UNIVERSITÉ ABDELMALEK ESSAADI

FACULTÉ DES SCIENCES - TÉTOUAN

Licence Fondamentale Sciences de la Matière Physique

Semestre 3 - M20 : Analyse Numérique et Algorithmique

SUPPORT DE COURS

Rédigé par : **Bouchaib FERRAHI**

Département de Mathématiques

2023-2024

Les documents relatifs à ce cours sont disponibles sur : www.ferrahi.ma

---

Faculté des Sciences de Tétouan, BP. 2121 M'Hannech II, 93030 Tétouan Maroc.

Table des matières

Avant-propos

Ce polycopié, destiné aux étudiantx du semestre trois de la Licence Fondamentale Sciences de la Matière Physique, est conforme au nouveau programme appliqué depuis 2014. En particulier, les deux cous de Mathématiques visent le développement de l'esprit d'analyse et de synthèse et la valorisation de l'approche scientifique dans le traitement des problèmes théoriques et expérimentaux.

Le contenu proposé, pour ce cours d'analyse numérique et algorithmique, consiste en un recueil de méthodes de résolution numérique de plusieurs problèmes Mathématiques allant des equations non linéaires, à l'interpolation polynomiale, au calcul des intégrales et finalement calcul différentiel et certaines équations différentielles. Les méthodes numériques sont très utiles dans les sciences appliquées et dans la traitement des problèmes expérimentaux, ce cours vise à munir les étudiants d'outils indispensables pour leur cursus dans la filière Licence es sciences Physique et éventuellement dans les cycles supérieurs.

Ce polycopié est adapté à la filière sus-mentionnée et se limitera à la présentation des notions, définitions, propriétés et résultants fondamentaux avec des exemples d'application, les démonstrations et les fondements théoriques ont été omit pour bien cibler les étudiants concernés.

Ce travail ne constitue pas une référence complète, le lecteur intéressé peut consulter d'autres références qui traitent ce même contenu d'une manière plus profonde et rigoureuse.

BOUCHAIB FERRAHI

Introduction et Préliminaires

L'analyse numérique est une discipline Mathématique qui peut être considérée comme partagée en deux grands thèmes ou deux axes. Le premier est constitué des méthodes numériques de résolution de grands systèmes linéaires, de l'intégration numérique, de la résolution numérique des équations différentielles, et des méthodes d'optimisation,... Le deuxième axe est constitué des méthodes d'approximation numérique des équations aux dérivées partielles, méthodes basées éléments finis et les volumes finis, méthodes spectrales,...

Bien que cette discipline existait bien avant les ordinateurs avec des méthodes Mathématiques basées sur des techniques appliquées appropriées, elle a bénéficié des développements permanents de l'informatique et a connue des avancées importantes durant les dernières décennies. La résolution théorique des problèmes Mathématiques est importante mais parfois les méthodes utilisées sont difficiles ou non adaptées à des situations pratiques. Pour les sciences appliquées, les sciences de l'ingénieur et d'autres domaines d'application des Mathématiques, les utilisateurs, souvent non spécialistes, ont besoin de résultats fiables et immédiatement utilisables, l'analyse numérique combiné à l'algorithmique et à la programmation informatique peuvent être très utiles dans ce sens et fournissent des réponses aux questions précédentes.

Dans ce documents, des méthodes numériques, pour un certain nombre de problèmes Mathématiques, sont présentées et dans un but pédagogique, les résultats obtenus sont comparer avec les résultats exacts lorsque ces derniers sont bien déterminés théoriquement. Nous commençant par présenter quelques notions d'écriture suivant une base b avec des applications sur les opérations binaires. Ces préliminaires trouvent place dans "l'arithmétique des ordinateurs" qui s'intéressent à l'étude des méthodes de calcul utilisées par les ordinateurs en relation avec les "erreurs" liées à ces méthodes. en effet, Il faut rappeler que toute opération mathématique sur des nombres (Dans $\mathbb{R}$ ou $\mathbb{C}$) effectuée sur ordinateur ne peut se faire que si les nombres en question sont constitués d'un nombre de chiffres significatifs. Et donc toute les opérations arithmétiques est soumise à la contrainte technique du nombre de chiffres utilisables : il faudra, impérativement, **arrondir** ou **tronquer**.

Dans les quatre chapitres de ce document, nous traitons des questions classiques et usuelles de l'analyse numérique. Dans le premier chapitre, la résolution numérique des equations non linéaires est étudiée en présentant quelques méthodes assurant le calcul d'une **valeur approchée** de la racine de l'équation $f(x) = 0$ et en maîtrisant et évaluant **"l'erreur"** commise. Ensuite, dans le deuxième chapitre, l'interpolation polynômial est étudié en explorant les méthodes permettant d'approcher une fonction f par un polynôme de degré n, cette étude est complétée par l'evaluation des "erreurs" commises. Finalement, dans les deux derniers chapitres, deux questions liées sont étudiées, à savoir le calcul numérique d'une intégrale ainsi que la résolution numérique des équations différentielles d'un type particulier dit problème de Cauchy.

Dans les sections suivantes, nous présentons quelques éléments de base qui servent à développer l'arithmétique des ordinateurs. En effet, en informatique et en électronique, nous utilisons d'autres bases, d'écriture des nombre, que celle que nous utilisons quotidiennement dans vie courante.

Écriture suivant une base b

Les nombres que nous utilisons chaque jour peuvent être interpréter de la manière décrite dans l'exemple suivant :
$2132 =$ "Deux" milles + "un" cent + "trois" dizaines + "deux" unités
$2132 = 2 \times 1000 + 1 \times 100 + 3 \times 10 + 2 \times 1$
$2132 = 2 \times 10^3 + 1 \times 10^2 + 3 \times 10^1 + 2 \times 10^0$
Cette représentation n'est d'autre que l'écriture habituelle suivant la base décimale (base 10). Tout entier $a_n a_{n-1}....a_1 a_2$ s'écrit d'une manière naturelle : $a_n \times 10^n + a_{n-1} \times 10^{n-1} + ... + a_1 \times 10^1 + a_0 \times 10^0$.
D'une manière générale l'écriture suivant une base b est définie par :

Définition 0.0.1
Soit b un entier positif, tout entier naturel m admet une écriture unique suivant la base b donée par :

$$m = a_n \times b^n + a_{n-1} \times b^{n-1} + ... + a_1 \times b^1 + a_0 \times b^0$$

avec : $a_n \neq 0$ et pout tout i on a : $a_i = 0, 1, 2,...,b-1$. On note :

- *on note :*

$$m = (a_n a_{n-1}....a_1 a_0)_b$$

ou

$$m = \overline{a_n a_{n-1}.....a_1 a_0}^b$$

- *Si $b = 10$ on écrit habituellement, lorsque aucune confusion n'est possible, $a_n a_{n-1}....a_1 a_0$ au lieu de*

$$(a_n a_{n-1}....a_1 a_0)_{10}$$

ou

$$\overline{a_n a_{n-1}.....a_1 a_0}^{10}$$

Exemples 0.0.2

- *Base binaire : $b = 2$, tout entier naturel s'écrit $\overline{a_n a_{n-1}.....a_1 a_0}^2$ avec $a_i = 0$ ou $a_i = 1$, utilisée en Électronique et Informatique*
- *Base octale : $b = 8$, tout entier naturel s'écrit $\overline{a_n a_{n-1}.....a_1 a_0}^8$ avec $a_i = 0, 1,...,7$, utilisée en Informatique*
- *Base hexadécimale, $b = 16$, tout entier naturel s'écrit $\overline{a_n a_{n-1}.....a_1 a_0}^{16}$ avec $a_i = 0, 1,...,9$, $a = 10$, $b = 11,...f = 15$, utilisée en Informatique*
- $1830 = (1830)_{10} = \overline{1830}^{10} = 1 \times 10^3 + 8 \times 10^2 + 3 \times 10^1 + 0 \times 10^0$;

- $1830 = (11100100110)_2 = \overline{11100100110}^2 = 1 \times 2^{10} + 1 \times 2^9 + 1 \times 2^8 + 0 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$;
- $1830 = (726)_{16} = \overline{726}^{16} = 7 \times 16^2 + 2 \times 16^1 + 6 \times 16^0$.

Les écritures binaires, octales et hexadécimales des premiers entiers naturels sont données dans le tableau suivant :

Décimale (10)	Binaire (2)	Octale (8)	Hexadécimale (16)
0	00000	00	0
1	00001	01	1
2	00010	02	2
3	00011	03	3
4	00100	04	4
5	00101	05	5
6	00110	06	6
7	00111	07	7
8	01000	10	8
9	01001	11	9
10	01010	12	A
11	01011	13	B
12	01100	14	C
13	01101	15	D
14	01110	16	E
15	01111	17	F
16	10000	20	10

Remarques 0.0.3

Pour éviter les confusions, il est nécessaire de préciser la base utilisée, par exemple :

- 1101 *en base* $b = 10$ *:* $(1101)_{10} = 1101$
- 1101 *en base* $b = 8$ *:* $(1101)_8 = 577$
- 1101 *en base* $b = 2$ *:* $(1101)_2 = 13$

Conversion des écritures entre deux bases :

Définition 0.0.4 *base* $b \to$ *base décimale :*

La conversion d'une écriture $(a_n a_{n-1} ... a_1 a_0)_b$ *en base* b *quelque à une écriture en base décimale s'obtient en calculant la somme :*

$$a_n \times b^n + a_{n-1} \times b^{n-1} + ... + a_1 \times b^1 + a_0 \times b^0$$

Exemples 0.0.5

- $(11001)_2 = 1\times 2^4 + 1\times 2^3 + 1\times 2^0 = 16+8+1 = 25$;
- $(1201)_3 = 1\times 3^3 + 2\times 3^2 + 1\times 3^0 = 27+18+1 = 46$;
- $(7A9E)_{16} = 7\times 16^3 + 10\times 16^2 + 9\times 16^1 + 14\times 16^0 = 7\times 4096 + 10\times 256 + 9\times 16 + 14 = 28672+2560+144+14 = 31390$.

Définition 0.0.6 *base décimale $\to$ base b :*
La conversion s'obtient en effectuant des divisions successives du nombre, écrit en base décimale, par b et en classant les restes dans le sens inverse : $m = bq_0 + r_0$, $q_0 = bq_1 + r_1$, ...,$q_{n-2} = bq_{n-1} + r_{n-1}$, $q_{n-1} = b\times 0 + r_n$, avec $q_{n-1} = r_n < b$ et $q_n = 0$, alors :

$$m = (r_n r_{n-1} .... r_1 r_0)_b$$

Exemples 0.0.7

- $41 = 2\times 20 + 1$, $20 = 2\times 10 + 0$, $10 = 2\times 5 + 0$, $5 = 2\times 2 + 1$, $2 = 2\times 1 + 0$ *et* $1 = 2\times 0 + 1$ *donc :* $(41)_{10} = (101001)_2$
- $1830 = 16\times 114 + 6$, $114 = 16\times 7 + 2$, $7 = 16\times 0 + 7$, *donc :* $(1830)_{10} = (726)_{16}$
- $479 = 7\times 68 + 3$, $68 = 7\times 9 + 5$, $9 = 7\times 1 + 2$, $1 = 7\times 0 + 1$, *donc :* $479_{10} = (1253)_7$

Définition 0.0.8 *base $b \to$ base b' :*
La conversion s'obtient en passant par la base décimale :
Base $b \to$ base $10 \to$ base b'.

Définition 0.0.9 *Base puissance de l'autre ($b \to b^k$)*
On découpe la représentation de m en base b en tranches de k chiffre, en commençant par la droite et en rajoutant des 0 à gauche si le nombre de chiffres de m n'est pas un multiple de k. Chaque tranche de k chiffres est alors transformée en un chiffre en base b^k. Ces chiffres, écrits dans cet ordre, constituent l'écriture de m en base b^k.

Exemples 0.0.10

- *Écriture de $(1022102)_3$ en base $9 = 3^2$, on coupe l'écriture à des tranches de 2 en commençant par la droite et en ajoutant 0 à gauche :*

$$\underbrace{\overline{\mathbf{0}1}^3}_{=1}\ \underbrace{\overline{02}^3}_{=2}\ \underbrace{\overline{21}^3}_{=7}\ \underbrace{\overline{02}^3}_{=2} = (1272)_9$$

- *Écriture de $(10101101110)_2$ en base $16 = 2^4$:*

$$\underbrace{\overline{\mathbf{0}101^2}}_{=5}\ \underbrace{\overline{0110}^2}_{=6}\ \underbrace{\overline{1110}^2}_{=14=E} = (56E)_{16}$$

Définition 0.0.11 *Base puissance de l'autre ($b^k \to b$)*
On exprime chaque chiffre en base b^k comme un nombre écrit en base b sur k chiffres, en rajoutant des 0 (à gauche) si l'écriture du nombre obtenu comporte moins de k chiffres en base b.

