

MT09 - Analyse numérique

Chapitre 0 - Qu'est-ce que l'analyse numérique ?

Laboratoire de Mathématiques Appliquées de Compiègne

© *Draft date September 10, 2013*

Sommaire

1	Introduction	2
2	Effet des erreurs d'arrondi : Exemple : Calcul d'une suite	3
2.1	Calcul par récurrence : première méthode	3
2.2	Calcul par récurrence : deuxième méthode	4
2.3	Calcul par intégration numérique	4
3	Discrétisation d'un problème continu	5
3.1	Dérivée première : définition	5
3.2	Formule de Taylor	5
3.3	Dérivée première : première approximation	5
3.4	Dérivée première : deuxième approximation	6
3.5	Dérivée première : troisième approximation	6
3.6	Ordre de convergence	6
3.7	Dérivée seconde	6
3.8	Équation de la chaleur	7
A	Fonction scilab : suite récurrente	8
B	Fonction scilab : dérivation numérique	9
C	Fonction scilab : dérivation numérique (suite)	9

1 Introduction

L'analyse numérique est une discipline carrefour, ce qui en fait l'intérêt et la difficulté.

1. L'étude et la résolution d'un problème d'analyse numérique en vraie grandeur passe par plusieurs phases :
 - (a) Le problème provient tout d'abord en général, de la mécanique, de la physique, des sciences de l'ingénieur. Il y a ainsi un travail de modélisation, se traduisant par une mise en équations. Le plus souvent cette modélisation est non triviale. Elle est faite, ou du moins engagée par celui qui pose le problème (physicien, mécanicien, ingénieur).
 - (b) L'étape précédente conduit à un modèle. C'est alors le travail du mathématicien d'en étudier la cohérence. Il propose un cadre fonctionnel adéquat et s'efforce de montrer que le problème posé, ou du moins la traduction mathématique qui en a été construite, admet une solution unique. Malheureusement, dans de nombreuses situations réelles, cette étude ne peut être menée à son terme, soit du fait de la difficulté mathématique soit du fait du manque de temps ou de moyens. Ainsi, l'étude mathématique de certaines équations de la mécanique des fluides connues depuis plusieurs siècles, est loin d'être achevée.
 - (c) Le pas suivant consiste à définir une approximation du modèle, afin de permettre une résolution numérique du problème posé. Là commence le travail proprement dit de l'analyste numérique. Il lui faut construire un algorithme de calcul et, lorsque faire ce peut, démontrer que l'algorithme définit bien une solution approchée du problème.
 - (d) Enfin, il reste à écrire un logiciel implantant l'algorithme et à valider les résultats fournis en bout de chaîne par la machine.
2. Tout au long de ce processus, les sources d'erreur sont multiples :
 - (a) Le modèle construit a-t-il pris en compte les phénomènes physiques essentiels, ne négligeant que ce qui est négligeable ?
 - (b) Le modèle construit est-il mathématiquement cohérent ? Définit-il une seule solution ? En admet-il au moins une ?
 - (c) L'approximation est-elle bonne ? Quelle est la sensibilité du modèle aux erreurs ?
 - (d) Le logiciel d'implantation est-il correct ? Dans un logiciel de plusieurs dizaines, voire centaines, de milliers d'instructions, il y a toujours des erreurs de programmation.
 - (e) Enfin chaque opération arithmétique est effectuée avec une erreur d'arrondi. Quand un super calculateur en effectue des milliards à la seconde et mouline, disons vingt minutes, on peut légitimement craindre que cela ne donne lieu à une accumulation catastrophique, suffisante à elle seule à enlever toute validité aux résultats.
3. On voit ainsi l'intérêt essentiel d'une **validation des résultats**. On voit aussi que l'on n'aura jamais de certitudes, mais seulement au mieux un faisceau de présomptions de validité. Les méthodes à utiliser peuvent se classer sommairement de la manière suivante :

- (a) Étude des propriétés qualitatives de la solution calculée et de sa vraisemblance physique. Ici intervient de manière essentielle le physicien qui saura à quoi doit ressembler la solution cherchée (on sait par exemple que les contraintes mécaniques dans un solide deviennent très grandes près des arêtes).
- (b) Calcul de solutions analytiques, dans des cas particuliers. Avant que les ordinateurs ne deviennent disponibles, tout un arsenal de fonctions spéciales (fonctions de Bessel, harmoniques sphériques, polynômes orthogonaux) avait été construit pour exprimer les solutions de problèmes aux limites dans le cas de géométries simples. Cet arsenal est précieux et doit être utilisé chaque fois que possible.
- (c) On peut souvent se donner une solution arbitraire, construire un jeu de données, puis lancer le logiciel sur ce jeu de données et voir si l'on retrouve la solution de départ.
- (d) Comparaison de ses résultats à ceux fournis par d'autres logiciels résolvant le même problème par d'autres méthodes.

