifs représentables. Ainsi à l'intérieur de cet intervalle fini, seulement un nombre fini de nombres sont représentables exactement, et l'on doit recourir à la troncature ou à l'arrondi pour représenter les autres réels.

Définition 2.2

La précision machine ε_m est la plus grande erreur relative que l'on puisse commettre en représentant un nombre réel sur ordinateur en utilisant la troncature.

Note: La précision machine dépend bien sûr du processeur utilisé et du nombre de bits de la mantisse. Si l'on utilise l'arrondi, la précision machine est tout simplement $\frac{\varepsilon_m}{2}$.

Théorème 2.1

Si b est la base utilisée pour réprésenter un nombre avec n nombre de chiffres (bits si $b = 2$) de la mantisse, alors la précision machine vérifie

$$\varepsilon_m \leq b^{1-n} \tag{1.4}$$

3 Arithmétique flottante

Dans cette section, nous allons décrire la multiplication, l'addition et la division sur une machine travaillant en arithmétique flottante, tout en suivant l'évolution des erreurs au fil de ces opérations élémentaires. Nous utilisons le système décimal, mais les effets décrits valent également pour les autres bases.

Définition 3.1

Soit x, un nombre réel défini sous la forme

$$x = \pm 0, d_1 d_2 d_3 \cdots d_n d_{n+1} \cdots \times 10^l.$$

On note $fl(x)$, sa représentation en virgule flottante à n chiffres définie par

$$fl(x) = \pm 0, d_1 d_2 d_3 \cdots \tilde{d}_n \times 10^l,$$

où la $n^{ième}$ décimale, $\tilde{d}_n$, dépendra de la méthode de remplissage (troncature/arrondi).

Note: La notation flottante d'un nombre dépend du nombre n de chiffres dans la mantisse, mais aussi du procédé retenu pour éliminer les derniers chiffres à savoir la troncature (qui est **biaisée** car si $x \geq 0$, on a $fl(x) \leq x$) ou l'arrondi (qui est non biaisé, on a tour à tour $fl(x) \leq x$ ou $fl(x) \geq x$).

Remarque(s) : La convention IEEE-754 impose l'utilisation de l'arrondi dans la représentation binaire des nombres. Dans les exemples à venir, nous utiliserons l'arrondi.

Exemple 10

Avec $n = 4$ chiffres à la mantisse, on a la représentation en notation flottante de

- $x = \frac{1}{3} \longrightarrow fl(\frac{1}{3}) = 0,3333 \times 10^0$,
- $x = \pi \longrightarrow fl(\pi) = 0,3142 \times 10^1$,
- $x = 12,4551 \longrightarrow fl(12,4551) = 0,1246 \times 10^2$.

3.1 - Opérations élémentaires en arithmétique flottante

Les opérations élémentaires (en l'occurrence l'addition, la soustraction, la multiplication et la division) sont effectuées dans une mémoire auxiliaire (accumulateur) travaillant généralement en double précision.

Méthode : Pour effectuer une opération élémentaire en arithmétique flottante, on doit représenter les deux opérandes en notation flottante, effectuer l'opération de la façon habituelle et exprimer le résultat en notation flottante. Si x et y sont deux nombres réels, les opérations élémentaires en arithmétique flottante s'effectueront par

$$\begin{aligned} x + y &\longrightarrow fl\left(fl(x) + fl(y)\right) \\ x - y &\longrightarrow fl\left(fl(x) - fl(y)\right) \\ x \div y &\longrightarrow fl\left(fl(x) \div fl(y)\right) \\ x \times y &\longrightarrow fl\left(fl(x) \times fl(y)\right) \end{aligned}$$

Exemple 11

- Si l'on prend $n = 4$ chiffres à la mantisse, alors on a l'opération en arithmétique flottante suivante

$$\begin{aligned} \frac{1}{3} \times 3 \longrightarrow fl\left(fl\left(\frac{1}{3}\right) \times fl(3)\right) &= fl\left(\left(0,3333 \times 10^0\right) \times \left(0,3000 \times 10^1\right)\right) \\ &= fl\left(0,09999000 \times 10^1\right) \\ &= 0,9999 \times 10^0 \end{aligned}$$

Note: On remarque une légère perte de précision par rapport à la valeur exacte qui est 1.

