ANALYSE NUMERIQUE 1

Programme :

Chap.1 : Introduction à l’analyse numérique.

Chap.2 : Résolution d’équations non linéaires.

Chap.3 : Résolution des systèmes linéaires.

Introduction à l’Analyse Numérique

1- Introduction :

L’Analyse Numérique consiste en la conception et l’étude d’algorithmes de résolution numérique de problèmes mathématiques. Ces problèmes mathématiques proviennent pour la plupart de la modélisation de phénomènes ou comportements issus du monde réel, au sein de disciplines aussi variées que la physique, la biologie, ou bien encore l’économie.

On ne peut pas résoudre analytiquement tous les problèmes mathématiques issus de ces disciplines. Dans ces cas, on remplace la résolution mathématique de ces problèmes par une approximation numérique en utilisant les techniques d’analyse numérique. L’analyse numérique permet d’obtenir au moyen d’un algorithme la solution approchée d’un problème mathématique posé et ceci avec une précision désirée et après un nombre fini d’opérations élémentaires. Les résultats finaux et leurs degrés de précision vont dépendre des algorithmes utilisés et de leur mise en œuvre sur ordinateur.

La notion fondamentale autour de laquelle s’articule l’Analyse Numérique est la notion d’erreur. Le terme erreur désigne ici la différence entre le résultat numérique d’un problème et la valeur théorique ou réelle que l’on est censé obtenir.

Ces erreurs peuvent avoir plusieurs origines.

On distingue :

- les erreurs de modélisation, liées au fait que la modélisation mathématique d’un problème repose sur un certain nombre d’hypothèses et d’approximations, permettant de simplifier le modèle final,

- les erreurs de données, liées à l’imprécision des mesures physiques ou des calculs permettant de donner des valeurs aux différents paramètres d’un modèle,

- les erreurs de discrétisation, liées à la transformation du problème mathématique continu en un problème numérique discret, permettant de calculer la solution d'un problème en un nombre fini de points,

- les erreurs d'arrondi, liées au fait que les ordinateurs ne peuvent représenter les réels qu'avec un nombre fini de chiffres.

Exemple d'erreurs :

$$e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!} \cong \sum_{n=0}^{100} \frac{x^n}{n!}$$

$$(0.1)_{10} = (0.0001100110011)_2$$

2- Notion d'erreur

Soit x, un nombre représentant une solution théorique d'un problème (1, 2, 5, π, $\sqrt{2}$, $\frac{1}{6}$), et la résolution numérique de celui-ci conduit à une valeur approchée notée x^* (3.14, 0.1666666, 1.4142). Il existe deux manières de quantifier l'erreur commise entre la valeur théorique et la valeur numérique :

- l'erreur absolue, calculée en prenant la valeur absolue de la différence entre x et x^*,

$$\Delta x = |x - x^*| \qquad (1)$$

- l'erreur relative, calculée en divisant l'erreur absolue par la valeur absolue de la valeur théorique x.

$$E_r = \frac{|x-x^*|}{|x|} = \frac{\Delta x}{|x|} \qquad (2)$$

De plus, en multipliant par 100 %, on obtient l'erreur relative en pourcentage

On remarquera que l'erreur absolue possède une unité physique (mètres, grammes...) qui est la même que celle de x et x^*, tandis que l'erreur relative est sans unité.

Parmi ces deux erreurs, seule l'erreur relative permet d'estimer correctement la qualité de l'approximation numérique x^* par rapport à la valeur théorique x.

Exemple d'un chronomètre dont la précision est limitée au dixième de seconde $\frac{1}{10}$.

L'erreur absolue entre la durée réelle T d'une course et sa durée chronométrée T^* est donc au maximum d'un ΔT=0,1 de seconde, et ce indépendamment de la course considérée.

Ce n'est pas le cas de l'erreur relative.

Dans le cas d'un marathon couru en approximativement 2 heures 20, l'erreur relative de chronométrage sera au maximum d'environ :

$$E_r = 1{,}19 \times 10^{-5},$$

Soit une valeur suffisamment faible pour permettre a priori de classer sans problème tous les concurrents.

En revanche, dans le cas d'un 100 mètres couru en 10 secondes, l'erreur relative sera de l'ordre de : $E_r = 0{,}01$ soit 1%

Ce qui pourrait s'avérer très problématique dans le cas fréquent d'une arrivée serrée.