2 Effet des erreurs d'arrondi : Exemple : Calcul d'une suite

Nous allons voir sur cet exemple qu'un algorithme a priori naturel et raisonnable n'est pas toujours un bon algorithme. Nous allons mettre ainsi en évidence qu'il faut par tous les moyens disponibles vérifier la validité des résultats obtenus.

$$I_n = \int_0^1 x^n e^{-x} dx.$$

2.1 Calcul par récurrence : première méthode

Soit donc à calculer l'intégrale

$$I_n = \int_0^1 x^n e^{-x} dx$$

La méthode analytique standard consiste à obtenir une relation de récurrence entre I_n et I_{n-1} . Pour ce faire, on fait une **intégration par parties**, soit

$$\int_0^1 x^n e^{-x} dx = x^n (-e^{-x}) \Big|_0^1 - \int_0^1 n x^{n-1} (-e^{-x}) dx$$

d'où découle la relation de récurrence :

$$I_n = -\frac{1}{e} + n I_{n-1}$$

En partant de $I_0 = 1 - 1/e$, on calcule alors successivement I_1, I_2, \dots, I_n . Il est donc naturel de commencer par utiliser la même méthode pour un calcul numérique. Qu'observe-t-on alors ? On voit tout d'abord que les valeurs de I_n ainsi calculées sont décroissantes pour les premières valeurs de n , pour ensuite se mettre à croître violemment. On peut voir aussi, si l'on dispose de plusieurs ordinateurs ou de plusieurs calculatrices, que les résultats obtenus sont très vite très différents d'un ordinateur ou d'une calculatrice à l'autre.

Il y a donc un problème. Que peut-on alors dire mathématiquement sur le comportement que doivent avoir les valeurs de I_n lorsque n tend vers l'infini ? On peut obtenir la majoration suivante :

$$|I_n| \leq \int_0^1 x^n dx = \frac{1}{n+1} \xrightarrow{n \rightarrow \infty} 0.$$

et comme l'on voit qu'en fait les valeurs de I_n obtenues numériquement par notre méthode numérique deviennent très grandes, cela signifie qu'elles sont complètement fausses.

Que se passe-t-il donc ? En examinant de plus près la relation de récurrence ci-dessus, nous voyons qu'au moment de calculer I_n , l'erreur commise sur I_{n-1} est multipliée par n . Ainsi quand on arrive au calcul de I_n , l'erreur commise sur I_0 a été multipliée par $n!$. Cette erreur initiale a beau être très petite, multipliée par $n!$, elle devient vite catastrophique. C'est ce que l'on observe effectivement.

2.2 Calcul par récurrence : deuxième méthode

Que faire alors ? On utilise le fait que I_n tend vers 0 quand n tend vers l'infini. On prend une valeur $N \gg n$, n valeur pour laquelle on veut calculer I_n , et l'on part de $I_N = 0$, pour utiliser ensuite la récurrence à l'envers, calculant ainsi successivement I_{N-1} , I_{N-2}, \dots, I_{n+1} et enfin I_n . Comme la relation de récurrence à l'envers s'écrit

$$I_{n-1} = \frac{1}{n} \left(\frac{1}{e} + I_n \right),$$

nous voyons que l'erreur initiale sur I_N est divisée successivement par $N(N-1)(N-2) \dots (n+1)$.

On voit alors, en calculant I_n par cette deuxième méthode, que les valeurs de I_n obtenues tendent bien vers 0 quand n tend vers l'infini. On observe en outre, que d'un ordinateur à l'autre les valeurs obtenues sont bien les mêmes, sauf pour les dernières décimales éventuellement.

2.3 Calcul par intégration numérique

Pour valider de manière encore plus satisfaisante les valeurs de I_n obtenues par la méthode précédente, nous allons calculer à nouveau ces valeurs, par une méthode complètement différente : l'intégration numérique.

