

Résumé du cours de Calcul numérique - 2007-2008.

Références : Institut Paul Lambin - cours de Annick Dupont.

1e semestre	2
Les erreurs	2
erreur absolue	2
erreur relative	2
Les nombres à virgules flottantes	2
Les équations du second degré	2
Théorème de Bolzano.....	2
Méthode dichotomique ou par bisection.....	2
Méthode de la corde	3
2e semestre	3
Les équations du second degré.....	3
Méthode de Newton	3
Méthode de Newton modifiée	3
Système d'équation linéaire	4
Méthodes exactes	4
- Formules de Cramer.....	4
- Méthode de Gauss	4
Méthodes approchées	5
- Approximations successives	5
- Méthode de Seidel.....	5

1e semestre

Les erreurs

- arrondi
- discrétisation
- convergence
- inhérente aux données
- machine et humaine
- absolue et relative (nombre approché)

$$\tilde{a} \approx a$$

$\tilde{a} < a \rightarrow$ approché par défaut

$\tilde{a} > a \rightarrow$ approché par excès

erreur absolue

$$\Delta = a - \tilde{a}$$

Borne supérieure : $\Delta = |a - \tilde{a}|$

$$\underbrace{\tilde{a} - \Delta a}_{\text{Par défaut}} \leq a \leq \underbrace{\tilde{a} + \Delta a}_{\text{Par excès}}$$

Décimales exactes :

Si $\Delta < 0,5 * 10^{-n}$

Alors \tilde{a} a n décimales exactes $\rightarrow a = \tilde{a} \pm 0,5 * 10^{-n}$

erreur relative

$$\delta = |a - \tilde{a}| / a$$

modéré : * et /

catastrophe : + et - \rightarrow perte de signification : éviter d'additionner (ou soustraire) des nombres voisins.

Les nombres à virgules flottantes

+ petit = $0,1 * 10^{-n}$

+ grand = $0,99 * 10^n$

fl(x) = élément de p chiffres la + proche de x.

= $x * (1 + \epsilon)$ et $|\epsilon| \leq 5 * 10^{-p}$ (Pour plus d'info sur les arrondis fl(x) : voir le cours.).

overflow $> 0,99...9 * 10^n$

underflow $< 10^{-n-1}$

Les équations du second degré

$$\Delta = b^2 - 4ac$$

Si $\Delta < 0 \rightarrow$ pas de solution

$\Delta = 0 \rightarrow$ une seule solution

$\Delta > 0 \rightarrow$ 2 solutions : $(-b \pm \sqrt{\Delta}) / 2a$

Théorème de Bolzano

Sur une fonction continue, si on passe d'une image négative à une image positive (et vice-versa) on a au moins une racine.

Méthode dichotomique ou par bisection

A chaque fois, on divise l'intervalle en deux et on garde la moitié qui contient la racine. Si en divisant on tombe sur la racine, on a la valeur, pas la peine de continuer. d décimales exactes = $0,5 * 10^{-d} = \epsilon$

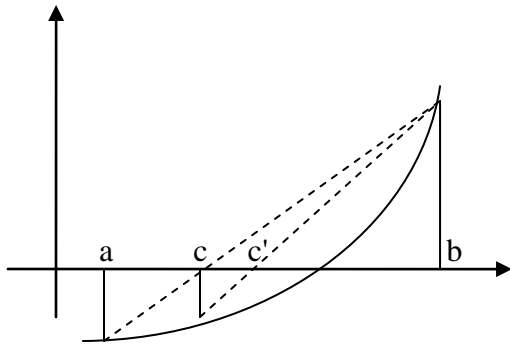
Méthode de la corde

Au lieu de prendre le point milieu du segment $[a,b]$, on va approcher la fonction f par la droite passant par les points $(a,f(a))$ et $(b,f(b))$ et prendre le point d'intersection avec l'axe des x (abscisse).

En développant on trouve :

$$c = a - \frac{f(a)}{f(b) - f(a)} * (b-a)$$

Selon le signe de $f(a) * f(c)$, on prend l'intervalle $[a,c]$ ou $[c,b]$.