- Encore avec $n = 4$ chiffres à la mantisse, on a l'opération en arithmétique flottante suivante

$$\begin{aligned} \left(0,4035 \times 10^6\right) \times \left(0,1978 \times 10^{-1}\right) \longrightarrow & fl\left(fl\left(0,4035 \times 10^6\right) \times fl\left(0,1978 \times 10^{-1}\right)\right) \\ &= fl\left(\left(0,4035 \times 10^6\right) \times \left(0,1978 \times 10^{-1}\right)\right) \\ &= fl\left(0,0798123 \times 10^5\right) \\ &= fl\left(0,798123 \times 10^4\right) \\ &= 0,7981 \times 10^4 \end{aligned}$$

- Toujours avec $n = 4$ chiffres à la mantisse, on effectue l'opération suivante

$$\begin{aligned} \left(0,4035 \times 10^6\right) + \left(0,1978 \times 10^4\right) \longrightarrow & fl\left(fl\left(0,4035 \times 10^6\right) + fl\left(0,1978 \times 10^4\right)\right) \\ &= fl\left(0,4035 \times 10^6 + 0,1978 \times 10^4\right) \\ &= fl\left(0,4035 \times 10^6 + 0,001978 \times 10^6\right) \\ &= fl\left(0,405478 \times 10^6\right) \\ &= 0,4055 \times 10^6 \end{aligned}$$

Note: Par contre, il faut être plus prudent avec l'addition et la soustraction. On ajoute d'abord des zéros à la mantisse du nombre ayant le plus petit exposant de telle sorte que les deux exposants soient égaux. On effectue ensuite l'opération habituelle et l'on met le résultat en notation flottante.

- Toujours avec $n = 4$ chiffres à la mantisse, on effectue l'opération suivante

$$\begin{aligned}(0,56789 \times 10^4) - (0,1234321 \times 10^6) &\longrightarrow fl\left(fl\left(0,56789 \times 10^4\right) - fl\left(0,1234321 \times 10^6\right)\right)\\ &= fl\left(0,5679 \times 10^4 - 0,1234 \times 10^6\right)\\ &= fl\left(0,005679 \times 10^6 - 0,1234 \times 10^6\right)\\ &= -fl\left(0,11772 \times 10^6\right)\\ &= -0,1177 \times 10^6\end{aligned}$$

Note: Il est primordial de décaler la mantisse avant d'effectuer l'addition ou la soustraction.

3.2 - Opérations risquées en arithmétique flottante

Un certain nombre d'opérations élémentaires sont particulièrement sensibles aux erreurs d'arrondi. Nous en présentons deux types de calculs à éviter, dans la mesure du possible.

Exemple 12

- En arithmétique flottante $n = 4$, on obtient

$$\begin{aligned}(0,4000 \times 10^4) + (0,1000 \times 10^{-2}) &\longrightarrow fl\left(fl\left(0,4000 \times 10^4\right) + fl\left(0,1000 \times 10^{-2}\right)\right)\\ &= fl\left(0,4000 \times 10^4 + 0,1000 \times 10^{-2}\right)\\ &= fl\left(0,4000 \times 10^4 + 0,0000001 \times 10^4\right)\\ &= fl\left(0,4000001 \times 10^4\right)\\ &= 0,4000 \times 10^4\end{aligned}$$

Note: Additionner deux nombres dont les ordres de grandeur sont très différents, est une opération dangereuse ! En effet, dans cet exemple la loi des exposants et l'arrondi ont fait de sorte que le petit nombre disparaît complètement devant le plus grand.

- En arithmétique flottante, soustraire deux nombres presque identiques est dangereux

$$\begin{aligned}(0,5678 \times 10^6) - (0,5677 \times 10^6) &\longrightarrow fl\left(fl\left(0,5678 \times 10^6\right) - fl\left(0,5677 \times 10^6\right)\right)\\ &= fl\left(0,5678 \times 10^6 - 0,5677 \times 10^6\right)\\ &= fl\left(0,0001 \times 10^6\right) = 0,1000 \times 10^3\end{aligned}$$

Note: La soustraction de ces 2 nombres de valeur très proche fait apparaître trois 0 non significatifs dans la mantisse du résultat. On appelle ce phénomène l'élimination par soustraction des chiffres significatifs.

Exercice du manuel (fortement suggéré) 2: Donner la représentation en notation flottante en base 10 des nombres suivants (arrondir en conservant 4 chiffres dans la mantisse).

$a)$	e	$b)$	$\frac{1}{6}$	$c)$	$\frac{2}{3}$
$d)$	$12,487 \times 10^5$	$e)$	$213\,456$	$f)$	$2000,1$

Exercice du manuel (fortement suggéré) 3: Montrer que la loi d'associativité de l'addition n'est pas toujours respectée en arithmétique flottante. Utiliser l'arithmétique flottante à 3 chiffres et les nombres suivants : $x = 0,854 \times 10^3$, $y = 0,251 \times 10^3$ et $z = 0,852 \times 10^3$.