La valeur de I_n mesure l'aire de la surface sous-tendue par la courbe $f_n(x) = x^n e^{-x}$ entre les droites $x = 0$, $x = 1$ et l'axe des x . Étant donné l'intervalle $[0, 1]$ de la droite réelle, introduisons une subdivision

$$x_0 = 0 < x_1 < \dots < x_i < x_{i+1} < \dots < x_N = 1$$

de cet intervalle. On approche l'intégrale de f_n sur chaque sous-intervalle $[x_i, x_{i+1}]$ par l'aire du trapèze correspondant : bases $f_n(x_i)$ et $f_n(x_{i+1})$, hauteur $x_{i+1} - x_i$, soit

$$\frac{f_n(x_i) + f_n(x_{i+1})}{2} (x_{i+1} - x_i)$$

et donc la valeur approchée de l'aire totale :

$$\int_0^1 f(x) dx \simeq \sum_{i=0}^{N-1} \frac{f_n(x_i) + f_n(x_{i+1})}{2} (x_{i+1} - x_i)$$

ce qui s'écrit aussi

$$\int_a^b f_n(x) dx \simeq \frac{f_n(a)}{2} + \sum_{i=1}^{N-2} f_n(x_i)(x_{i+1} - x_i) + \frac{f_n(b)}{2}$$

Pour une subdivision suffisamment fine de $[0, 1]$, nous pouvons raisonnablement espérer obtenir une valeur approchée de I_n .

On constate alors que l'on retrouve ainsi les valeurs de I_n obtenues par la deuxième méthode basée sur la relation de récurrence. Les fonctions scilab de l'appendice (A) permettent d'observer tout ce qui vient d'être dit.

3 Discrétisation d'un problème continu

Exemple : Dérivation numérique et application

3.1 Dérivée première : définition

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h}$$

Cette définition nous conduit à penser, que pour h suffisamment petit, le quotient dont la limite est $f'(x)$, en est une bonne approximation.

3.2 Formule de Taylor

C'est un outil fondamental pour toutes les estimations d'*erreurs locales*.

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2!}f''(x) + \dots + \frac{h^n}{n!}f^{(n)}(x) + \frac{h^{n+1}}{(n+1)!}f^{(n+1)}(x+\theta h), \theta \in]0, 1[.$$

où θ est un nombre inconnu dépendant de x et h .

Cette formule, valable lorsque toutes les dérivées utilisées existent et sont continues, nous sera utile pour h petit.

Nous allons maintenant construire successivement 3 formules d'approximation de la dérivée première. Ces 3 expressions vont être construites en utilisant la formule de Taylor. La première est effectivement celle dont la limite sert de définition de la dérivée. La limite des deux suivantes est également bien sûr $f'(x)$.

Nous verrons en outre que cette méthode nous donne en plus une expression de l'erreur commise en remplaçant la valeur exacte de $f'(x)$ par chacune des 3 expressions que nous allons obtenir. *Ceci est précieux, car que vaut une approximation si l'on n'a pas une idée de l'erreur commise en l'utilisant ?*

3.3 Dérivée première : première approximation

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x+\alpha h), \alpha \in]0, 1[$$

où α est un nombre inconnu dépendant de x et h .

D'où

$$\frac{f(x+h) - f(x)}{h} = f'(x) + \frac{h}{2}f''(x+\alpha h)$$

3.4 Dérivée première : deuxième approximation

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2}f''(x - \beta h), \quad \beta \in]0, 1[$$

où β est un nombre inconnu dépendant de x et h . On en déduit

$$\frac{f(x) - f(x-h)}{h} = f'(x) - \frac{h}{2}f''(x - \beta h)$$

3.5 Dérivée première : troisième approximation

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{3!}f'''(x + \alpha h), \quad \alpha \in]0, +1[\quad (1)$$

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{3!}f'''(x - \beta h), \quad \beta \in]0, 1[\quad (2)$$

où α et β sont des nombres inconnus dépendant de x et h . Il vient alors

$$\frac{f(x+h) - f(x-h)}{2h} = f'(x) + \frac{h^2}{6}(f'''(x + \alpha h) + f'''(x - \beta h)),$$

Appliquons alors le théorème des valeurs intermédiaires, lorsque la dérivée seconde de f est continue. On arrive ainsi à

$$\frac{f(x+h) - f(x-h)}{2h} = f'(x) + \frac{h^2}{3}f'''(x + \gamma h), \quad \gamma \in]-1, 1[.$$

Nous voyons ainsi que pour les deux premières approximations de la dérivée première, l'erreur est en h . Cela signifie concrètement que quand on divise le pas h par 10, l'erreur se divise par 10.