Condition :

Si dérivée première (d') existe
et dérivée seconde (d'') existe
et sont de signe constant sur $[a,b]$.

A l'étape n , on a trouvé une approximation x_n avec
une borne d'erreur absolue égale à $|f(x_n) / f'(x_n)|$

C'est-à-dire :

$$|x_n - x| \leq |f(x_n) / f'(x_n)| \quad \text{où } x \text{ est la racine exacte.}$$

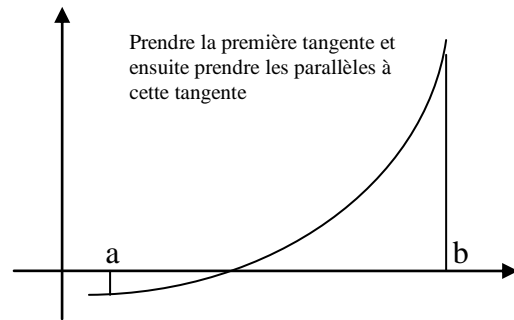
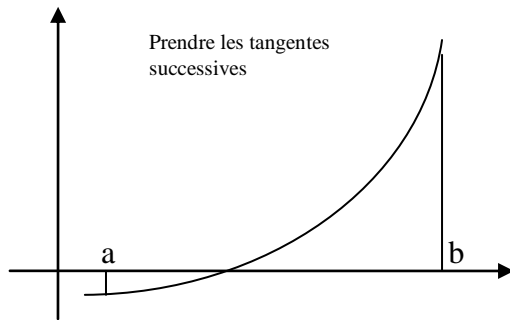


2e semestre

Les équations du second degré

	Méthode de Newton	Méthode de Newton modifiée
1 ^e approximation	Dépend du signe de f'' Si $+$, on prend l'extrémité $+$ Si $-$, on prend l'extrémité $-$ L'autre extrémité sera le point fixe.	
Approximations suivantes	$x_{n+1} = x_n - (f(x_n)/f'(x_n))$	$x_{n+1} = x_n - (f(x_n)/f'(x_0))$
Conditions de convergence	1° la dérivée 1e (d') existe et est constante sur l'intervalle $[a,b]$. 2° la dérivée 2e (d'') existe et est constante sur l'intervalle $[a,b]$.	
Borne d'erreur absolue	$ f(x_n) / m $ où m est la dérivée du point fixe $\rightarrow x_n - x \leq f(x_n) / m $	
≠ entre les 2 méthodes	Calcul récurrent, on recalcule la dérivée 1 ^e .	On ne calcule qu'une fois la dérivée 1 ^e .

Si $x_0 \in [a,b]$ et si $f(x_0) * f''(x_0) > 0$, la suite $\{x_n\}$ converge vers la racine située dans $[a,b]$.



Système d'équation linéaire

Méthodes exactes

- Formules de Cramer

$$\begin{cases} x_1 + x_2 + x_3 = 100 \\ x_1 = x_2 \\ x_3 = x_1 + 10 \end{cases} \Rightarrow \begin{cases} x_1 + x_2 + x_3 = 100 \\ x_1 - x_2 = 0 \\ -x_1 + x_3 = 10 \end{cases}$$

$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 100 \\ 0 \\ 10 \end{pmatrix} \Rightarrow x = A^{-1} * b$$

$$x_1 = \Delta_1 / \Delta \Rightarrow \Delta_1 = \begin{vmatrix} 100 & 1 & 1 \\ 0 & -1 & 0 \\ 10 & 0 & 1 \end{vmatrix} = 100 \begin{vmatrix} -1 & 0 \\ 0 & 1 \end{vmatrix} - 1 \begin{vmatrix} 0 & 0 \\ 10 & 1 \end{vmatrix} + 1 \begin{vmatrix} 0 & -1 \\ 10 & 0 \end{vmatrix}$$