4 Erreurs de troncature

L'évaluation efficace d'une fonction $f(x)$ est parfois difficile voire impossible, au regard de la forme complexe que celle-ci peut avoir. Par contre, pour un polynôme $p_n(x)$ de degré n défini par

$$p_n(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + \cdots + a_{n-1}x^{n-1} + a_nx^n, \tag{1.5}$$

son évaluation est bien possible, car impliquant le calcul d'une série de puissances successives de x. De plus, en remarquant que tout polynôme de la forme (1.5) peut s'écrire sous la forme suivante

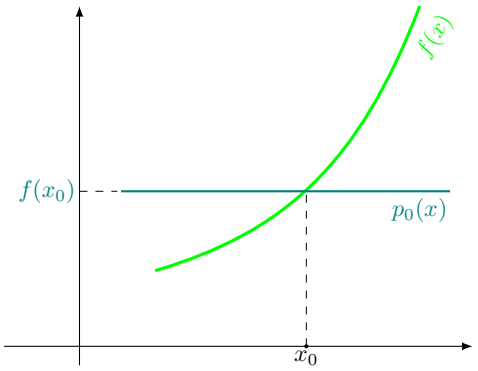
$$\begin{aligned} p_n(x) &= a_0 + x\left(a_1 + a_2x + a_3x^2 + \cdots + a_{n-1}x^{n-2} + a_nx^{n-1}\right) \\ &= a_0 + x\left(a_1 + x\left(a_2 + a_3x + \cdots + a_{n-1}x^{n-3} + a_nx^{n-2}\right)\right) \\ &= a_0 + x\left(a_1 + x\left(a_2 + x\left(a_3 + \cdots + a_{n-1}x^{n-4} + a_nx^{n-3}\right)\right)\right) \\ &\vdots \\ &= a_0 + x\left(a_1 + x\left(a_2 + x\left(a_3 + \cdots + x\left(a_{n-1} + a_nx\right)\cdots\right)\right)\right), \end{aligned}$$

alors l'évaluation sera plus efficace grâce à l'algorithme de Horner [1] avec au plus $2n$ opérations élémentaires, contrairement à la forme (1.5) où $\frac{n(n+1)}{2} + n$ opérations (au plus) doivent être faites.

Il devient ainsi légitime, sachant qu'il n'est pas chose aisée d'évaluer efficacement une fonction $f(x)$, de l'approximer par un polynome dont le degré dépendra de l'information qu'on aura de celle-ci. Cette section sera donc destinée à décrire une forme bien connue d'approximation d'une fonction par un polynôme qui est le développement de Taylor, tout en étant capable de quantifier l'erreur commise dite l'erreur de troncature.

Remarque(s) : Il est important de ne pas confondre la troncature utilisée pour la représentation des nombres sur ordinateur (étudiée précédemment) et les erreurs de troncature qui seront traitées dans cette section.

• 4.1 - Développement de Taylor en une variable

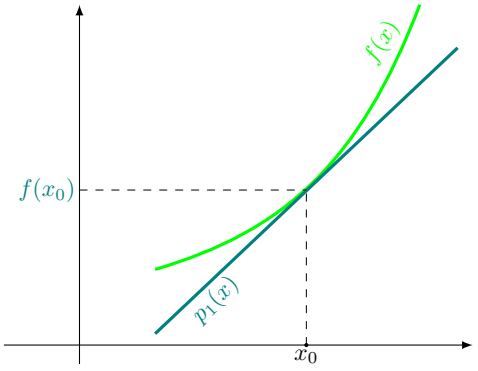


Si $f(x)$ une fonction dont on ne connait que son évaluation au point x_0.

Alors la meilleure approximation polynômiale de $f(x)$, autour de ce point x_0, sera une fonction constante.

C'est à dire un polynôme de degré zero, qu'on va noter $p_0(x)$ et définir au voisinage de x_0 par

$$p_0(x) = f(x_0).$$

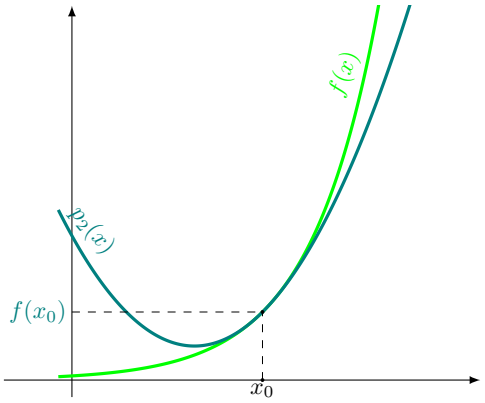


Si on connait la valeur de la fonction $f(x)$ en x_0 ainsi que la pente $f'(x_0)$ en x_0 de cette même fonction.