Exemples 0.0.12

- *Écriture de* $1A2F_{16=2^4}$ *en base* $b = 2$ *:*

$$\overline{1}^{16}\overline{A}^{16}\overline{2}^{16}\overline{F}^{16} = \underbrace{\overline{\mathbf{00}1}^{2}}\,\underbrace{\overline{1010}^{2}}\,\underbrace{\overline{\mathbf{00}10}^{2}}\,\underbrace{\overline{1111}^{2}}$$

et

$$= (1A2F)_{16} = (1101000101111)_2$$

- *Écriture de* $156_{8=2^3}$ *en base* $b = 2$ *:*

$$\overline{1}^{8}\overline{5}^{8}\overline{6}^{8} = \underbrace{\overline{\mathbf{00}1}^{2}}\,\underbrace{\overline{101}^{2}}\,\underbrace{\overline{110}^{2}} = (001101110)_2 = (1101110)_2$$

Codage binaire et octal

On entend souvent parler de bit ou octet (en anglais, bit or Byte), quelle est la différence entre les deux notions et comment elles sont utilisées ?

- Une information binaire (symbolisée couramment par 0 ou 1) s'appelle un bit (en anglais... bit)
- Un groupe de huit bits s'appelle un octet (en anglais, byte). Donc, Il ne faut pas confondre le byte (en abrégé, B majuscule), qui vaut un octet, c'est à dire huit bits (en abrégé, b minuscule) avec un bit.

Remarques 0.0.13 *États possibles :*

- *bit* $\rightarrow$ *2 états possibles : 0 ou 1*
- *octet* $\rightarrow 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 = 2^8 = 256$ *états possibles*
- *2 octets* $\rightarrow 256 \times 256 = 65536$ *états possibles*
- *3 octets* $\rightarrow 256 \times 256 \times 256 = 16777216$ *états possibles*
- *Pas que les chiffres, un octet peut représenter : lettres, chiffres, signes de ponctuation* $\rightarrow$ *un caractère par octet est choix pertinent*
- *Pour faciliter le codage et la communication entre deux machines, il faut unifier la manière de codage en fixant une norme standard : Quel état de l'octet correspond à quel signe du clavier ?*
- *ASCII (American Standard Code for Information Interchange) est un standard universellement appliqué par les fabricants d'ordinateurs et de logiciels.*
- *ASCII traite aussi les cas particuliers des signes propres à telle ou telle langue (comme les lettres accentuées en franç ais, par exemple).*

Opérations habituelles dans une base b

Les opérations habituelles (somme, soustraction et multiplication) peuvent être effectuées dans une base b en prenant en considération que $0 \leq a_i < b$ et en utilisant, s'il le faut, une retenue.

Exemples 0.0.14 *Opérations binaires*
Addition : $0+0=0$, $1+0=0+1=1$ *et* $1+1=1$ *avec une retenue de* 1.
Effectuons en binaire l'addition suivante : $(34)_{10}+(27)_{10}$ *:*

$$(34)_{10}+(27)_{10}=(100010)_2+(11011)_2=(111101)_2$$

Vérification : $(34)_{10}+(27)_{10}=(61)_{10}$ *et* $(61)_{10}=(111101)_2$

Représentation des nombres sur machine

Les informations traitées par un ordinateur sont de différents types (nombres, instructions, images, vidéo...) mais elles sont toujours représentées sous un format binaire. Seul le codage changera suivant les différents types de données à traiter. Elles sont représentées physiquement par 2 niveaux de tensions différents. En binaire, une information élémentaire est appelé bit et ne peut prendre que deux valeurs différentes : 0 ou 1 . Une information plus complexe sera codée sur plusieurs bit. On appelle cet ensemble un mot. Un mot de 8 bits est appelé un octet. Pour représenter les nombres en machine on choisira une base b et une norme de codage. Les ordinateurs emploient en général trois bases :

— la base 2 ou binaire,
— la base 8 ou octale et
— la base 16 ou hexadécimale.

Dans la représentation binaire, les chiffres se réduisent aux deux symboles 0 et 1 , appelés bits (de l'anglais binary digits). En hexadécimal les symboles utilisés pour la représentation des chiffres sont $0, 1, \ldots, 9, A, B, C, D, E, F$.

Toute opération qu'effectue un ordinateur ("opération machine") est entachée par des erreurs d'arrondi. Elles sont dues au fait qu'on ne peut représenter dans un ordinateur qu'un sous-ensemble fini de l'ensemble des nombres réels.

Représentation des nombres en machine - Les nombres entiers

Sur machine le terme nombres entiers désigne l'ensemble $\mathbf{Z}$. Ils sont exprimés par des chiffres pouvant prendre deux valeurs 0 ou 1. A chaque chiffre est affecté un poids exprimé en puissance de 2
- Exemples (Écriture suivant différentes bases)
$10 : (2019)_{10} = 2.10^3 + 0.10^2 + 1.10^1 + 9.10^0$

$$(101)_2 = 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = (5)_{10}$$

$$(39)_{10} = 32 + 4 + 2 + 1 = 2^5 + 2^2 + 2^1 + 2^0 = (100111)_2$$

Le nombre représentable dépend du nombre d'octets utilisés.

Le nombre représentable dépend du nombre d'octets utilisés :

- bit $\rightarrow$ 2 états possibles : 0 ou 1
- octet $\rightarrow 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 \times 2 = 2^8 = 256$ états possibles
- 2 octets $\rightarrow 256 \times 256 = 65536$ états possibles
- 3 octets $\rightarrow 256 \times 256 \times 256 = 16777216$ états possibles

- avec deux (2) octets, on peut représenter les entiers compris entre -32768 et 32767
- avec quatre (4) octets on peut représenter les entiers compris entre -2147483648 et 2147483647

Représentation des nombres en machine - Les réels à virgule fixe

Soit une base fixée $b \in \mathbb{N}$ avec $b \geqslant 2$, et soit x un nombre réel comportant un nombre fini de chiffres a_k avec $0 \leqslant a_k < b$. On se propose de travailler avec un ordinateur disposant de N cases mémoires pour représenter x. Une case mémoire sera

réserver pour le signe, $N-k-1$ pour les chiffres entiers et k pour les chiffres situés après la virgule, de sorte que :

$$x = (-1)^s \times [a_N a_{N-1} \dots a_k, a_{k-1} \dots a_0], \quad a_N \neq 0$$

Qu'on peut écrire :

$$x = (-1)^s \times b^{-k} \times \left(\sum_{j=0}^{N} a_j b^j \right).$$

où s dépend du signe de $\times$($s = 0$ si $\times$ est positif, 1 si $\times$ est négatif).

L'ensemble des nombres de ce type est appelé système à virgule fixe.

- Exemple :

- $(-526,73)_{10} = (-1)^1 \times 10^{-2} \times \left(3 \times 10^0 + 7 \times 10^1 + 6 \times 10^2 + 2 \times 10^3 + 5 \times 10^4\right)$.
- $(425,31)_6 = (-1)^0 \times 6^{-2} \times \left(1 \times 6^0 + 3 \times 6^1 + 5 \times 6^2 + 2 \times 6^3 + 4 \times 6^4\right)$. L'utilisation de la virgule fixe limite considérablement les valeurs minimales et maximales des nombres pouvant être représentés par l'ordinateur, à moins qu'un très grand nombre N de cases mémoires ne soit employé.

Représentation des nombres en machine - Les réels à virgule flottante

Soit x un nombre réel non nul, sa représentation en virgule flottante est donnée par :

$x = (-1)^s \times (0, a_1 a_2 \dots \dots a_t) \times b^e = (-1)^s \times m \times b^{e-t}$ avec

- $t \in \mathbb{N}$ est le nombre de chiffres significatifs a_i (avec $0 \leq a_i < b$),
- $m = a_1 a_2 \dots a_t$ un entier vérifiant $0 \leq m \leq b^{t-1}$ appelé **mantisse**,
- e un entier appelé **exposant**.

L'exposant ne peut varier que dans un intervalle fini de valeurs admissibles : posons

$$L = e_{min} \leq e \leq U = e_{max}.$$

Les N cases mémoires sont à présent réparties ainsi :

- une case pour le signe,
- t cases pour les chiffres significatifs et
- $N-t-1$ cases pour les chiffres de l'exposant.

Remarque : Le nombre zéro a une représentation à part.

Il y a typiquement sur un ordinateur deux formats disponibles pour les nombres à virgule flottante : les représentations en simple et en double précision. Dans le cas de la représentation binaire, ces formats sont codés dans les versions standards avec $N = 32$ bits (simple précision)

Dans le standard $IEEE$ 754 utilisé par Matlab, on a $b = 2$ et :

- en simple précision : $t = 24$, $e_{min} = -125$, $e_{max} = 128$

et avec $N = 64$ bits (double précision)

Dans le standard $IEEE$ 754 utilisé par Matlab, on a $b = 2$ et :

- en double précision : $t = 53$, $e_{min} = -1021$, $e_{max} = 1024$

Représentation des nombres en machine - Arrondi d'un nombre réel en représentation machine

Sur tout ordinateur, seul le sous-ensemble $\mathcal{F} \subset \mathbb{R}$ soit effectivement disponible. Ainsi un $x \in \mathbb{R}$, sera remplacer par son représentant dans $\mathcal{F}$ qu'on notera $fl(x)$. Par exemple si $x = \dfrac{1}{3} = 0,33333333\ldots$ et $t = 4$ alors $fl(x) = 0,3333$.

D'une manière générale pour calculer $fl(x)$ on procède soit par **Arrondi** soit **Troncature** :

Arrondi

Soit $x \in \mathbb{R}$ en notation normalisée, en **arrondi** $fl(x)$ est défini par :

$$fl(x) = (-1)^s \times (0.a_1a_2\ldots\tilde{a}_t) \times b^e, \quad \tilde{a}_t = \begin{cases} a_t & \text{si } a_{t+1} < \dfrac{b}{2} \\ a_t + 1 & \text{si } a_{t+1} \geq \dfrac{b}{2}. \end{cases}$$

Exemple

Avec 3 chiffres pour $x = 0,8573$ et $y = 0,8576$ on a en arrondi

- $fl(x) = 0,857$
- $fl(y) = 0,858$.

Représentation des nombres en machine - Arrondi d'un nombre réel en représentation machine

Troncature

Dans l'approche **troncature** on prendrait $\tilde{a}_t = a_t$.

$$fl(x) = (-1)^s \times (0.a_1a_2\ldots a_t) \times b^e.$$

Exemple

Avec 3 chiffres, pour $x = 0,8573$ et $y = 0,8576$ on a en troncature :

- $fl(x) = 0,857$
- $fl(y) = 0,857$.

Représentation des nombres en machine - Estimation d'erreurs Soit x un réel et $\overline{x}$ une valeur approchée de x.

- L'erreur absolue $E = E(x)$ est défini par $E = |x - \overline{x}|$.
- L'erreur relative est $E_{rel} = \dfrac{|x - \overline{x}|}{|x|}$.

Soit x un réel. Si x est dans $\mathcal{F}$, alors

$$fl(x) = x(1 + \delta) \quad \text{avec } |\delta| \leq u,$$

où

$$u = \frac{1}{2}b^{1-t} = \frac{1}{2}\varepsilon_M$$

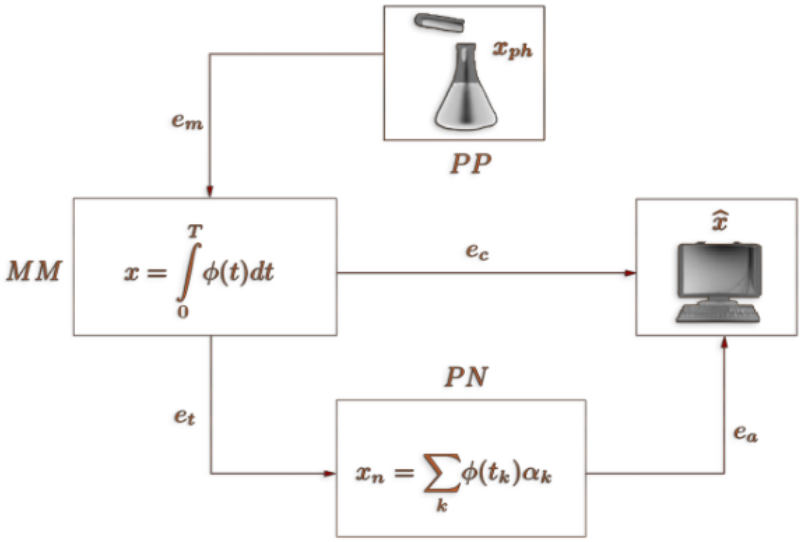


FIGURE 1.1 – Source : A.Quarteroni and all : Calcul Scientifique Cours, exercices corrigés et illustrations en Matlab et Octave.

Le nombre u est l'erreur relative maximale que l'ordinateur peut commettre en représentant un nombre réel en arithmétique finie. Pour cette raison, on l'appelle **unité d'arrondi.**
On déduit immédiatement la majoration suivante de l'erreur relative :

$$E_{rel} = \frac{|x - fl(x)|}{|x|} \leq u$$

Exemple :
En base $b = 10$, $x = 1/15 = 0,066666666.....$

1. Dans le cas d'une **représentation tronquée** nous aurons, pour $s = 5$, $fl(x) = 0,66666 \times 10^{-1}$.
   — L'erreur absolue $x - fl(x)$ est de $6 \times 10^{-7.}$.
   — L'erreur relative est de l'ordre de 10^{-5}
   — Dans une représentation tronquée à s chiffres, l'erreur relative maximale est de l'ordre de 10^{-s}
2. Dans le cas d'une **représentation arrondie** nous aurons $fl(x) = 0,66667 \times 10^{-1}$
   — L'erreur absolue serait alors $3,333 \times 10^{-7}$.
   — L'erreur relative serait 5×10^{-6}
   — En général, l'erreur relative dans une représentation arrondie à s chiffres est de $5 \times 10^{-(s+1)}$ soit la moitié de celle d'une représentation tronquée.