En pratique, il est difficile d'évaluer les erreurs absolue et relative, car on ne connaît généralement pas la valeur exacte de x et on n'a que x^*.

Dans le cas de quantités mesurées expérimentalement dont on ne connaît que la valeur approximative, on dispose souvent d'une borne supérieure pour l'erreur absolue qui dépend de la précision de l'instrument de mesure utilisé.

Alors on écrit : $$|x - x^*| \leq \Delta x \qquad (3)$$

ce qui peut également s'écrire $$x^* - \Delta x \leq x \leq x^* + \Delta x \qquad (4)$$

ou $$x = x^* \mp \Delta x \qquad (5)$$

On peut interpréter 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.

L'erreur absolue donne une mesure quantitative de l'erreur commise et l'erreur relative en mesure l'importance.

3- Ecriture des nombres dans les différentes bases :

Pour tout réel x, on sait qu'il existe un entier relatif N tel que x est compris entre 10^N et 10^{N+1}, $(10^N \leq x \leq 10^{N+1})$ et une suite de chiffres a_i compris entre 0 et 9 tels que x s'écrive comme la somme des $a_i * 10^{N-i}$ pour i allant de 0 à éventuellement l'infini (si le nombre de décimales de x est infini par exemple).

$$x = a_i \times 10^{N-i}$$

Exemple : prenons $x = 1.25$, son écriture décimale vient du fait qu'il se décompose en base 10 de la manière suivante :

$$1.25 = 1 \times 10^0 + 2 \times 10^{-1} + 5 \times 10^{-2}.$$

Cette représentation, dite décimale à virgule fixe, est celle que nous utilisons dans la vie de tous les jours.

4- Chiffres significatifs

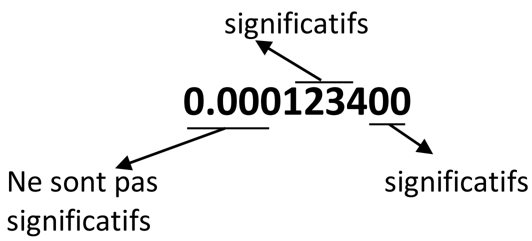
Un chiffre significatif d'un nombre approché est le seul chiffre qu'on doit garder, c'est à dire tout chiffre dans sa représentation décimale différent du zéro ; et un zéro qui se trouve entre deux chiffres, ou il constitue un chiffre conservé.

Exemple : Une approximation à 5 décimales de 0.02010 est :

$\underline{0.0}2010$ Les zéros soulignés ne sont pas significatifs car ils ne servent qu'à indiquer les ranges des autres chiffres.

$0.02\underline{0}10$ Le zéro souligné étant placé entre les chiffres significatifs 2 et 1, zéro est lui même un chiffre significatif.

$0.0201\underline{0}$ Le zéro souligné traduit le fait que le nombre approché a conservé la décimale 10^{-5}, c'est un chiffre significatif.



Un chiffre significatif d'un nombre approché x^* est dit exact si l'erreur absolue de x^* vérifie :

$$\Delta x \leq 0.5 \times 10^{-m} \qquad (5)$$

Alors le chiffre 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. Inversement, si un nombre est donné avec *n* chiffres significatifs, on commence à compter à partir du premier chiffre non nul à gauche et l'erreur absolue est inférieure à 0,5 puissance de 10 correspondant au dernier chiffre significatif.

Par exemple, on obtient une approximation de π (x = π) au moyen de la quantité 22/7 ($x^* = 22/7 = 3{,}142857 \cdots$).

On en conclut que :

$$\Delta x = |\pi - 22/7| = 0.00126 \ldots \cong 0.126 * 10^{-2}$$

Puisque l'erreur absolue est plus petite que $0{,}5 \times 10^{-2}$, le chiffre des centièmes est significatif et on a en tout 3 chiffres significatifs (3.14).