La meilleur approximation polynômiale de la fonction $f(x)$, au voisinage du point x_0, est un polynôme de degré 1 qui est définie comme suit

$$p_1(x) = f(x_0) + f'(x_0)(x - x_0).$$

1. André Fortin : Analyse numérique pour ingénieurs, Presses Internationales Polytechnique, Montréal, Québec (2016), page 27.



Et si connait de la fonction $f(x)$ sa valeur $f(x_0)$, sa pente $f'(x_0)$ et sa dérivée seconde $f''(x_0)$.

Alors on a un polynôme de degré 2, défini comme suit

$$p_2(x) = f(x_0) + f'(x_0)(x - x_0) + f''(x_0)\frac{(x - x_0)^2}{2},$$

qui est la meilleur approximation polynômiale de la fonction $f(x)$ au voisinage du point x_0.

Note: On peut ainsi continuer cette construction autant de fois que la fonction $f(x)$ sera dérivable en x_0. Le polynôme construit, donne une approximation de $f(x)$ au voisinage de x_0 et est appelé **polynôme de Taylor**.

Définition 4.1

Le polynôme de Taylor de degré n de la fonction $f(x)$ autour de x_0 est défini par

$$p_n(x) = f(x_0) + f'(x_0)(x - x_0) + f''(x_0)\frac{(x - x_0)^2}{2!} + f'''(x_0)\frac{(x - x_0)^3}{3!} + \cdots + f^{(n)}(x_0)\frac{(x - x_0)^n}{n!}, \tag{1.6}$$

où $f^{(n)}(x_0)$ désigne la dérivée d'ordre n de $f(x)$ en x_0.

Théorème 4.1

Soit $f(x)$, une fonction dont les dérivées jusqu'à l'ordre $(n+1)$ existent au voisinage du point x_0. On a l'égalité

$$f(x) = p_n(x) + r_n(x), \tag{1.7}$$

où $p_n(x)$ est le polynôme de Taylor défini en (1.6) et $r_n(x)$ est l'erreur commise en approximant $f(x)$ par $p_n(x)$

$$r_n(x) = f^{(n+1)}(\zeta(x))\frac{(x - x_0)^{n+1}}{(n+1)!}, \tag{1.8}$$

pour un certain $\zeta(x)$ compris entre x_0 et x.

Note: L'erreur $r_n(x)$ est appelée l'erreur de troncature, commise chaque fois que l'on utilise un développement de Taylor pour approximer une fonction. Il n'est possible d'évaluer ce terme d'erreur (car on ne connait pas $\zeta(x)$) mais on cherchera à déterminer sa borne supérieure.

Remarque(s) :

- L'équation (1.7) est une égalité et ne devient une approximation que lorsque le terme d'erreur est négligé.
- Le terme d'erreur de l'équation (1.8) devient de plus en plus grand lorsque x s'éloigne de x_0 en vertu du terme $(x - x_0)^{n+1}$.
- Inversement, pour une valeur de x près de x_0, le terme d'erreur de l'équation (1.8) est de plus en plus petit lorsque n augmente.
- On sait que le point $\zeta(x)$ existe et qu'il varie avec x, mais on ne connaît pas sa valeur exacte. Il n'est donc pas possible d'évaluer le terme d'erreur exactement. On peut tout au plus lui trouver une borne supérieure dans la plupart des cas.

Une forme plus pratique du développement de Taylor est obtenue en posant $h = x - x_0$ ou $x = x_0 + h$:

$$f(x_0 + h) = p_n(h) + r_n(h), \tag{1.9}$$

où le polynôme de Taylor devient

$$p_n(h) = f(x_0) + f'(x_0)h + f''(x_0)\frac{h^2}{2!} + f'''(x_0)\frac{h^3}{3!} + \cdots + f^{(n)}(x_0)\frac{h^n}{n!}, \tag{1.10}$$

et l'erreur de troncature donnée sous la forme

$$r_n(h) = f^{(n+1)}(\zeta(h))\frac{h^{n+1}}{(n+1)!}, \tag{1.11}$$

avec $\zeta(h)$ compris entre x_0 et $x_0 + h$.