Chapitre 1

Résolution numérique des équations non linéaires

Dans ce chapitre, nous présentons quelques méthodes de résolution numérique d'une équation de la forme $f(x) = 0$, avec f une fonction quelconque et non pas, forcément, un polynôme. Les solutions de cette équation, lorsque elles existent, sont appelées **zéros de la fonction** f. Les méthodes Mathématiques exactes nous permettent de résoudre plusieurs types d'équations mais ne peuvent pas être utilisées pour résoudre toutes les équations. A titre d'exemple, peut on résoudre explicitement l'équation : $e^{x.tan(x)} - 4 = 0$? La réponse évidente, à ce niveau d'études, est non !
Soit $f : \mathbb{R} \to \mathbb{R}$. Faute de méthodes générales applicables pour toutes les équations, il n'est pas toujours possible de résoudre, explicitement, une équation de la forme :

$$f(x) = 0 \tag{1.1}$$

Les méthodes de **résolution numérique** nous permettent d'abord de "localiser" les "zéros de f" dans un intervalle de faible amplitude, puis de calculer des solutions approchées de l'équation(1.1). Avant l'utilisation de ces méthodes numériques, il faut d'abord s'assurer de l'existence de la solution (racine de l'équation ou encore zéro de la fonction) dans un intervalle bien déterminé et étudier, aussi, l'unicité. Cette première étape, nécessite l'utilisation de plusieurs outils Mathématiques (Théorème des valeurs intermédiaires, théorème du point fixe,...).

1.1 Existence, unicité et localisation des racines, d'une fonction, dans un intervalle

Theorem 1.1.1
*Soit f une fonction **continue**, à valeurs dans $\mathbb{R}$, et définie sur un intervalle $[a, b]$. On suppose que*

$$f(a).f(b) < 0$$

*Alors, il existe **au moins** $c \in]a, b[$ tel que :*

$$f(c) = 0$$

*Si, en plus, f est **monotone** sur $]a, b[$ alors c est **unique** sur $]a, b[$.*

Theorem 1.1.2
*Soit g une fonction **continue** définie sur un intervalle $[a, b]$ et vérifiant $g([a, b]) \subset [a, b]$. Alors il existe **au moins** $c \in]a, b[$ tel que :*

$$g(c) = c$$

*Si, en plus, g est **monotone** alors c est **unique** dans $]a, b[$.*

Remarques 1.1.3

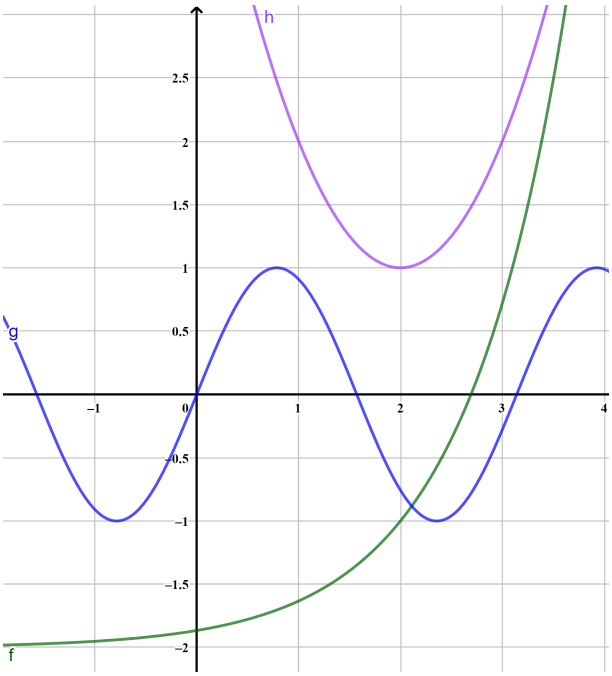
Avant de présenter les méthodes les plus utilisées, faisant les remarques suivantes :

- *Nous pouvons choisir l'intervalle $[a, b]$ de telle façon qu'il ne contient q'une seule solution de (1.1) (ce choix est possible à travers une étude graphique, les propriétés de f, les données relatives au problème initial,...) ;*
- *L'existence et l'unicité de la solution doivent être rigoureusement prouvées avant d'appliquer les méthodes numériques (on peut utiliser les outils Mathématiques présentés précédemment) ;*
- *En général, les méthodes de résolution numérique de l'équation (1.1) sont des méthodes itératives qui consistent à construire une suite $(x_n)_n$ qui converge vers la solution x_0 de (1.1) ;*
- *Nous pouvons comparer les méthodes suivant différents critères : l'erreur commise (différence entre la solution exacte et la solution approchée), la "rapidité" de la convergence,...*

Exemples 1.1.4

Soient f, g et h les fonctions définies sur $[0, 4]$ par :

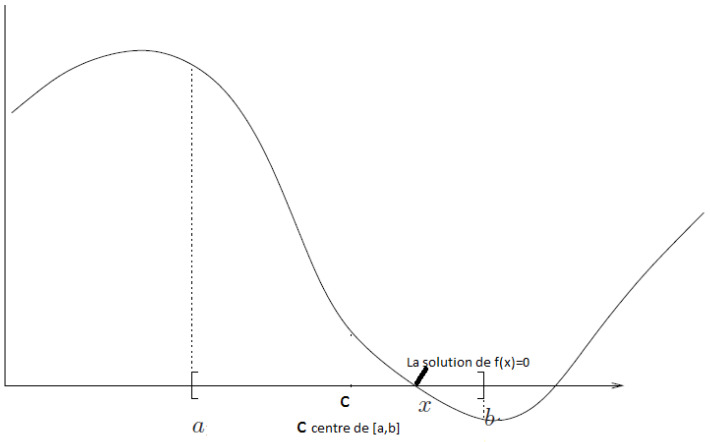
$$f(x) = \exp^{x-2} - 2 \qquad g(x) = \sin(2x) \qquad \text{et } h(x) = (x-2)^2 + 1$$



$f(x) = 0$ *admet **une et une seule** solution sur* $[0, 4]$
$g(x) = 0$ *admet **au moins une** solution sur* $[0, 4]$
$h(x) = 0$ ***n'admet pas de** solution sur* $[0, 4]$

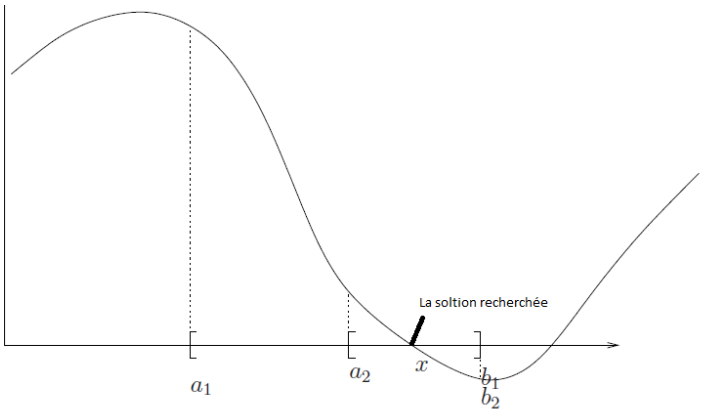
1.2 Méthode de dichotomie ou de la bisection

Soit f une fonction continue sur $[a, b]$ telle que $f(a).f(b) < 0$ (voir figure). Soit $c = \frac{a+b}{2}$, on a :



- Si $f(c) = 0$ alors $x_0 = c$ est solution de (1.1) ;
- Si $f(a).f(c) < 0$ alors la solution est dans $[a, c]$ et c prend la place de b ;
- $f(c).f(b) < 0$ alors la solution est dans $[c, b]$ et c prend la place de a ;
- La méthode est répétée autant de fois nécessaires pour localiser la solution dans un intervalle de petite amplitude (suivant le critère, de tolérance, initialement fixé).

La méthode du dichotomie peut être illustrée par la représentation graphique et le schéma des opérations suivantes : On pose



$a_0 = a$, $b_0 = b$ et $x_n = \frac{a_n+b_n}{2}$. Alors :

- Si $f(x_n) = 0 \rightarrow$ Fin ;
- Si $f(a_n).f(x_n) > 0 \rightarrow$ on pose $a_{n+1} = x_n$ et $b_{n+1} = b_n$,
- Si $f(a_n).f(x_n) < 0 \rightarrow$ on pose $a_{n+1} = a_n$ et $b_{n+1} = x_n$,

La méthode de la dichotomie génère une suite $(x_n)_n$ qui converge vers la solution recherchée, mais, il faut préciser un test d'arrêt, par exemple, l'amplitude de l'intervalle $[a_n, b_n]$ devient plus petit d'un seuil de tolérance bien déterminée ($|b_n - a_n| \leq \varepsilon$) ou la variation des valeurs x_n devient insignifiante ($|x_{n+1} - x_n| \leq \varepsilon$).

Le schéma du calculs peut prendre la forme suivante :

Initialisation : $a_0 = a$, $b_0 = b$, choisir une précision ε ;

Tant que $|b_k - a_k| > \varepsilon \rightarrow$ faire $\downarrow$ (une boucle de calcul à refaire tant que la condition est vérifiée) :

Calculer $x_k = \frac{a_k+b_k}{2}$ et :

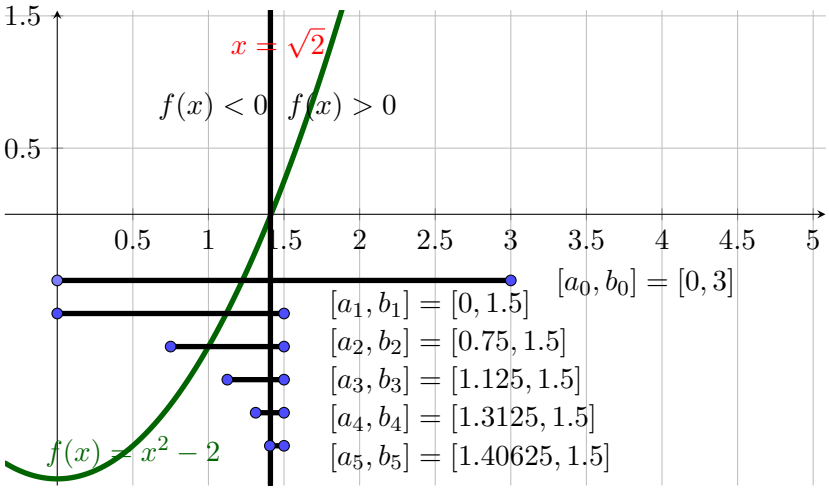
Si $f(a_k)f(x_k) < 0$ Alors $a_{k+1} := a_k$ et $b_{k+1} := x_k$
Sinon $a_k := x_k$ et $b_{k+1} := b_k$
Si $|b_n - a_n| \leq \varepsilon \rightarrow$ Fin.
Conclusion : $x_n = \frac{a_n+b_n}{2}$ est une valeur approchée de $\overline{x}$ avec une précision ε.

Exemples 1.2.1
Soit f la fonction continue sur $\mathbb{R}$ définie par :

$$f(x) = x^2 - 2$$

L'équation $f(x) = 0$ admet deux racines $-\sqrt{2}$ et $\sqrt{2}$ cherchant une valeur approchée de $\sqrt{2}$ en utilisant la méthode de Dichotomie.
- Localisation de la racine positive : *On a $f(0) = -2$ et $f(3) = 7$, la fonction est continue et croissante sur $[0, 3]$, donc l'équation $f(x) = 0$ admet une seule solution $\overline{x} \in [0, 3]$.*
- Initialisation : $a_0 = 0$, $b_0 = 3$ *et* $x_0 = \frac{a_0+b_0}{2} = \frac{3}{2} = 1,5$.
- Étape 1 : *On a :* $f(1,5) = 2,25 - 2 = 0,25$ *est de signe contraire à* $f(a_0)$ *donc :* $a_1 = a_0 = 0$, $b_1 = x_0 = 1,5$ *et* $x_1 = \frac{1,5+0}{2} = 0,75$.
- Étape 2 : *On a :* $f(0,75) = 0,5625 - 2 = -1,4375$ *est de même signe que* $f(a_1)$ *donc :* $a_2 = x_1 = 0,75$, $b_2 = b_1 = 1,5$ *et* $x_2 = \frac{1,5+0,75}{2} = 1,125$.
- Étape 3 : *On a :* $f(1,125) = 1,265625 - 2 = -0,734375$ *est de même signe que* $f(a_2)$ *donc :* $a_3 = x_2 = 1,125$, $b_3 = b_2 = 1,5$ *et* $x_3 = \frac{1,5+1,125}{2} = 1,3125$.
On a $|b_3 - a_3| = |1,5 - 1,125| = 0,375$ *donc :* $1,325$ *est une valeur approchée de la solution* $\sqrt{2}$ *avec une précision de l'ordre* 10^{-1}.