Si l'on retient 3.1416 comme approximation de π, on a :

$$\Delta x = |\pi - 3.1416| \cong 0.73 * 10^{-5}$$

et l'erreur absolue est inférieure à $0{,}5 \times 10^{-4}$. Le chiffre correspondant à cette puissance de 10 (6) est significatif au sens de la définition, ainsi que tous les chiffres situés à sa gauche. Il est à remarquer que le chiffre 6 dans 3,1416 est significatif même si la quatrième décimale de π est un 5 (3,141 59 · · ·). L'approximation 3,1416 possède donc 5 chiffres significatifs

5- Représentations des nombres réels dans une machine :

La première chose à comprendre est que, du fait du caractère fini de la mémoire d'un ordinateur, on ne peut pas représenter tous les nombres réels de manière exacte. En effet, en admettant que l'on travaille en base 10, il faudrait une mémoire de taille infinie pour représenter exactement le nombre π, du fait du nombre infini de ses décimales. Les nombres réels seront donc, pour une bonne partie d'entre eux, représentés de manière approchée.

Le principal mode de représentation des nombres réels en machine est la représentation en virgule flottante normalisée. A tout réel x on associe sa représentation en machine notée $fl(x)$ par l'expression suivante : $fl(x) = \pm m.b^e$ où b est la base de la représentation, m est appelé la mantisse et e l'exposant.

- La plupart des machines utilisent la base 2 pour représenter les réels, afin d'utiliser les unités de stockage binaires.

- La mantisse *m* est un réel compris entre 1 et *b* strictement. Cette mantisse sera stockée en mémoire en utilisant son écriture en base *b*.

- La mantisse ne peut avoir plus de *p* chiffres en base *b*, *p* étant un entier positif fixé par le format utilisé.

La représentation d'un nombre réel en machine peut se faire de deux manières différentes :

- Simple précision, on a $p = 24$ et une taille de stockage de 32 bits pour chaque réel.

- double précision, ces valeurs sont plus élevées : *p* est égal à 53 et une taille de stockage de 64 bits.

6- Conditionnement et Stabilité

6-1 Conditionnement

Le conditionnement décrit la sensibilité de la valeur d'une fonction à une petite variation de son argument, c'est-à-dire :

$$\frac{f(x)-f(x^*)}{f(x)} \text{ en fonction de } \frac{x-x^*}{x}$$

Lorsque $x-x^*$ est petit. Pour une fonction suffisamment régulière, on a évidemment :

$$\left|\frac{f(x)-f(x^*)}{f(x)} \Big/ \frac{x-x^*}{x}\right| \cong \left|\frac{xf'(x)}{f(x)}\right|$$

Donc on appelle conditionnement d'une fonction numérique *f* en un point *x*, le nombre : $cond(f)_x = \left|\frac{xf'(x)}{f(x)}\right|$

Exemple : $f(x) = \sqrt{x}$

$$\left|\frac{xf'(x)}{f(x)}\right| = \frac{1}{2}$$

Ceci correspond à un bon conditionnement, puisque l'erreur relative sur *f* sera au plus moitié d'une erreur relative sur *x*.

$f(x) = a - x$

$$\left|\frac{xf'(x)}{f(x)}\right| = \left|\frac{x}{a-x}\right|$$

Ici, le conditionnement est très mauvais si *x* est voisin de *a*.

6-2 Stabilité

La stabilité décrit la sensibilité d'un algorithme numérique pour le calcul d'une fonction *f(x)*.

Exemple : $f(x) = \sqrt{x+1} - \sqrt{x}$

$$\left|\frac{xf'(x)}{f(x)}\right| = \frac{1}{2}\sqrt{\frac{x}{x+1}}$$

Cette dernière expression étant proche de *1/2* pour *x* grand. Donc, si *x* est grand, le conditionnement de *f* est bon. Prenons un nombre quelconque :

$f(12345) = \sqrt{12346} - \sqrt{12345}$=111,113-111,108=0,50000x10^{-2} .

Alors qu'avec une calculatrice on trouve $f(12345) = 0{,}45000 \times 10^{-2}$ Donc on a une erreur de 10%. Ce qui est important et peu en accord avec le bon conditionnement de *f*. Ceci est dû à l'algorithme utilisé dans ce calcul.

Algorithme :

Un algorithme est un procédé de calcul qui ne met en œuvre que les opérations arithmétiques et logiques.

Méthodes exactes et les Méthodes approchées : Les méthodes exactes gagnent en l'optimalité des solutions et perdent en temps d'exécution, ce qui est l'inverse pour les méthodes approchées qui ne garantissent pas de trouver une solution exacte, mais seulement une approximation en des temps raisonnables de calculs.

Méthodes itératives : elles consistent en l'approche de la solution exacte par une suite. de solutions approchées.