$$= 1(-1) - 1(1) + 1(0)$$

$$= -90$$

Déterminant du système |A|

$$\Delta = \begin{vmatrix} 1 & 1 & 1 \\ 1 & -1 & 0 \\ -1 & 0 & 1 \end{vmatrix} = 1 \begin{vmatrix} -1 & 0 \\ 0 & 1 \end{vmatrix} - 1 \begin{vmatrix} 1 & 0 \\ -1 & 1 \end{vmatrix} + 1 \begin{vmatrix} 1 & -1 \\ -1 & 0 \end{vmatrix}$$

$$= 1(-1) - 1(1) + 1(0)$$

$$= -2$$

- Méthode de Gauss

$$x_i = 1 / C_{ii} * (b_i - \sum C_{ik} x_k) \rightarrow i = n, n-1, \dots$$

$$\rightarrow \sum \text{des } n \text{ partant de } k = i+1$$

= Triangularisation

But : arriver à faire un triangle.

Dès que le système dépassera 3 équations, on préférera utiliser Gauss plutôt que Cramer !

$$\begin{array}{rcl} \pm x_1 & \pm x_2 & \pm x_3 = \dots \\ 0 & \pm x_2 & \pm x_3 = \dots \\ 0 & 0 & \pm x_3 = \dots \end{array}$$

Programmation : 3 for \rightarrow r = étape
i = ligne
j = colonne

Méthodes approchées

On travaille ici avec Alpha (α) et Beta (β).

On les forme ainsi :

$$\alpha_{ij} = a_{ij} / a_{ii} \text{ (pour } i \neq j) \text{ et } \alpha_{ii} = 0$$

$$\beta_i = b_i / a_{ii}$$

- Approximations successives

$$x_{(0)} = \beta$$

$$x_{(1)} = \beta + \alpha x_{(0)}$$

$$x_{(2)} = \beta + \alpha x_{(1)}$$

...

$$x_{(n)} = \beta + \alpha x_{(n-1)}$$

Ce qui donne :

x1	= $\beta_1 + \alpha_{12} * X_2$ (à l'étape précédente) + $\alpha_{13} * X_3$ (à l'étape précédente) + $\alpha_{14} * X_4$ (à l'étape précédente)
x2	= $\beta_2 + \alpha_{21} * X_1$ (à l'étape précédente) + $\alpha_{23} * X_3$ (à l'étape précédente) + $\alpha_{24} * X_4$ (à l'étape précédente)
x3	= $\beta_3 + \alpha_{31} * X_1$ (à l'étape précédente) + $\alpha_{32} * X_2$ (à l'étape précédente) + $\alpha_{34} * X_4$ (à l'étape précédente)
x4	= $\beta_4 + \alpha_{41} * X_1$ (à l'étape précédente) + $\alpha_{42} * X_2$ (à l'étape précédente) + $\alpha_{43} * X_3$ (à l'étape précédente)

Condition de convergence :

$$|a_{ii}| > \sum_{j=1, j \neq i}^n |a_{ij}| \quad (i = 1, 2, \dots, n)$$

- Méthode de Seidel

Pareil que pour les approximations successive sauf qu'on calcul en fonction des dernières approximations.

Ce qui donne :

x1	= $\beta_1 + \alpha_{12} * X_2$ (à l'étape précédente) + $\alpha_{13} * X_3$ (à l'étape précédente) + $\alpha_{14} * X_4$ (à l'étape précédente)
x2	= $\beta_2 + \alpha_{21} * X_1$ (à cette étape ci) + $\alpha_{23} * X_3$ (à l'étape précédente) + $\alpha_{24} * X_4$ (à l'étape précédente)
x3	= $\beta_3 + \alpha_{31} * X_1$ (à cette étape ci) + $\alpha_{32} * X_2$ (à cette étape ci) + $\alpha_{34} * X_4$ (à l'étape précédente)
x4	= $\beta_4 + \alpha_{41} * X_1$ (à cette étape ci) + $\alpha_{42} * X_2$ (à cette étape ci) + $\alpha_{43} * X_3$ (à cette étape ci)