Les étapes peuvent être observées sur la représentation suivante :



Remarques 1.2.2 *Remarquons que :*

- *Différents tests d'arrêt sont possibles : trouver une majoration de l'erreur définie par $|x_n - \overline{x}| \leq \varepsilon$ ou choisir un seuil pour la différence entre deux valeurs consécutives $|x_{n+1} - x_n| \leq \varepsilon$ ou déterminer préalablement le nombre des étapes à exécuter (appelées itérations),...*

- *Pour la méthode de dichotomie une majoration de l'erreur commise à l'étape (ou itération) n est donnée par :*

$$|x_n - \overline{x}| \leq \frac{1}{2^n}|b-a|$$

Cette majoration est du au fait que à chaque itération, l'intervalle est coupé en deux!! et elle ne peut pas être généralisée aux autres méthodes ;
- *Avec cette formule, valable pour la méthode du dichotomie, on peut déterminer le nombre minimum d'itérations nécessaires pour obtenir une précision donnée ;*
- *Pour obtenir une précision à 10^{-2}, il faudrait au moins 8 itérations de dichotomie ;*
- *La méthode de dichotomie converge toujours (mais la convergence est linéaire) ;*
- *L'erreur à chaque étape est divisée par 2 ce qu'induit, généralement, un nombre élevé d'itérations si on souhaite obtenir une bonne précision.*

Remarques 1.2.3 ***Algorithmes.***

L'algorithme de la méthode peut être écrit de différentes méthodes suivant les données et les résultats attendus :

Algorithme 1

On suppose qu'une racine unique, de l'équation $f(x) = 0$ a été localiser dans un intervalle $[a, b]$ et ε précision préalablement fixée :

$a_0 = a$, $b_0 = b$ *et* $x_k = \frac{a_k+b_k}{2}$

While *(Tant que)* $|b_k - a_k| > \varepsilon$ *Test d'arrêt avec l'amplitude de l'intervalle (ou* $|f(x_k)| > \varepsilon$ *Test d'arrêt avec la valeur de la fonction)* ***Do*** *(faire) :*

$x_k = \frac{a_k+b_k}{2}$

If *(si)* $f(a_k) \times f(x_k)$ 0 ***than*** *(alors)* $a_{k+1} = x_k$, $b_{k+1} = b_k$ ***else*** *(sinon)* $a_{k+1} = a_k$, $b_{k+1} = x_k$ ***end if*** *(fin de la condition "si")*

end while *(fin de la boucle "tant que")*

$x_{k+1} = \frac{a_k+b_k}{2}$ *est la valeur, approchée, recherchée*

L'algorithme 1, calcule et stocke (conserve) toutes les valeurs des suites a_k, b_k et x_k et donc utilise plus d'espace mémoire !

Algorithme 2

On suppose qu'une racine unique, de l'équation $f(x) = 0$ a été localiser dans un intervalle $[a, b]$ et ε précision préalablement fixée :

$c = \frac{a+b}{2}$

While $|b - a| > \varepsilon$ *(ou* $|f(c)| > \varepsilon$*)* ***do*** *:*

$c = \frac{a+b}{2}$

If $f(a) \times f(c) > 0$ ***than*** $a := c$ ***else*** $b := c$

end if

end while

$c = \frac{a+b}{2}$ *est la valeur, approchée, recherchée.*

L'algorithme 2, calcule et écrase (supprime et remplace) les valeurs a, b et c, à chaque itération, et donc utilise moins d'espace mémoire !

Nous pouvons ajouter d'autres algorithmes en utilisant différents tests d'arrêt.

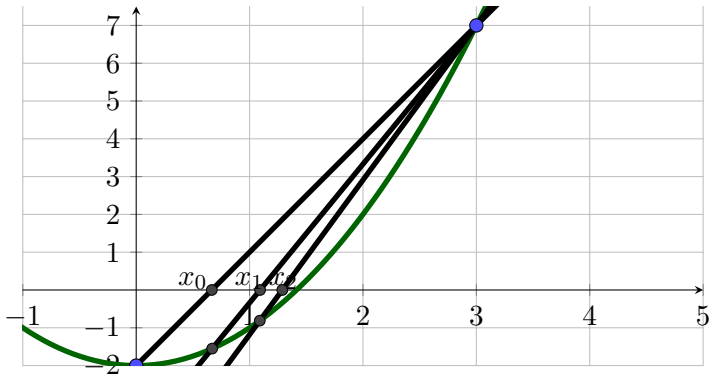
Lorsque on souhaite exécuter exactement N itérations, on utilise la boucle : ***for*** $k = 1$ ***to*** $k = N$ ***do*** *(de... à... faire...).*

1.3 Méthode de Lagrange

La méthode permet la recherche d'une valeur approchée de la racine $\overline{x}$ de l'équation $f(x) = 0$ dans l'intervalle $[a, b]$ ($\overline{x}$ est la seule racine dans $[a, b]$) avec une précision ε. La méthode est une généralisation de la méthode de dichotomie, la valeur de x_k est déterminée comme intersection de la droite qui passe par les deux points $(a_k, f(a_k))$ et $(b_k, f(b_k))$ et l'axe des abscisses et non pas comme centre de l'intervalle $[a_k, b_k]$. Le reste de la démarche est identique à la méthode de Dichotomie :

Initialisation : $a_0 = a$, $b_0 = b$
Tant que $|b_k - a_k| > \varepsilon \rightarrow$ faire $\downarrow$ (une boucle de calcul à refaire tant que la condition est vérifiée) :
Calculer $x_k = a_k - \frac{b_k - a_k}{f(b_k) - f(a-k)} f(a_k) = \frac{a_k f(b_k) - b_k f(a_k)}{f(b_k) - f(a_k)}$ et :
Si $f(a_k) f(x_k) < 0$ Alors $a_{k+1} := a_k$ et $b_{k+1} := x_k$
Sinon $a_k := x_k$ et $b_{k+1} := b_k$
Si $|b_n - a_n| \leq \varepsilon \rightarrow$ Fin.
Conclusion : $x_n = a_n - \frac{b_n - a_n}{f(b_n) - f(a_n)} f(a_n) = \frac{a_n f(b_n) - b_n f(a_n)}{f(b_n) - f(a_n)}$ est une valeur approchée de $\overline{x}$ avec une précision ε.

Illustration des trois premiers itérations, avec la méthode de Lagrange, pour l'exemple précédent ($f(x) = x^2 - 2 = 0$) sur $[0, 3]$.



Remarques 1.3.1 ***Algorithmes :***
Nous présentons deux écritures algorithmiques de cette méthode. A noter que pouvons écrire d'autres en modifiant le test d'arrêt.
Algorithme 1 :
$a_0 = a$, $b_0 = b$ *et* $x_k = \frac{a_k f(b_k) - b_k f(a_k)}{f(b_k) - f(a_k)}$
While $|b_k - a_k| > \varepsilon$ *(ou* $|f(x_k)| > \varepsilon$*)* ***Do*** *:*
$x_k = \frac{a_k f(b_k) - b_k f(a_k)}{f(b_k) - f(a_k)}$
If $f(a_k) \times f(x_k)$ 0 ***than*** $a_{k+1} = x_k$, $b_{k+1} = b_k$ ***else*** $a_{k+1} = a_k$, $b_{k+1} = x_k$ ***end if***
end while
$x_{k+1} = \frac{a_k f(b_k) - b_k f(a_k)}{f(b_k) - f(a_k)}$ *est la val Algorithme 2 :*
On suppose qu'une racine unique, de l'équation $f(x) = 0$ *a été localiser dans un intervalle* $[a, b]$ *et* ε *précision préalablement fixée :*
$c = \frac{af(b) - bf(a)}{f(b) - f(a)}$

While $|b-a| > \varepsilon$ *(ou* $|f(c)| > \varepsilon$*)* ***do*** *:*
$c = \frac{af(b)-bf(a)}{f(b)-f(a)}$
If $f(a) \times f(c) > 0$ ***than*** $a := c$ ***else*** $b := c$
end if
end while
$c = \frac{af(b)-bf(a)}{f(b)-f(a)}$ *est la valeur, approchée, recherchée.*

1.4 Méthode de Newton

L'introduction de la méthode de Newton, nécessite de rappeler la définition d'une approximation affine :

Définition 1.4.1
Soit f *une fonction continue sur* $[a, b]$*, derivable autant de fois que nécessaire. Soit* $\alpha_0 \in [a, b]$*, une approximation affine de* f *autour de* α_0 *est donnée par la tangente d'équation :*

$$y = g(x) = f(\alpha_0) + (x - \alpha_0)f'(\alpha_0) \tag{1.2}$$

Cette tangente coupe l'axe des abscisses au point de coordonnes $(\alpha_0 - \frac{f(\alpha_0)}{f'(\alpha_0)}, 0)$.

Lorsque l'équation $f(x) = 0$ admet une seule racine dans l'intervalle $[a, b]$, la méthode de Newton consiste à utiliser cette approximation affine pour trouver une valeur approchée de la racine $\overline{x}$. en effet, on pose :

$$\alpha_1 = \alpha_0 - \frac{f(\alpha_0)}{f'(\alpha_0)}$$

et on renouvelle la même démarche avec α_1 pour construire α_2 et ainsi de suite pour construire une suite α_n qui peut converger vers $\overline{x}$ (lorsque les conditions de convergence sont vérifiées !).

Soit $f : \mathbb{R} \to \mathbb{R}$ une fonction donnée telle que l'équation $f(x) = 0$ admet une seule solution. La méthode de Newton consiste à exécuter les étapes suivantes :

- Initialisation : x_0 bien choisie dans l'intervalle d'étude (la convergence de la méthode dépend aussi de ce choix) ;
- à partir du terme x_k, on construit x_{k+1} comme étant l'intersection de l'axe des abscisses et la tangente T_{x_k} à la courbe de la fonction f qui passe par le point $(x_k, f(x_k))$. On a :

$$(T_{x_k}) : y = g(x) = f(x_k) + (x - x_k)f'(x_k)$$

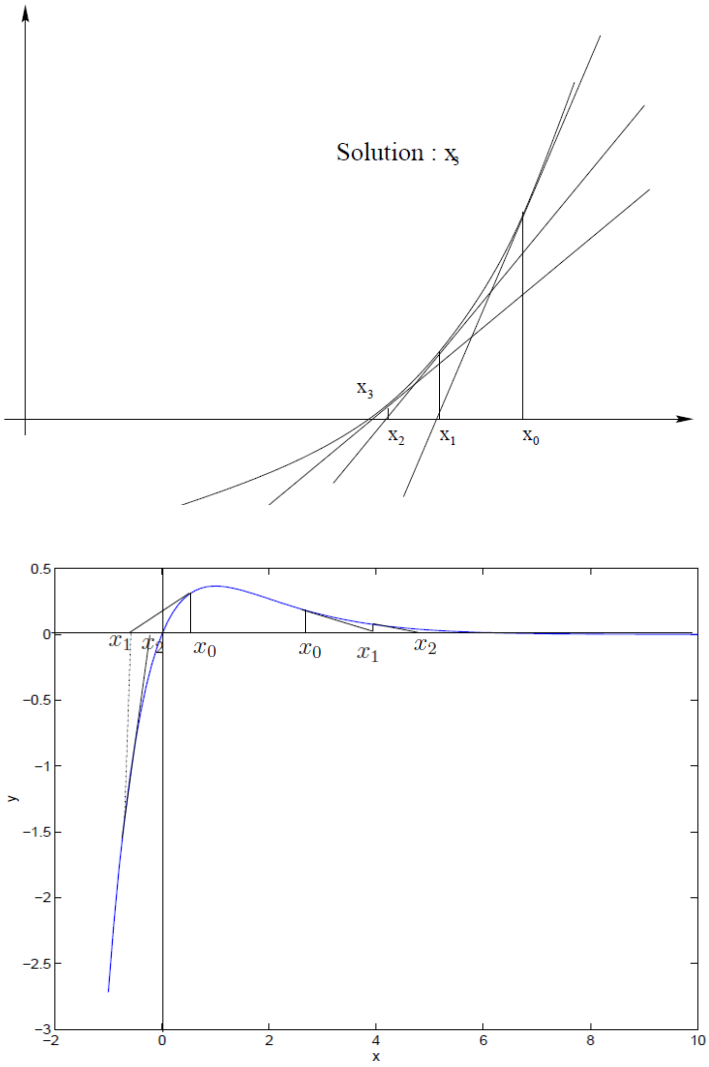
et

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

- Vérifier les conditions de convergence qui garantissent que la suite $(x_k)_k$ converge vers la racine $\overline{x}$;
- préciser le test d'arrêt pour choisir une valeur approchée parmi les termes de la suite $(x_k)_k$.