Par contre pour la troisième approximation l'erreur est en h^2 . Cela signifie que quand on divise le pas h par 10, cette fois-ci l'erreur est divisée par 100.

3.6 Ordre de convergence

Pour les deux premières approximations, l'erreur tend vers 0 comme h . On dit que la convergence est d'**ordre 1 (ou linéaire)**. Pour la troisième, l'erreur tend vers 0 comme h^2 . On dit que la convergence est d'**ordre 2 (ou quadratique)**.

Ces ordres de convergence peuvent s'observer expérimentalement. À cet effet, on pourra écrire une fonction Scilab qui le permette effectivement (Appendices B et C).

3.7 Dérivée seconde

Ce calcul est aussi important pour les applications. En effet, beaucoup de lois de la physique s'expriment par des équations différentielles (ou aux dérivées partielles) faisant intervenir des dérivées secondes. C'est le cas par exemple de la mécanique (loi de Newton $\vec{f} = m\vec{\gamma}$).

À nouveau, la formule de Taylor est notre outil de construction.

$$f(x+h) = f(x) + hf'(x) + \frac{h^2}{2}f''(x) + \frac{h^3}{3!}f'''(x) + \frac{h^4}{4!}f^{(4)}(x + \alpha h), \quad \alpha \in]0, 1[\quad (3)$$

$$f(x-h) = f(x) - hf'(x) + \frac{h^2}{2}f''(x) - \frac{h^3}{3!}f'''(x) + \frac{h^4}{4!}f^{(4)}(x - \beta h), \quad \beta \in]0, 1[\quad (4)$$

où α et β sont des nombres inconnus dépendant de x et h . Il vient alors

$$\frac{f(x-h) - 2f(x) + f(x+h)}{h^2} = f''(x) + \frac{h^2}{4!} (f^{(4)}(x + \alpha h) + f^{(4)}(x - \beta h))$$

Appliquons alors le théorème des valeurs intermédiaires, lorsque la dérivée quatrième de f est continue. On arrive ainsi à

$$\frac{f(x-h) - 2f(x) + f(x+h)}{h^2} = f''(x) + \frac{h^2}{12} f^{(4)}(x + \gamma h), \quad \gamma \in]-1, 1[.$$

Nous voyons ainsi que la convergence est d'ordre 2 (quadratique).

3.8 Équation de la chaleur

Avant de commencer, voyons un exemple simple qui montre que l'on peut être amené à résoudre des systèmes linéaires de grande taille. Il s'agit de la résolution de l'équation de la chaleur stationnaire en dimension 1 :

$$\begin{cases} -\frac{d^2 u}{dx^2}(x) &= f(x), \quad x \in]0, 1[, \\ u(0) &= 0, \\ u(1) &= 0. \end{cases}$$

Nous avons ici un problème dont l'inconnue est une fonction u à déterminer sur l'intervalle $]0, 1[$. Autrement dire, il nous faut connaître les valeurs $u(x)$ en tout point de l'intervalle $]0, 1[$.

On peut imaginer $u(x)$ comme la distribution de température d'une barre de longueur 1 (dont toutes les constantes physiques ont été normalisées), chauffée de l'intérieur par un flux de chaleur $f(x)$. Les extrémités de la barre sont maintenues à une température nulle. Dans certaines situations particulières il est possible d'obtenir la solution exacte ; c'est le cas lorsque l'on connaît une double primitive de f . Dans le cas général on peut chercher une approximation de $u(x)$ en certains points (x_k) , avec $k = 0 \dots N$. Pour simplifier, on choisit des points x_k équidistants, on a donc $h = \frac{1}{N}$, et $x_k = kh$.

Aux points x_k , l'équation se récrit :

$$\begin{cases} -\frac{d^2 u}{dx^2}(x_k) &= f(x_k), \quad k = 1, \dots, N-1, \\ u(x_0) &= 0, \\ u(x_N) &= 0. \end{cases} \quad (5)$$

Cette fois-ci, nous nous contentons de vouloir calculer les valeurs de u en un nombre fini de points : les points x_k , pour $k = 1, \dots, N-1$.

On remarque que dans ce problème, la valeur de u est donnée sur les bords de l'intervalle $[0, 1]$.