Exemple 13

Déterminons une approximation de la fonction $f(x) = e^x$ autour de $x_0 = 0$. Remarquons d'abord que la dérivable à tout ordre n et en tout x de cette fonction est $f^{(n)}(x) = e^x$. En particulier en $x_0 = 0$, sachant que $f^{(n)}(0) = 1$, on a grace à la forme (1.9) du développement de Taylor

$$e^{0+h} = \underbrace{1 + h + \frac{h^2}{2!} + \frac{h^3}{3!} + \cdots + \frac{h^n}{n!}}_{p_n(h)} + \underbrace{e^{\zeta(h)}\frac{h^{n+1}}{(n+1)!}}_{r_n(h)}$$

Ainsi en négligeant le terme d'erreur $r_n(h)$, on a l'approximation suivante de la fonction $f(x)$ autour de $x_0 = 0$

$$e^{0+h} = e^h \simeq p_n(h) = 1 + h + \frac{h^2}{2!} + \frac{h^3}{3!} + \cdots + \frac{h^n}{n!}, \tag{1.12}$$

où l'expression du terme d'erreur, grâce à l'équation (1.11), est donnée par

$$r_n(h) = e^{\zeta(h)}\frac{h^{n+1}}{(n+1)!},$$

et où $\zeta(h) \in [0, h]$. Il n'est pas possible d'évaluer le terme d'erreur mais on peut déterminer sa borne supérieure. La fonction exponentielle étant croissante et puisque $\zeta(h) \leq h$, alors on a

$$e^{\zeta(h)} \leq e^h.$$

On en conclut la borne suivante de l'erreur $r_n(h)$ sur l'intervalle $[0, h]$

$$r_n(h) = e^{\zeta(h)}\frac{h^{n+1}}{(n+1)!} \leq e^h\frac{h^{n+1}}{(n+1)!}. \tag{1.13}$$

Définition 4.2

Une fonction $E(h)$ est dite grand ordre de h^n au voisinage de 0, s'il existe une constante c telle que

$$E(h) \leq ch^n, \text{ au voisinage de } 0 \text{ (c'est à dire si } h \longrightarrow 0).$$

Note: On notera la fonction $E(h)$ comme suit $E(h) = \mathcal{O}(h^n)$.

- Lorsque h est assez petit, la fonction $E(h) = \mathcal{O}(h^n)$ décroît en se comportant comme ch^n.
- Plus n est grand, plus la décroissance est rapide. Ainsi, une fonction $\mathcal{O}(h^3)$ décroît plus vite qu'une fonction $\mathcal{O}(h^2)$, qui elle-même décroît plus vite qu'une fonction $\mathcal{O}(h)$.
- Une fonction $\mathcal{O}(h^n)$ est aussi $\mathcal{O}(h^m)$ si $m < n$.

Question: Comment vérifier (ou déterminer) qu'une fonction $E(h)$ est un grand ordre de h^n au voisinage de 0 ?

Il suffit de faire le rapport de $E(h)$ sur $E(\frac{h}{2})$ et de remarquer qu'on a :

$$\frac{E(h)}{E(\frac{h}{2})} \approx 2^n. \tag{1.14}$$

En effet, si $E(h) \approx ch^n$ alors $E(\frac{h}{2}) \approx c(\frac{h}{2})^n = \frac{ch^n}{2^n}$. Ainsi, on fait le rapport $\frac{E(h)}{E(\frac{h}{2})} \approx ch^n \times \frac{2^n}{ch^n} = 2^n$.

Exemple 14

Utilisons le développement de la relation (1.12), pour estimer la valeur de e^h pout tout h autour de zéro.
À partir de l'égalité (1.9), on a l'expression de l'erreur définie par $r_n(h) = f(x_0 + h) - p_n(h)$.

- Ainsi en prenant $h = 0,1$ et $n = 3$, on a l'erreur en valeur absolue au voisinage de $x_0 = 0$ suivante

$$|r_3(0,1)| = |f(0,1) - p_3(0,1)| = 0,4245 \times 10^{-5}.$$

- D'autre part, pour $h = 0,05$ et $n = 3$, l'erreur absolue au voisinage de $x_0 = 0$ est la suivante

$$|r_3(0,05)| = |f(0,05) - p_3(0,05)| = 0,263 \times 10^{-6}.$$

Le rapport des erreurs absolues liées au polynôme d'approximation $p_3(h)$ est donné ainsi par

$$\frac{|r_3(0,1)|}{|r_3(0,05)|} = \frac{0,4245 \times 10^{-5}}{0,263 \times 10^{-6}} = 16,14 \approx 2^4.$$

Note: La valeur de ce rapport n'est pas fortuite. En effet, face à un polynôme de Taylor de degré $n = 3$, on obtain à partir de l'inégalité (1.13), un terme d'erreur qui est un grand ordre de h^4 au voisinage de 0 (i.e. une erreur de type $\mathcal{O}(h^4)$). Et grâce à la relation (1.14), ce rapport se comporte comme 2^4.