La représentation ci-après donne une illustration de la méthode de construction des premiers termes de la suite $(x_k)_k$ en utilisant la méthode de Newton :

L'exemple ci-après montre l'impact du choix de x_0 sur les autres termes et sur la convergence de la méthode :



Remarques 1.4.2

- *Lorsque la méthode de Newton converge, la convergence est beaucoup plus rapide que la méthode de dichotomie,*
- *La convergence n'est pas automatique, elle dépend des conditions initiales comme le montre l'example suivant : Soit* $f(x) = xe^{-x}$ *l'équation* $f(x) = 0$ *admet* $x = 0$ *comme racine, mais la méthode converge pour* $0 < x_0 < 1$ *et ne converge pas (diverge) pour* $x_0 > 1$

La méthode de Newton peut engendrer une suite $(x_n)_n$ qui **ne converge pas vers la solution** $\overline{x}$. Il existe plusieurs résultats qui donnent des conditions nécessaires et suffisantes de convergence de la méthode de Newton. La proposition suivante présente des conditions largement suffisantes pour assurer convergence de cette méthode :

Proposition 1.4.3

- *La fonction* f *est définie et continue sur un intervalle* $[a, b]$ *qui contient une racine* $\overline{x}$ *et un point d'initialisation* x_0,

- *La fonction f est deux fois derivable sur l'intervalle $[a, b]$,*
- *La fonction dérivée f' ne s'annule pas sur cet intervalle (f est monotone),*
- *la dérivée seconde f'' est continue et ne s'annule pas (f n'a pas de point d'inflection ;*
- *$f(x_0)$ est de même signe que $f''(x)$.*
  Sous ces conditions la suite $(x_n)_n$ converge vers $\overline{x}$.
- *Pour obtenir une solution approchée avec une précision ϵ, il suffit d'exécuter la méthode en choisissant un test d'arrêt, pare exemple :*
  - *Test basé sur résidu : $|f(x-k)| \leq \varepsilon$*
  - *Test basé sur d'incrément : $|x_k - x_{k-1}| \leq \epsilon$*

Exemples 1.4.4

Soit f la fonction continue sur $\mathbb{R}$ définie par : $f(x) = x^2 - 2$, f est deux derivable sur $\mathbb{R}$ en particulier sur $[0, 3]$ et :

$$f'(x) = 2x \text{ et } f''(x) = 2$$

Cherchons une approximation de la solution $\overline{x} = \sqrt{2}$ en utilisant successivement les deux valeurs initiales $x_0 = 0,5$ et $x_0 = 2,5$

Traitons séparément les deux cas :

- Pour $x_0 = 2,5$. On a $f(x_0) \geq 0$, $f'(x) > 0$ et $f''(x) > 0$ sur $]0, 3[$ alors, les conditions de la proposition précédente sont vérifiées et la méthode est convergente.
  -
  $$x_1 = x_0 - \frac{x_0^2 - 2}{2x_0} = 2,5 - \frac{2,5^2 - 2}{2 \times 2,5} = 1,65$$
  -
  $$x_2 = x_1 - \frac{x_1^2 - 2}{2x_1} = 1,65 - \frac{1,65^2 - 2}{2 \times 1,65} = 1,431061$$
  On a : $|x_2 - x_1| = 0,218939$ et $|f(x_2)| = 0,047935585721$, en deux itérations nous avons obtenu une approximation avec une bonne précision (de l'ordre de 10^{-1} ou 10^{-2} suivant le test d'arrêt choisi !)

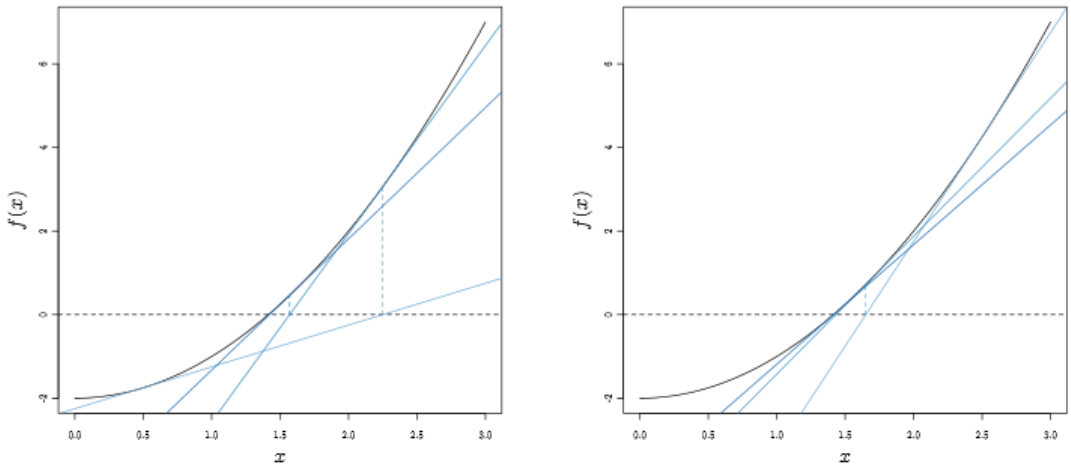
- Pour $x_0 = 0,5$. On a $f(x_0) \leq 0$, $f'(x) > 0$ et $f''(x) > 0$ sur $]0, 3[$ alors, les conditions de la proposition précédente ne sont pas vérifiées et à ce niveau nous ne pouvons pas affirmer que la méthode est convergente. D'autre part, le de x_1 donne :
  -
  $$x_1 = x_0 - \frac{x_0^2 - 2}{2x_0} = 0,5 - \frac{0,5^2 - 2}{2 \times 0,5} = 2,25$$
  Cette fois, on a : $f(x_1) \geq 0$ et nous pouvons dire que la méthode converge en considérons x_1 comme premier terme.
  -
  $$x_2 = x_1 - \frac{x_1^2 - 2}{2x_1} = 2,25 - \frac{2,25^2 - 2}{2 \times 2,25} = 1,569444$$

-

$$x_3 = x_2 - \frac{x_2^2 - 2}{2x_2} = 1,569444 - \frac{1,569444^2 - 2}{2 \times 1,569444} = 1,42189$$

On a : $|x_3 - x_2| = 0,147554$ et $|f(x_3)| = 0,0217711721$, en trois itérations nous avons obtenu une approximation avec une bonne précision (de l'ordre de 10^{-1} ou 10^{-2} suivant le test d'arrêt choisi !)

La figure ci après montre, graphiquement, la construction des points x_k avec le choix des deux valeurs de x_0.



(a) x_0 0.5 (b) x_0 2.5

Remarques 1.4.5

- *L'application de la méthode de Newton demande le calcul de la dérivée f' et éventuellement de la dérivée seconde f'' (si elles existent ! !)*
- *Lorsque la fonction n'est pas derivable ou que le calcul de la dérivée est difficile, nous pouvons remplacer $f'(x_k)$ par le rapport de variation entre les points d'abscisses x_{k-1} et x_k :*

$$\frac{f(x_k) - f(x_{k-1})}{x_k - x_{k-1}}$$

Ainsi, nous obtenons la méthode de la sécante qui est une variate de la méthode de Newton et se présente comme suit :

- *Initialisation : x_0 et x_1 dans le même intervalle qui contient la racine,*
-

$$x_{k+1} = x_k - f(x_k) \times \frac{x_k - x_{k-1}}{f(x_k) - f(x_{k-1})}$$

- *la tangente est remplacée par la droite qui passe par les points $(x_k, f(x_k))$ et $(x_{k-1}, f(x_{k-1}))$.*

Exemples 1.4.6

Soit f la fonction continue sur $\mathbb{R}$ définie par : $f(x) = x^2 - 2$, les premières étapes d'application de la méthode de la sécante :

Initialisation : *dans l'intervalle* $[0,3]$ *soient* $x_0 = 0,5$ *et* $x_1 = 2,5$

x_k	$f(x_k)$
$x_0 = 0,5$	$f(x_0) = 0,5^2 - 2 = -1,75$
$x_1 = 2,5$	$f(x_1) = 2,5^2 - 2 = 4,25$
$x_2 = x_1 - f(x_1) \times \frac{x_1 - x_0}{f(x_1) - f(x_0)} =$ $2,5 - 4,25 \times \frac{2,5-0,5}{4,25+1,75} = 1,4166$	...

Remarques 1.4.7

Remarquons, qu'il existe plusieurs variantes de la méthode de Newton :

- *Dans la méthode de la sécante, on peut remplacer* x_{k-1} *par* b *pour obtenir une variante ou chaque valeur* x_k *dépend seulement de la valeur précédente* x_{k+1} *et non pas deux valeurs :*

$$x_{k+1} = x_k - f(x_k) \times \frac{x_k - b}{f(x_k) - f(b)}$$

- ***(Méthode de la corde)*** *Dans la méthode de Newton, on remplace* $f'(x_k)$ *par une constante* q *(souvent* $q = f'(x_0)$ *ou* $q = \frac{f(b)-f(a)}{b-a}$*),*
- *La méthode de Newton est appelée dans plusieurs références : méthode de* ***Newton-Raphson****.*

1.5 Méthode de point fixe

Proposition 1.5.1

- *Soit* g *une fonction continue définie sur un intervalle* $[a,b]$ *et vérifiant* $g([a,b]) \subset [a,b]$ *(i.e.* $g(a) \geq a$ *et* $g(b) \leq b$*). Alors il existe au moins* $\alpha \in [a,b]$ *tel que :*

$$g(\alpha) = \alpha$$

  α *est dit point fixe de la fonction* f *dans* $[a,b]$*. Si, en plus, la fonction* f *est monotone alors* α *est unique.*
- *La résolution de l'équation* $f(x) = 0$*, sur un intervalle* $[a,b]$*, peut être transformée à un problème de recherche d'un point fixe d'une fonction auxiliaire* $g : [a,b] \to \mathbb{R}$ *telle que :*

$$g(x) = x \Leftrightarrow f(x) = 0$$

Nous avons besoin des définitions suivantes :

Définition 1.5.2

Soit $g : \mathbb{R} \to \mathbb{R}$ *une application continue*

- $\overline{x} \in \mathbb{R}$ *est dit point fixe de* g *s'il vérifie* $g(\overline{x}) = \overline{x}$*,*
- k *un réel positif,* g *est dite* **k-*lipschitzienne*** *si :*

$$|f(y) - f(x)| \leq k|y - x| \text{ pour tout } x, y \in \mathbb{R}$$

- g *est dite* ***contractante*** *s'il existe* $k \in [0,1[$ *tel que* g *est* **k**-*lipschitzienne :*

$$\text{Il existe } k \in [0,1[\text{ tel que } |f(y) - f(x)| \leq k|y - x| \text{ pour tout } x, y \in \mathbb{R}$$

Exemples 1.5.3

- *La fonction constante $f(x) = a$ est 0-lipschitzienne sur $\mathbb{R}$,*
- *La fonction affine $f(x) = ax + b$ est a-lipschitzienne sur $\mathbb{R}$,*
- *$f(x) = x^2$ est lipschitzienne sur tout intervalle $[a, b]$ (k dépend de l'intervalle) et elle n'est pas lipschitzienne sur $\mathbb{R}$,*
- *$f(x) = x^3$ est lipschitzienne sur tout intervalle $[a, b]$ (k dépend de l'intervalle) et elle n'est pas lipschitzienne sur $\mathbb{R}$,*
- *$f(x) = \sqrt{x}$ est lipschitzienne sur tout intervalle $[a, +\infty]$ avec $a > 0$ (k dépend de l'intervalle) et elle n'est pas lipschitzienne sur $[0, 1]$,*
- *$f(x) = \frac{1+x}{2+x}$ est contractante ($k = \frac{1}{4}$) sur $[0, +\infty]$.*

Le résultat suivant montre l'existence d'un point fixe pour les fonctions contractante :

Proposition 1.5.4

Soit $g : \mathbb{R} \to \mathbb{R}$ une fonction contractante, Alors :

- *g admet un point fixe unique $\overline{x} \in \mathbb{R}$,*
- *et, pour tout point initial x_0, la suite itéré (ou récurrente) $(x_n)_n$ définie par $x_{n+1} = g(x_n)$ converge vers $\overline{x}$.*

Faisant les remarques suivantes :

Remarques 1.5.5

- *On peut localiser la racine et travailler sur un intervalle $[a, b]$ on considérant $g : [a, b] \to [a, b]$ et $x_0 \in [a, b]$;*
- *La méthode de Newton est une méthode de point fixe, il suffit de considérer la fonction g définie par :*

$$g(x) = x - \frac{f(x)}{f'(x)}$$

- *Les méthode de dichotomie et de Lagrange ne peuvent pas être décrites comme méthode de point fixe, car x_{n+1} ne s'écrit pas comme $g(x_n)$ mais plutôt en fonction de a_k et b_k.*

Exemples 1.5.6

Soit g définie sur $[0, +\infty]$ par :

$$g(x) = \frac{1+x}{2+x}$$

On sait que g est contractante et que pour tout $x \geq 0$, on a $g(x) > 0$ et $g([0, +\infty]) \subset [0, +\infty]$. On applique la méthode de pont fixe pour résoudre l'équation $g(x) = x$:

$$x_0 = 4 \text{ et } x_{n+1} = g(x_n)$$

$$x_1 = g(x_0) = \frac{5}{7} \text{ et } x_2 = g(x_1) = \frac{12}{19}$$

$$x_3 = g(x_2) = \frac{31}{50} = 0.62 \text{ et } x_4 = g(x_3) = \frac{81}{131} \simeq 0.6183...$$