Maintenant nous allons approcher les dérivées secondes apparaissant dans l'équation, par des formules n'utilisant que la valeur de u sur le réseau de points (x_k) . Nous avons vu au paragraphe précédent, que l'on peut approcher la dérivée seconde de u à l'aide de la formule classique :

$$\frac{d^2 u}{dx^2}(x_k) = \frac{u(x_{k-1}) - 2u(x_k) + u(x_{k+1}))}{h^2} + \mathcal{O}(h^2) \quad (6)$$

Si l'on se donne le droit de négliger le reste $\mathcal{O}(h^2)$ (on peut légitimement penser que ceci est raisonnable si le nombre N de subdivisions est assez grand), on peut obtenir une approximation $v_k \approx u(x_k)$ en écrivant les équations suivantes :

$$\begin{cases} -\frac{v_{k-1} - 2v_k + v_{k+1}}{h^2} &= f(x_k), \quad k = 1, \dots, N-1 \\ v_0 &= 0, \\ v_N &= 0, \end{cases}$$

qui peuvent se mettre sous la forme matricielle suivante :

$$\frac{1}{h^2} \begin{pmatrix} 2 & -1 & & & 0 \\ -1 & 2 & -1 & & \\ & \ddots & \ddots & \ddots & \\ & & -1 & 2 & -1 \\ 0 & & & -1 & 2 \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \\ \vdots \\ v_{N-2} \\ v_{N-1} \end{pmatrix} = \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_{N-2}) \\ f(x_{N-1}) \end{pmatrix}$$

et $v_0 = v_N = 0$.

Voilà donc un exemple de problème physique dont la résolution approchée conduit à la résolution d'un système linéaire (ici la matrice du système a une forme assez particulière : c'est une matrice ***tridiagonale***). Nous allons maintenant voir quelles sont les différentes méthodes "raisonnables" de résolution des systèmes linéaires, raisonnables dans le sens où ces méthodes peuvent être facilement implantées sur ordinateur.

A Fonction scilab : suite récurrente

```
function [Inp1]=suite(n)
// un tableau scilab commence a l'indice 1
// I(n+1) contient I_n
I(1)=1-1/exp(1)
for k=1:n
    I(k+1)=-1/exp(1)+k*I(k)
end
Inp1=I(n+1)
endfunction
```

```
function [Jnp1]=suiteinv(n,N)
// un tableau scilab commence a l'indice 1
// J(n+1) contient J_n
J(N+1)=0
for k=N+1:-1:n+2
    J(k-1)=(1/exp(1)+J(k))/(k-1)
end
Jnp1=J(n+1)
endfunction
```

```
function [f]=fn(n,x)
f=x^n*exp(-x)
endfunction
```

```
function [T]=trapeze(f,n,N)
h=1/N
T=(f(n,0)+f(n,1))/2
for k=1:N-1
    T=T+f(n,k*h)
end
T=T*h
endfunction
```



```

function [direct,inverse,trapez]=boucle(n,N,NT)
    for i=1:n+1
        direct(i)=suite(i-1)
        inverse(i)=suiteinv(i-1,N)
        trapez(i)=trapeze(fn,i-1,NT)
    end
endfunction

```

B Fonction scilab : dérivation numérique

```

function [tfe,tfe2]=convergence(f,fp,n)
// Derivation numerique
// f : fonction a derivier, fp : fonction derivee de la precedente
// tfp : valeurs donnees par la 1ere formule d'approximation
// tfp2 : valeurs donnees par la 2eme formule d'approximation
//tfe, tfe2 : valeurs des erreurs correspondantes
h=1
for i=1:n
    tfp(i)=(f(h)-f(0))/h
    tfe(i)= fp(0)-tfp(i)
    h=h/10
end
h=1
for i=1:n
    tfp2(i)=(f(h)-f(-h))/(2*h)
    tfe2(i)= fp(0)-tfp2(i)
    h=h/10
end
endfunction

```

C Fonction scilab : dérivation numérique (suite)

```

function [tfe]=convergence2(f,fs,n)
// Approximation de la derivee seconde
// f : fonction a derivier, fs : valeur exacte de sa derivee seconde
// tfs : valeurs donnees par l'approximation
// tfe : erreurs correspondantes
h=1
for i=1:n
    tfs(i)=(f(-h)-2*f(0)+f(h))/(h^2)
    tfe(i)= fs(0)-tfs(i)
    h=h/10
end
endfunction

```