Définition 4.3

Une approximation dont le terme d'erreur est un grand ordre de h^n (noté $\mathcal{O}(h^n)$) est dite d'ordre n.

Remarque(s) : En général le polynôme de Taylor de degré n est une approximation d'ordre $n+1$ car le terme d'erreur est $r_n(h) = \mathcal{O}(h^{n+1})$ (ou $r_n(x) = \mathcal{O}((x - x_0)^{n+1})$) :

$$f(x_0 + h) = p_n(h) + \mathcal{O}(h^{n+1}) \text{ ou } f(x) = p_n(x) + \mathcal{O}((x - x_0)^{n+1}). \tag{1.15}$$

Note: Mais il s'agit d'un ordre minimal, on peut être d'ordre plus élevé : calculer le développement de Taylor d'ordre 5 de $\sin(x)$ autour de $x_0 = 0$.

Note: Ne pas confondre le degré du polynôme de Taylor et son ordre d'approximation. Le degré du polynôme de Taylor est la puissance la plus grande de ce polynôme, alors que l'ordre d'approximation (qui donne la qualité de l'approximation) correspond à la puissance de h sur le terme d'erreur.

Exercice du manuel (fortement suggéré) 4: Effectuer les développements de Taylor suivants à l'ordre demandé. Utiliser la forme de l'équation (1.10). Donner l'expression analytique du terme d'erreur. Donner également une borne supérieure de l'erreur lorsque c'est possible.

a) $cos(x)$ autour de $x_0 = 0$ (ordre 8)

b) $sin(x)$ autour de $x_0 = 0$ (ordre 9)
c) $arctan(x)$ autour de $x_0 = 0$ (ordre 5)
d) $cos(x)$ autour de $x_0 = \frac{\pi}{2}$ (ordre 7)
e) $sin(x)$ autour de $x_0 = \frac{\pi}{2}$ (ordre 8)

Exercice du manuel (fortement suggéré) 5:

a) Calculer le développement de Taylor d'ordre 5, c'est-à-dire dont le terme d'erreur est de type $\mathcal{O}(h^5)$, de la fonction $f(x) = ln(x)$ autour de $x_0 = 1$. Donner l'expression analytique du terme d'erreur.
b) À l'aide de ce développement, donner une approximation de $ln(1,1)$. Par comparaison avec la valeur exacte $(ln(1,1) = 0,0953101798)$, donner le nombre de chiffres significatifs de l'approximation.
c) Par quel facteur approximatif l'erreur obtenue en b) serait-elle réduite si l'on évaluait $ln(1,025)$ au moyen du développement de Taylor obtenu en a) ? (Ne pas faire les calculs.)

4.2 - Développement de Taylor en plusieurs variables

On se limite, pour les fins de l'exposé, à un développement de Taylor en deux variables et d'ordre 2 ou 3.

Théorème 4.2

Soit $f(x, y)$ une fonction de deux variables, à valeurs réelles que l'on suppose suffisamment différentiable.

- Le développement de Taylor d'ordre 2 de la fonction $f(x, y)$ autour du point (x_0, y_0) est défini par

$$f(x_0 + h_1, y_0 + h_2) = f(x_0, y_0) + \frac{\partial f}{\partial x}(x_0, y_0)h_1 + \frac{\partial f}{\partial y}(x_0, y_0)h_2 + r_1(h_1, h_2), \tag{1.16}$$

où le terme d'erreur est donné par

$$r_1(h_1, h_2) = \frac{1}{2}\frac{\partial^2 f}{\partial x^2}(\zeta, \eta)h_1^2 + \frac{\partial^2 f}{\partial x \partial y}(\zeta, \eta)h_1 h_2 + \frac{1}{2}\frac{\partial^2 f}{\partial y^2}(\zeta, \eta)h_2^2, \tag{1.17}$$

pour un certain point (ζ, η) sur le segment droit délimité par (x_0, y_0) et $(x_0 + h_1, y_0 + h_2)$.

Note: Le polynôme de Taylor de la fonction $f(x, y)$ autour de (x_0, y_0) est ainsi défini par

$$p_1(h_1, h_2) = f(x_0, y_0) + \frac{\partial f}{\partial x}(x_0, y_0)h_1 + \frac{\partial f}{\partial y}(x_0, y_0)h_2 \tag{1.18}$$

- Le développement de Taylor d'ordre 3 de la fonction $f(x, y)$ autour du point (x_0, y_0) est défini par

$$\begin{aligned} f(x_0 + h_1, y_0 + h_2) = f(x_0, y_0) &+ \frac{\partial f}{\partial x}(x_0, y_0)h_1 + \frac{\partial f}{\partial y}(x_0, y_0)h_2 \\ &+ \frac{1}{2}\frac{\partial^2 f}{\partial x^2}(x_0, y_0)h_1^2 + \frac{\partial^2 f}{\partial x \partial y}(x_0, y_0)h_1 h_2 + \frac{1}{2}\frac{\partial^2 f}{\partial y^2}(x_0, y_0)h_2^2 \\ &+ r_2(h_1, h_2) \end{aligned} \tag{1.19}$$

où $r_2(h_1, h_2)$ est le terme d'erreur.