Solution exacte :

$$g(x) = x \Leftrightarrow \frac{1+x}{2+x} = x \Leftrightarrow 1 + x = 2x + x^2 \Leftrightarrow -x^2 - x + 1 = 0$$

$$\textit{Deux solutions } \frac{-\sqrt{5}-1}{2} \textit{ et } \frac{\sqrt{5}-1}{2} \textit{ dont } \frac{\sqrt{5}-1}{2} \simeq 0.0743 \in [0,+\infty]$$

On a :

$$\lim_{n\to+\infty} x_n = \frac{\sqrt{5}-1}{2}$$

Dans ce dernier paragraphe, nous donnons un aperçu sur des notions qui jouent un rôle important dans le choix d'une méthode ou l'autre :

1.6 Convergence - Erreur - ordre de convergence

Définition 1.6.1

- *Une méthode numérique de résolution de l'équation ($f(x)=0$ ou $g(x)=x$) est dite convergente si elle définie une suite $(x_n)_n$ telle que : $\lim_{n\to+\infty} x_n = \overline{x}$ ou $\overline{x}$ est la solution exacte de l'équation*
- *à l'étape n, l'erreur absolue est donnée par : $r_n = |x_n - \overline{x}|$. L'erreur relative est égale à :*

$$\frac{r_n}{\overline{x}} = \frac{|x_n-\overline{x}|}{\overline{x}}$$

- *La méthode est dite d'ordre p si :*

$$\lim_{n\to+\infty} \frac{|r_{n+1}|}{|r_n|^p} = \frac{|x_{n+1}-\overline{x}|}{|x_n-\overline{x}|^P} > 0$$

- *La solution exacte $\overline{x}$ étant inconnue, on utilise des outils Mathématiques pour chercher une majoration de l'erreur :*

$$r_n = |x_n - \overline{x}| \leq \rho_n \textit{ avec } \rho_n \textit{ ne dépend pas de } \overline{x}$$

- *$p=1$: la convergence est dite linéaire (Exemple : Méthode de dichotomie)*
- *$p=2$: la convergence est dite quadratique (Exemple : Méthode de Newton)*

La majoration de l'erreur est parmi les étapes les plus difficiles et parfois on utilise des notions avancées. A ce niveau, si nous ne pouvons pas trouver une majoration de l'erreur, nous pouvons utiliser les tests d'arrêt en relation avec la précision souhaitée :

Remarques 1.6.2

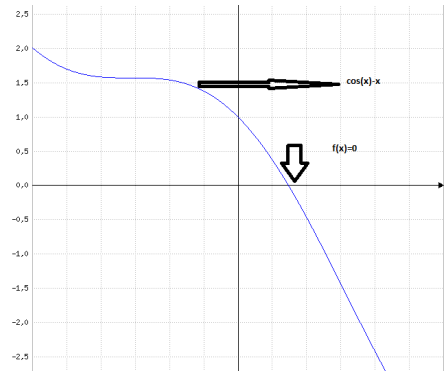
1) Majoration de l'erreur absolue par une quantité qui ne dépends pas de la racine recherchée $\overline{x}$ ni de x_k, par exemple, dans les méthodes de Dichotomie et de la Lagrange on a :

$$|x_k - \overline{x}| \leq |b_k - a_k|$$

Il suffisant de choisir $|b_k - a_k| \leq \varepsilon$ pour obtenir la précision recherchée

2) Test basé sur le résidu : $|f(x_k)| \leq \varepsilon$

3) Test basé sur l'incrément : $|x_{k+1} - x_k| \leq \varepsilon$
Suivant la situation et les conditions initiales, chaque test peut être considéré comme satisfaisant ou trop restrictif.



Zéros de fonctions non-linéaires Etude de cas On cherche à résoudre l'équation dans $[0, +\infty[$: $\cos(x) - x = 0$
Solution exacte : aucune méthode (à ce niveau),
Méthode numérique : Localisation d'une racine : $f(x) = \cos(x) - x$

Zéros de fonctions non-linéaires Etude de cas f est continue sur $\mathbb{R}$,
$f(0,5) = \cos(0,5) - 0,5 > 0$ et $f(1) = \cos(1) - 1 < 0$.
Donc, d'après le théorème des VI, l'équation admet une solution dans $[0,5; 1]$. Pour obtenir une valeur approchée, il faut utiliser une méthode numérique :
1) Méthodes pour résoudre $f(x) = 0$: Dichotomie, Newton ou l'une des variantes,
2) Méthode de point fixe : $g(x) = x$ ou $g(x) = \cos(x)$.

Peut-on appliquer la méthode de point fixe ? quelles conditions ? ?

g est elle contractante ? ?

Zéros de fonctions non-linéaires Etude de cas g est continue et derivable avec : $g'(x) = -\sin(x)$.
On a

$$0 \leq x \leq 1 \Rightarrow 0 \leq \cos(1) \leq \cos(x) \leq 1 \Rightarrow g([0,1]) \subset [0,1]$$

et

$$0 \leq x \leq 1 \Rightarrow |g'(x)| = |-\sin(x)| \leq \sin(1)$$

Donc, d'après l'inégalité des AF, on a pour tout $x, y \in [0,1]$:

$$|g(x) - g(y)| \leq sin(1)|x - y|$$

On en déduit que g est $sin(1)$-contractante sur $[0,1]$

Zéros de fonctions non-linéaires Etude de cas La méthode de point fixe nous permet de construire une suite $(x_n)_n$ telle que :
- x_0 est donnée (par exemple 0.5 centre de l'intervalle $[0,1]$)
- $x_{n+1} = g(x_n)$
En plus :

$$\lim_{n \to +\infty} x_n = \overline{x} \text{ ou } \overline{x} \text{ est solution de } g(x) = x$$

Remarque : La méthode ne permet pas le calcule de la solution exacte $\overline{x}$ mais une valeur approchée x_k avec une précision donné.

n	Précision inférieure à
1 ≈ 0,877582...	0,433
2 ≈ 0,639012...	0,375
3 ≈ 0,802685...	0,324
5 ≈ 0,768195...	0,243
7 ≈ 0,752355...	0,182
10 ≈ 0,7535006...	0,118
16 ≈ 0,738704...	0,050
27 ≈ 0,739081...	0,008
44 ≈ 0,739085...	0,0008

Zéros de fonctions non-linéaires Estimation de l'erreur Démontrons par récurrence que, pour tout n, $r_n = |x_n - \overline{x}| \leq k^n|x_0 - \overline{x}|$.

Pour $n = 1$, on a : $|x_1 - \overline{x}| = |g(x_0) - g(\overline{x})| \leq k|x_0 - \overline{x}|$.

Hypothèse de récurrence : $|x_n - \overline{x}| \leq k^n|x_0 - \overline{x}|$,

Pour $n + 1$ on a :

$$|x_{n+1} - \overline{x}| = |g(x_n) - g(\overline{x})| \leq k|x_n - \overline{x}| \leq kk^n|x_0 - \overline{x}| = k^{n+1}|x_0 - \overline{x}|$$

On en déduit que :

$$r_n = |x_n - \overline{x}| \leq (\sin(1))^n|0,5 - \overline{x}| \leq \frac{1}{2}(\sin(1))^n \leq \frac{1}{2}(\frac{\sqrt{3}}{2})^n = \frac{(\sqrt{3})^n}{2^{n+1}}$$

Car : $\sin(1) \leq \sin(\frac{\pi}{3}) = \frac{\sqrt{3}}{2}$ et $|0,5 - \overline{x}| \leq \frac{1}{2}$ ($\overline{x} \in [0,5; 1]$).

Zéros de fonctions non-linéaires Estimation de l'erreur

Zéros de fonctions non-linéaires Systèmes d'équations non-linéaires L'objectif est de résoudre le système suivant :

$$\begin{cases} f_1(x_1, x_2, ..., x_n) = 0 & ; \\ f_2(x_1, x_2, ..., x_n) = 0 & ; \\ ... & ; \\ f_n(x_1, x_2, ..., x_n) = 0 & . \end{cases}$$

x_1, x_2,... et x_n sont des inconnues et $f_i : \mathbb{R}^n \to \mathbb{R}$

On pose $F : \mathbb{R}^n \to \mathbb{R}^n$ telle que $F(X) = (f_1(X), f_2(X), ..., f_n(X))$ ou $X = (x_1, x_2, ..., x_n)$ et le système peut s'écrire :

$$F(X) = 0$$

Quelques algorithmes du cas réel ($n = 1$) peuvent être generalisés au cas vectoriel.

Zéros de fonctions non-linéaires Systèmes à deux équations non linéaires : Ce système s'écrit :

$$\begin{cases} f_1(x_1, x_2) = 0 & ; \\ f_2(x_1, x_2) = 0 & . \end{cases}$$

D'une manière équivalente :

$$F(X) = 0 \text{ ou } X = (x_1, x_2) \text{ et } F : \mathbb{R}^2 \to \mathbb{R}^2 \text{ telle que :}$$

$$F(X) = (f_1(x_1, x_2), f_2(x_1, x_2))$$

La notion de dérivée est remplacée par **la matrice Jacobienne** de F définie par :

$$DF(X) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{pmatrix}$$

Zéros de fonctions non-linéaires Systèmes à deux équations non linéaires : **Méthode Newton adaptée au cas vectoriel :**

- Initialisation : $X^{(0)} = (x_1^{(0)}, x_2^{(0)})$
- La méthode définit une suite $(X(n))_n$ telle que :

$$[DF(X^{(n)})](X^{(n+1)} - x^{(n)}) = -F(X^{(n)}) \text{ équation matricielle ! !}$$

- Les mêmes résultats que dans le cas scalaire sont valables dans le cas vectoriel.
- Le problème se ramène ainsi à la résolution des systèmes linéaires dont les matrices sont données par :

$$A_n = DF(X^{(n)})$$

Chapitre 2

Interpolation polynômiale

2.1 Approximation et Interpolation Polynômiale

Position du problème - une motivation pratique :

Une expérience au laboratoire a permet de relever les températures T_1, T_2,..., T_n d'une solution au cours de différents instants t_1, t_2,..., t_n. Ces données peuvent être représentées graphiquement (en un ensemble de points).

D'autre part, la théorie chimique permet d'exprimer la temperature comme fonction du temps : $T = f(t)$. La forme explicite de la fonction f peut être non déterminée !,

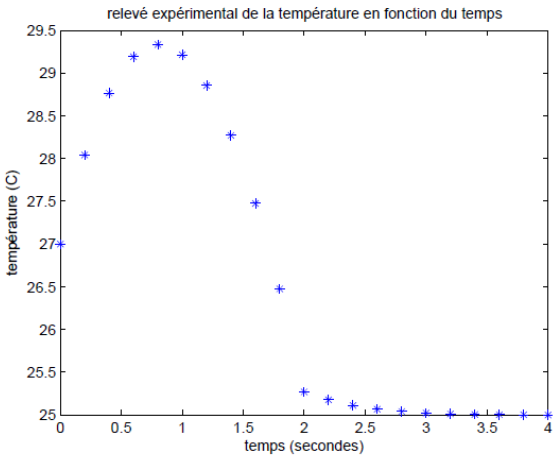
- Si f est connue explicitement, nous pouvons comparer T_i et $f(t_i)$ pour $i = 1, ..., n$ et vérifier si l'expérience a été effectuée dans de bonnes conditions.

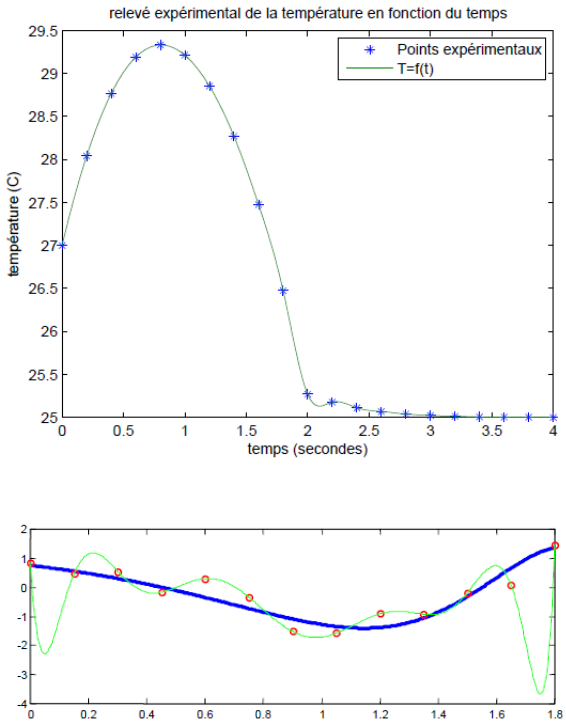
- Si f n'est pas connue explicitement, nous pouvons chercher une approximation de f (comme fonction) vérifiant : $T_i = f(t_i)$ pour $i = 1, ..., n$.

Remarques :

- En général, on cherche une approximation de f par des fonctions simples et régulières (Polynômes, Polynômes par morceaux, polynômes trigonométriques, exceptionnelles,...).

- Si la fonction f est donnée par sa forme explicite, l'approximation polynomiale permet de trouver une fonction "plus





simple" (souvent polynôme P_n) qui coincide avec f en un nombre de points et qui permet plus de "flexibilité" dans l'étude de f.

- Si la fonction f n'est pas donnée par sa forme explicite, l'approximation polynomiale permet de trouver une fonction (souvent polynôme P_n) dont la courbe passe par les points déterminés d'une fonçon expérimentale et qu'elle peut être considérée comme une forme explicite de f.

- Lorsque le courbe de P_n passe par TOUS les points $\rightarrow$ **Interpolation Polynomiale**.

- Lorsque le courbe de P_n ne passe pas par TOUS les points $\rightarrow$ **Approximation Polynomiale**.

- La courbe Bleu $\rightarrow$ Approximation Polynomiale,

- La courbe verte $\rightarrow$ Interpolation Polynomiale.

Outils Mathématiques : - $\mathbb{P}_n[X]$, muni de la somme et la multiplication par un nombre réel, est l'espace vectoriel des fonctions polynômes : $P_n : \mathbb{R} \rightarrow \mathbb{R}$ de degré inférieur ou égal à n. On a : $dim\mathbb{P}_n[X] = n + 1$ et une base de $\mathbb{P}_n[X]$ est donnée par : $\{1, X, X^2, ..., X^n\}$.

- **Position du problème :** Soit $f : [a, b] \rightarrow \mathbb{R}$ une fonction continue sur $[a, b]$ (i.e : $f \in C^0([a, b])$) et $x_0, x_1, ..., x_n \in [a, b]$ $n + 1$ points distincts dans $[a, b]$.

On cherche un polynôme P_n de degré au plus égal à n ($P_n \in \mathbb{P}_n[X]$) tel que : $P_n(x_i) = f(x_i)$ pour $i = 0, 1, ..., n$

Lorsque P_n existe, il est appelé **Polynôme d'interpolation** de f.

Remarques :

- On peut aussi chercher le polynôme d'interpolation aux points (x_i) des $n+1$ valeurs $y_0, y_1, ..., y_n$ en remplaçant $f(x_i)$ par y_i.
- Le polynôme d'interpolation n'est pas forcément de degré égal à n, il peut être inférieur strictement à n,
- Le problème tel que défini à une et une seule solution (voir La Méthode de Lagrange),
- La situation n'est plus la même si on cherche, dans les mêmes conditions, un polynôme de degré $m \neq n$. Dans ce cas, le problème peut avoir plusieurs solutions ou aucune solution !

Méthode d'interpolation de Lagrange :

Pour $x_0, x_1, ..., x_n$, on définit les polynômes caractéristiques de Lagrange (L_i) avec $i = 0, 1, ..., n$ par :

$$L_i(x) = \prod_{j=0, j\neq i}^{j=n} \frac{x - x_j}{x_i - x_j} \text{ avec } i = 0, 1, ..., n$$

On a :
- L_i est de degré n,
- $L_i(x_i) = 1$ et si $j \neq i$ alors $L_i(x_j) = 0$ ce que nous pouvons noter avec le symbole de Kronecker $\delta_{ij} = 1$ si $i = j$ et 0 sinon :

$$L_i(x_j) = \delta_{ij} \text{ avec } i = 0, 1, ..., n \text{ et } j = 0, 1, ..., n$$

- Les $n+1$ polynômes caractéristiques de Lagrange $(L_i)_i$ constituent une base de $\mathbb{P}_n[X]$ (il suffit de montrer que cette famille est libre) ;
- Le polynôme d'interpolation de f aux points $x_0, x_1, ..., x_n$ est donnée par :

$$P_n(X) = \sum_{i=0}^{i=n} f(x_i) L_i(X)$$

- Ce polynôme est unique, en effet, soit $Q_n(X)$ un autre polynôme d'interpolation, alors : $P(x_i) - Q(x_i) = f(x_i) - f(x_i) = 0$;

Le polynôme $P - Q$ de degré n s'annule en $n+1$ points distincts donc $P - Q = 0$ d'ou l'unicité.

Théorème : Existence et unicité

Le problème de trouver un polynôme P_n d'interpolation d'une fonction f aux points $x_0, x_1, ..., x_n$, admet une solution et une seule donnée par :

$$P_n(X) = \sum_{i=0}^{i=n} f(x_i) L_i(X)$$

$(L_i)_i$: les $n+1$ polynômes caractéristiques de Lagrange.

Remarque : Le problème de chercher le polynôme d'interpolation des $n+1$ points $y_0, y_1, ..., y_n$ (données expérimentales par exemple) s'obtient en remplaçant $f(x_i)$ par y_i.

Exemples :

• $n = 1$: Nous avons deux points (x_0, y_0) et (x_1, y_1) et nous cherchons le polynôme de degré 1 (ou la droite qui passe par les deux points) tel que : $y = ax + b$, on a :

$$y_0 = ax_0 + b \text{ et } y_1 = ax_1 + b$$

donc :

$$a = \frac{y_0 - y_1}{x_0 - x_1} \text{ et } b = \frac{x_0 y_1 - x_1 y_0}{x_0 - x_1}$$

On obtient :

$$y = \frac{y_0 - y_1}{x_0 - x_1}x + \frac{x_0 y_1 - x_1 y_0}{x_0 - x_1} = y_0 \underbrace{\frac{x - x_1}{x_0 - x_1}}_{L_0(x)} + y_1 \underbrace{\frac{x - x_0}{x_1 - x_0}}_{L_1(x)}$$

• $n = 2$ Soit f avec fonction telle que : $f(-1) = 8$, $f(0) = 3$ et $f(1) = 6$ cherchons le polynôme d'interpolation de f aux points $x_0 = -1$, $x_1 = 0$ et $x_2 = 1$. On a :

$$L_0(x) = \prod_{j=0, j\neq 0}^{j=2} \frac{x - x_j}{x_0 - x_j} = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)} = \frac{1}{2}(x^2 - x)$$

$$L_1(x) = \prod_{j=0, j\neq 1}^{j=2} \frac{x - x_j}{x_1 - x_j} = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)} = -x^2 + 1$$

$$L_2(x) = \prod_{j=0, j\neq 2}^{j=2} \frac{x - x_j}{x_2 - x_j} = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)} = \frac{1}{2}(x^2 + x)$$

Donc :

$$P_2(x) = 8L_0(x) + 3L_1(x) + 6L_2(x)$$

$$P_2(x) = 8(\frac{1}{2}(x^2 - x) + 3(-x^2 + 1) + 6(\frac{1}{2}(x^2 + x)) = 4x^2 - x + 3$$

Cas ou f est donnée par sa forme explicite : Soit $f(x) = e^x$ et soient les points $x_0 = -1$, $x_1 = 0$ et $x_2 = 1$. Alors, le polynôme d'interpolation de f par rapport aux trois points est donnée par :

$$P_2(x) = e^{-1}L_0(x) + e^0 L_1(x) + e^1 L_2(x)$$

$$P_2(x) = (\frac{e^1 + e^{-1}}{2} - 1)x^2 + \frac{e^1 + e^{-1}}{2}x + 1$$

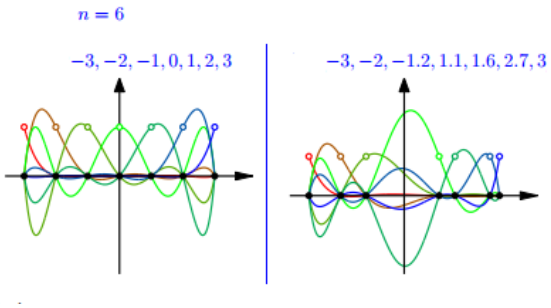
Remarques :

Le problème peut être abordé d'un autre point vue. En effet, P_n peut s'écrire sous la forme :

$$P_n(x) = a_0 + a_1 x + ... + a_{n-1}x^{n-1} + a_n x^n$$

En exploitant les égalités $P_n(x) = f(x_i) = y_i$ pour $i = 0, 1, ..., n$, nous obtenons un système de $n + 1$ équations avec $n + 1$ inconnus :

$$\begin{cases} a_0 + a_1 x_0 + ... + a_{n-1}x_0^{n-1} + a_n x_0^n = y_0 \\ a_0 + a_1 x_1 + ... + a_{n-1}x_1^{n-1} + a_n x_1^n = y_1 \\ ..... \\ a_0 + a_1 x_n + ... + a_{n-1}x_n^{n-1} + a_n x_n^n = y_n, \end{cases}$$



ce système s'écrit :

$$\begin{pmatrix} 1 & x_0 & \dots & x_0^n \\ 1 & x_1 & \dots & x_1^n \\ . & . & & . \\ . & . & & . \\ . & . & & . \\ 1 & x_n & \dots & x_n^n \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ . \\ . \\ . \\ a_n \end{pmatrix} = \begin{pmatrix} y_0 \\ y_1 \\ . \\ . \\ . \\ y_n \end{pmatrix}$$

Ce système admet une solution et une seule mon matrice est une matrice de **Van de Monde** qui inversible car son déterminantégal à :

$$\prod_{1 \le i < j \le n} (xi - x_j) \text{ est non nul car les } x_i \text{ sont distincts !}$$

Ce résultat n'a qu'un caractère théorique : s'il énonce une condition nécessaire et suffisante simple pour que le problème d'interpolation admette une solution unique, il est pratiquement inutilisable en pratique si l'on veut calculer de manière effective le polynôme d'interpolation P_n. Il faut, en effet, résoudre un système linéaire plein.

Soit $P_n(x)$ le polynôme d'interpolation d'une fonction f par rapport aux points x_0, x_1,..., x_n.
et $P'_n(x)$ le polynôme d'interpolation d'une fonction f par rapport aux points a_0, a_1,..., a_n.
Existe-t-elle une relation entre $P_n(x)$ et $P'_n(x)$? ?

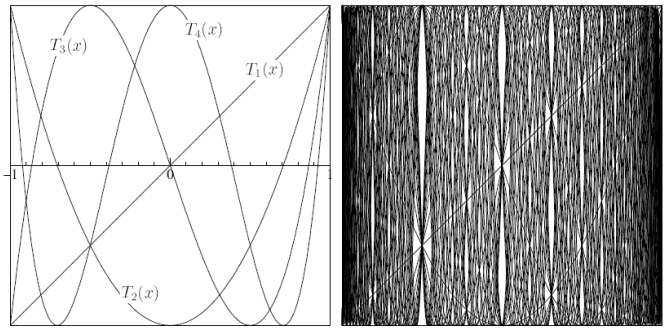
Lorsque on change les points utilisés (appelés Noeuds), le polynôme d'interpolation change aussi !
Exemple : Soit $f(x) = e^x$ et soient les points $x_0 = -1.5$, $x_1 = 0$ et $x_2 = 1.5$.
Polynôme d'interpolation : $P'_2(x) =$

à comparer avec $P_2(x) = (\frac{e^1+e^{-1}}{2} - 1)x^2 + \frac{e^1+e^{-1}}{2}x + 1$.

Comment choisir les points pour obtenir la meilleure interpolation (Le polynôme qui donne la meilleure approximation de la fonction) ?

La construction des noeuds donnant la meilleure interpolation est basée sur les polynômes de Tchebychev.

Polynômes de Tchebychev : - On se propose de répondre au problème suivant :
Existe-il un polynôme rél T_n tel que : $T_n(\cos\theta) = \cos(n\theta)$ pour tout $\theta \in \mathbb{R}$.



- Les polynômes de Tchebychev, définies comme suit, permettent de formuler une réponse à ce problème :

$$T_n(x) = \cos(n \arccos x) \text{ avec } x \in [-1, 1] \text{ et } n \in \mathbb{N}$$

On pose : $\theta = \arccos x$ (d'une manière équivalente : $x = \cos\theta$) avec $\theta \in [0, \pi]$, on a :

$$T_n(x) = \cos(n\theta)$$

et :

$$\begin{aligned} T_{n+1}(x) + T_{n-1}(x) &= \cos((n+1)\theta) + \cos((n-1)\theta) \\ &= \cos(n\theta + \theta) + \cos(n\theta - \theta) \\ &= 2\cos(n\theta)\cos(\theta) \\ &= 2xT_n(x) \end{aligned}$$

Les polynômes de Tchebychev, T_n, sont définies par la formule récurrente suivante :

$$\begin{cases} T_0(x) = 1 \text{ et } T_1(x) = x \\ T_{n+1} = 2xT_n(x) - T_{n-1}(x) \end{cases}$$

Les polynômes de Tchebychev, T_n, vérifient les propriétés suivantes :

- $T_n(x)$ est un polynôme de degré n dont le coefficient de x^n est 2^{n-1},
- $T_n(x)$ est pair si n est pair et impair sinon,
- $|T_n(x)| \leq 1$ pour tout $x \in [-1, 1]$,
- $T_n(\cos(\frac{k\pi}{n})) = (-1)^k$ pour $k = 0, 1,...,n-1$,
- $T_n(\cos(\frac{(2k+1)\pi}{2n})) = 0$ pour $k = 0, 1,...,n-1$. Donc, les racines du polynôme $T_n(x)$ sont données par :

$$\overline{x}_k = \cos(\frac{(2k+1)\pi}{2n}) \text{ avec } k = 0, 1, ..., n-1$$

Remarquons que :

$$\begin{aligned}
\overline{x}_{n-k} &= \cos(\frac{(2(n-k)+1)\pi}{2n}) \\
&= \cos(\frac{(2n-2k+1)\pi}{2n}) \\
&= \cos(\pi - \frac{(2k-1)\pi}{2n}) \\
&= -\cos(\frac{(2k-1)\pi}{2n}) \text{ car } \cos(\pi-\alpha) = -\cos(\alpha) \\
&= -\cos(\frac{(2(k-1)+1)\pi}{2n}) = -\overline{x}_{k-1}
\end{aligned}$$

Les racines $\overline{x}_k$ avec $k = 0, 1,..., n-1$ sont répartis symétriquement autour de zéro.