Note: Si $f(x, y, z)$ est de trois variables, on a le développement de Taylor d'ordre 2 autour du point (x_0, y_0, z_0)

$$\begin{aligned} f(x_0 + h_1, y_0 + h_2, z_0 + h_3) = f(x_0, y_0, z_0) &+ \frac{\partial f}{\partial x}(x_0, y_0, z_0)h_1 + \frac{\partial f}{\partial y}(x_0, y_0, z_0)h_2 + \frac{\partial f}{\partial z}(x_0, y_0, z_0)h_3 \\ &+ \mathcal{O}(h_1^2) + \mathcal{O}(h_2^2) + \mathcal{O}(h_3^2), \end{aligned}$$

Exemple 15

Soit la fonction réelle $f(x,y)$ de deux variables définie par

$$f(x,y) = x^2 + x\sin(y). \tag{1.20}$$

Pour effectuer son développement de Taylor de degré 2 au point $(x_0,y_0)=(1,0)$, on doit au regard de la formule (1.19), évaluer $f(x,y)$, ses dérivées premières

$$\frac{\partial f}{\partial x}(x,y) = 2x + \sin(y), \quad \frac{\partial f}{\partial y}(x,y) = x\cos(y),$$

et ses dérivées secondes

$$\frac{\partial^2 f}{\partial x^2}(x,y) = 2, \quad \frac{\partial^2 f}{\partial y^2}(x,y) = -x\sin(y), \quad \frac{\partial^2 f}{\partial x \partial y}(x,y) = \cos(y),$$

Ainsi on a

$$\frac{\partial f}{\partial x}(1,0) = 2, \quad \frac{\partial f}{\partial y}(1,0) = 1, \quad \frac{\partial^2 f}{\partial y^2}(1,0) = 0 \text{ et } \frac{\partial^2 f}{\partial x \partial y}(x,y) = 1.$$

Et par ce développement de Taylor de degré 2 de la fonction $f(x,y)$ autour du point $(1,0)$, on a l'approximation suivante

$$\begin{aligned} f(1+h_1, 0+h_2) &\simeq \underbrace{1 + 2h_1 + 1h_2 + \frac{1}{2}(2h_1^2 + 0h_2^2) + (1h_1h_2)}_{p_1(h_1,h_2)} \\ &\simeq 1 + 2h_1 + h_2 + h_1^2 + h_1h_2. \end{aligned}$$

- En choisissant par exemple $h_1 = h_2 = 0,1$, on obtient l'approximation

$$f(1+0,1\,,0,1) \simeq 1,32$$

  qui est proche de la valeur exacte $1,319816758$ avec une erreur absolue d'environ $r_2(0,1;0,1) = 0,000183$.
- D'autre part, si on prend la moitié du pas précédent, $h_1 = h_2 = 0,05$, on a une approximation de $f(1,05,0,05)$ qui vaut $1,155$ et dont l'erreur absolue vaut $r_2(0,05;0,05) = 0,000021825$.

Cette dernière est $8,4$ fois plus petit que la précédente. Ce facteur de 8 s'explique par le choix d'un développement de degré 2 (et d'ordre 3).

4.3 - Propagation d'erreurs dans le cas général

Question: Pour une quantité inconnue x mais approchée par une valeur approximative x^*, Que peut-on dire de la précision de l'approximation de la valeur inconnue $f(x)$ par $f(x^*)$? En d'autres termes, si l'on a $x = x^* + \Delta x$, que serait l'erreur absolue $\Delta f = |f(x) - f(x^*)|$?

Pour répondre à cette question, on applique le développement de Taylor autour de x^*

$$f(x) = f(x^* + \Delta x) = f(x^*) \pm f'(x^*)\Delta x + \mathcal{O}\big((\Delta x)^2\big)$$

Note: En négligeant le termes d'erreur, on obtient l'estimation suivante

$$\Delta f = |f(x) - f(x^*)| \simeq f'(x^*)\Delta x \tag{1.21}$$

Remarque(s) : En pratique, si $f(x,y)$ est une fonction de deux variables x et y, elles-mêmes approchées par x^* et y^* avec une précision de Δx et Δy, respectivement, alors l'erreur absolue Δf est estimée par

$$\Delta f = |f(x,y) - f(x^*,y^*)| \simeq \left|\frac{\partial f}{\partial x}(x^*,y^*)\right| \Delta x + \left|\frac{\partial f}{\partial y}(x^*,y^*)\right| \Delta y \tag{1.22}$$

Pour une fonction de trois variables, $f(x,y,z)$, on a l'approximation suivante

$$\Delta f = |f(x,y,z) - f(x^*,y^*,z^*)| \simeq \left|\frac{\partial f}{\partial x}(x^*,y^*,z^*)\right| \Delta x + \left|\frac{\partial f}{\partial y}(x^*,y^*,z^*)\right| \Delta y + \left|\frac{\partial f}{\partial z}(x^*,y^*,z^*)\right| \Delta z$$

Exemple 16

On a mesuré la longueur d'un côté d'une boîte cubique et obtenu $l^* = 10,2\ cm$, avec une précision de l'ordre du millimètre ($\Delta l = 0,1\ cm$). Le volume v de cette boîte est déterminé par l'expression analytique $v(l) = l^3$. Ainsi, la valeur approximative du volume est $(10,2)^3 = 1061,2\ cm^3$.
D'autre part, grâce à la formule (1.21), l'erreur liée au calcul de ce volume est estimée comme suit

$$\Delta v \simeq |v'(l^*)|\Delta l = 3(10,2)^2 \times 0,1 = 31,212 \leq 0,5 \times 10^2.$$

Donc seuls les deux premiers chiffres du volume $v = 1061,2\ cm^3$ sont significatifs.

Note: De l'équation (1.22), on déduit la façon dont se propagent les erreurs dans les opérations élémentaires :

$$\begin{aligned}
f(x,y) = x + y \quad &\longrightarrow \quad \Delta f \simeq |\Delta x| + |\Delta y| \\
f(x,y) = x - y \quad &\longrightarrow \quad \Delta f \simeq |\Delta x| + |\Delta y| \\
f(x,y) = x \times y \quad &\longrightarrow \quad \Delta f \simeq |y^*||\Delta x| + |x^*||\Delta y| \\
f(x,y) = x \div y \quad &\longrightarrow \quad \Delta f \simeq \frac{|y^*||\Delta x| + |x^*||\Delta y|}{|y^*|^2}
\end{aligned}$$

Exercice du manuel (fortement suggéré) 6: Dans la revue Science and Vie de septembre 1996, on fournit les données suivantes pour le satellite Titania d'Uranus :

$$\begin{aligned}
Rayon: \quad & R = 800000 \pm 5000\, m \\
Densité: \quad & \rho = 1590 \pm 90\, kg/m^3
\end{aligned}$$

a) Donner le nombre de chiffres significatifs du rayon R et de la densité ρ.

b) En supposant que Titania soit parfaitement sphérique (de volume $V = \frac{4\pi R^3}{3}$), trouver une approximation de la masse de Titania et donner le nombre de chiffres significatifs de votre résultat.

Exercices suggérés du manuel !

- Exercices [a] suggérés : 1, 8 - 13, 15, 19 - 25, 31, 35, 37
- Exercices fortement suggérés : 1, 9, 20, 24, 31, 37

a. André Fortin : Analyse numérique pour ingénieurs, Presses Internationales Polytechnique 2016.

À Retenir !

Je dois pouvoir répondre aux questions [a] suivantes :

1. calculer/estimer l'erreur absolue et relative d'un nombre
2. établir les chiffres significatifs partant de l'erreur absolue et réciproquement.
3. faire une représentation en virgule flottante avec mantisse à n chiffres par troncature ou par arrondi
4. normaliser une mantisse et décaler la mantisse le cas échéant
5. faire des opérations élémentaires en virgurle flottante à n chiffres
6. reconnaitre les opérations dangereuses et comprendre l'importance de l'ordre dans les opérations élémentaires
7. utiliser la méthode de Horner
8. construire un polynôme de Taylor de degré n ainsi que son terme d'erreur

9. estimer le reste, approximer l'erreur absolue et obtenir le nombre de chiffres significatifs de mon approximations
10. comprendre et savoir calculer l'ordre d'une méthode
11. savoir calculer la propagation d'erreur pour une fonction d'une ou de plusieurs variables et ainsi calculer l'erreur/chiffres significatifs d'une évaluation de fonction

a. André Fortin : Analyse numérique pour ingénieurs, Presses Internationales Polytechnique, Québec 2016, p.1-50.