Propriétés et définitions :

- Les racines $\overline{x}_k = \cos(\frac{(2k+1)\pi}{2n})$ avec $k = 0, 1,..., n-1$ sont appelés : Les abscises de Tchebychev d'ordre n (sur $[-1, 1]$).
- Rappelons que les polynôme de Tchebychev sont définis sur $[-1, 1]$. Pour obtenir les abscises sur un intervalle $[a, b]$ quelconque, il suffit d'utiliser la transformation suivante :

$$x_k = \frac{a+b}{2} + \frac{b-a}{2}\overline{x}_k \text{ avec } k = 0, 1, ..., n-1$$

de point de vue Géométrique, cette transformation est équivalente à une action de translation et de homothétie.

- Les x_k avec $k = 0, 1,..., n-1$ sont appelés : Les abscises de Tchebychev d'ordre n (sur $[a, b]$).

Soit :

$$\mathbb{T}_n(x) = \prod_{k=0}^{k=n-1} (x - x_k)$$

et :

$$\mathbb{T}_n(x) = (\frac{b-a}{2})^n \prod_{k=0}^{k=n-1} (\overline{x} - \overline{x}_k) \text{ avec } x = \frac{a+b}{2} + \frac{b-a}{2}\overline{x}$$

Or,

$$T_n(x) = 2^{n-1} \prod_{k=0}^{k=n-1} (x - \overline{x}_k)$$

Donc :

$$\mathbb{T}_n(x) = (\frac{b-a}{2})^n \frac{1}{2^{n-1}} T_n(\overline{x})$$

Soit :

$$\mathbb{T}_n(x) = 2(\frac{b-a}{4})^n T_n(\overline{x})$$

avec

$$\overline{x} = \frac{2x-(a+b)}{b-a} = \frac{x - \frac{a+b}{2}}{\frac{b-a}{2}}$$

$\mathbb{T}_n(x)$ polynômes de Tchebychev définit sur $[a, b]$: $x \in [a, b] \to \overline{x} \in [-1, 1]$

$T_n(.)$ polynôme de Tchebychev sur $\in [-1, 1]$

de l'erreur avec les polynômes de Tchebychev : Théorème : Soient x_0, x_1, ..., x_n, $n+1$ points distincts dans $[a,b]$ et soit $f \in C^{n+1}([a,b])$. Alors, pour tout $x \in [a,b]$:

$$f(x) - P_n(x) = \frac{1}{(n+1)!}\mathbb{T}_{n+1}(x) f^{(n+1)}(\xi)$$

Où $\mathbb{T}_{n+1}(x) = \prod_{i=0}^{n}(x - x_i)$ et $\xi \in [a,b]$.

Démonstration :

Extension du théorème de Rolle : Soit $f : [a,b] \to \mathbb{R}$ une fonction continue sur $[a,b]$, derivable sur $]a,b[$ et admettant $n > 1$ zéros sur $[a,b]$ notés $x_1 < x_2 < \cdots < x_n$. Alors f' admet au moins $(n-1)$ zéros.

Si $x = x_i$, on a $f(x_i) = P_n(x_i)$ et $\mathbb{T}_{n+1}(x_i) = 0$ donc la propriété est vraie pour tout ξ.
Si $x \neq x_i$, soit $P_{n+1}(t)$ le polynôme d'interpolation de f sur la base des points x, x_0, x_1, ..., x_n. Par construction, f coincide avec P_{n+1} dans les points de d'interpolation, en particulier : $f(x) = P_{n+1}(x)$ Donc :

$$f(x) - P_n(x) = P_{n+1}(x) - P_n(x)$$

Or, $P_{n+1}(x) - P_n(x)$ est un polynôme de degré au plus égal à $n+1$ et s'annule aux $n+2$ points : x_0, x_1, ..., x_n et $x_{n+1} = x$. Donc :

$$P_{n+1}(t) - P_n(t) = c\mathbb{T}_{n+1}(t) \text{ avec } c \in \mathbb{R}$$

Soit la fonction g définie par :

$$g(t) = f(t) - P_{n+1}(t) = f(t) - P_n(t) - c\mathbb{T}_{n+1}(t)$$

La fonction f s'annule en $n+2$ points (x_0, x_1, ..., x_n et $x_{n+1} = x$)
Donc d'après **l'extension du théorème de Rolle**, il existe ξ tel que

$$g^{(n+1)}(\xi) = 0$$

D'où :

$$g^{(n+1)}(\xi) = f^{(n+1)}(\xi) - P_n^{(n+1)}(\xi) - c\mathbb{T}_{n+1}^{(n+1)}(\xi) = f^{(n+1)}(\xi) - c(n+1)! = 0$$

car :

$$P_n^{(n+1)}(\xi) = 0 \text{ et } \mathbb{T}_{n+1}^{(n+1)}(\xi) = (n+1)!$$

On obtient :

$$c = \frac{1}{(n+1)!} f^{(n+1)}(\xi)$$

On en déduit :

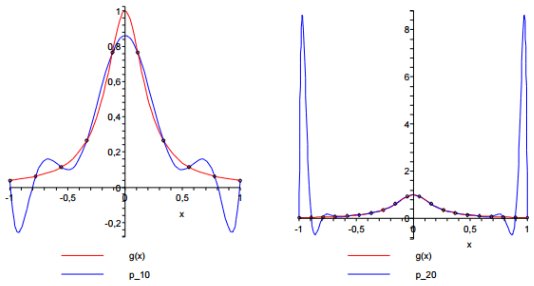
$$f(x) - P_n(x) = P_{n+1}(x) - P_n(x) = c\mathbb{T}_{n+1}(x) = \frac{1}{(n+1)!}\mathbb{T}_{n+1}(x) f^{(n+1)}(\xi)$$

Par conséquent, pour tout $x \in [a,b]$:

$$f(x) - P_n(x) = \frac{2}{(n+1)!}(\frac{b-a}{4})^n T_n(\overline{x}) f^{(n+1)}(\xi)$$

Soit

$$f(x) - P_n(x) = \frac{2}{(n+1)!}(\frac{b-a}{4})^n T_n(\frac{2x-(a+b)}{b-a}) f^{(n+1)}(\xi)$$



avec : $T_n(.) = \cos(n \arccos(.))$ Polynôme de Tchebychev sur $[0, 1]$.

Remarques :

1) Les polynômes de Tchebychev, interviennent dans l'interpolation de Lagrange pour :

- Démonter la convergence de la suite des polynômes interpolant une fonction f (limite égale à f) ;
- Choisir les points d'interpolation pour obtenir la meilleure interpolation (racines de Tchebychev) ;
- Estimer l'erreur commise.

2) Il existe d'autres méthodes d'interpolation :

- Méthode de Newton basée sur les différences divisées ;
- Méthode d'Hermite qui permet de construire un polynôme P qui coincide avec f et dont la dérivée P' coincide avec f' ;
- Interpolation par morceaux : l'intervalle $[a, b]$ est divisé à des intervalle $[a, a_1]$, $[a_1, a_2]$, $[a_2, a_3]$...$[a_{n-1}, b]$ et on cherche un polynôme d'interpolation sur chaque sous intervalle ;
- Interpolation par splines consiste à construire une polynôme d'interpolation par moreaux avec des polynômes présentant plus de régularité

3) Phénomène de Runge :

Chapitre 3

Intégration numérique

3.1 Intégration numérique

INTRODUCTION ET POSITION DU PROBLÈME

$$\text{Soit } I = \int_a^b f(x)dx \text{ où } f \text{ est continue sur } [a,b]$$

Le calcul explicite de cette intégrale n'est pas toujours possible :

$$\text{Par exemple : } \int_0^1 e^{-x^2}dx \ , \ \int_0^{\frac{\pi}{2}} \sqrt{1+\cos^2 x}dx \ , \ \int_0^1 \cos x^2 dx$$

• f n'a pas de primitive explicite,
• Le calcul analytique est long et compliqué,
• Le résultat de l'intégrale est une fonction compliquée qui fait appel à d'autres fonctions elles-même longues à évaluer.
• f n'est pas donnée par une forme explicite mais seulement par un nombre fini de couple $(x_i, y_i)_{0\leq i\leq n}$ (suite à une expérience)

INTRODUCTION ET DÉFINITION

Les méthode d'intégration numérique permettent d'obtenir une valeur approchée de l'intégrale :

$$I = \int_a^b f(x)dx$$

En utilisant des polynômes d'interpolation de f (dont le calcul de l'intégrale est beaucoup plus simple ! !).
En général, on cherche une approximation sous la forme :

$$\widetilde{I} = (b-a)\sum_{i=0}^{n} \omega_i f(x_i)$$

INTRODUCTION ET DÉFINITION

L'approximation

$$\widetilde{I} = (b-a)\sum_{i=0}^{n} \omega_i f(x_i)$$

est dite **une formule de quadrature** de I, avec :

• $x_i \in [a, b]$ sont les points d'intégration,

• ω_i sont les poids d'intégration et ils vérifiant :

$$\sum_{i=0}^{n} \omega_i = 1$$

L'approximation quadratique $\widetilde{I}$ de I dépend de la manière dont on choisit les points d'intégration (x_i) et les poids (ω_i)

FORMULE OU MÉTHODE DES TRAPÈZES

Les points d'intégration a_i sont équidistantes tels que :

$$a = a_0 < a_1 < ... < a_n < a_{n+1} = b \text{ avec } a_{i+1} - a_i = h \text{ pour} i = 0, 1, ..., n$$

et de considérer la valeur approchée :

$$I_n = \frac{h}{2} f(a) + h(f(a_1) + ... + f(a_i) + ... + f(a_n)) + \frac{h}{2} f(b)$$

que nous pouvons écrire :

$$I_n = \frac{h}{2} f(a) + \frac{h}{2} f(a_1) + \frac{h}{2} f(a_1) + ... + \frac{h}{2} f(a_i) + \frac{h}{2} f(a_{i+1}) + ... + \frac{h}{2} f(a_n) + \frac{h}{2} f(a_n)$$

$$+ \frac{h}{2} f(b)$$

Soit :

$$\begin{aligned} I_n &= \frac{h}{2}(f(a_0) + f(a_1)) \\ &+ ... \\ &+ \frac{h}{2}(f(a_i) + f(a_{i+1})) \\ &+ ... \\ &+ \frac{h}{2}(f(a_n) + f(a_{n+1})) \end{aligned}$$

Remarquons que la quantité :

$$\frac{h}{2}(f(a_i) + f(a_{i+1}))$$

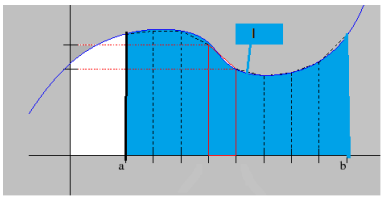
Représente l'aire du Trapèze T_i déterminé par l'axe des abscisses, $x = a_i$, $x = a_{i+1}$ et la droite qui passe par les points $(a_i, f(a_i))$ et $(a_{i+1}, f(a_{i+1}))$.

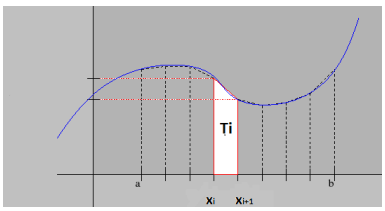
Cette méthode consiste à remplacer l'intégrale I (qui est égale à l'aire comprise entre l'axe des abscisses, $x = a$, $x = b$ et la courbe de f) par la valeur approchée donnée par la somme des aires des trapèze T_i :

$$I_n = T_0 + T_1 + ... + T_n$$

FORMULE DES TRAPÈZES

$I = \int_a^b f(x)dx$ $I_n = T_0 + T_1 + ... + T_n$





Remarques :

1) Si les points a_i ne sont pas équidistants, nous pouvons généraliser la méthode en posant :

$$h_i = a_{i+1} - a_i \text{ pour } i = 0, 1, ..., n$$

L'aire du trapèze T_i est donnée par :

$$\frac{h_i}{2}(f(a_i) + f(a_{i+1}))$$

et une valeur approchée de I est donnée par :

$$I_n = T_0 + T_1 + ... + T_n$$

2) Les sommes de Riemann définies par :

$$S_n = \sum_{j=0}^{n}(a_{j+1} - a_j)f(\theta_j) \text{ avec } \theta_j \in [a_j, a_{j+1}]$$

converge vers l'intégrale de Riemann de f ($I = \int_a^b f(x)dx$).

EXEMPLES :

1) Un seul point $x_0 \in [a, b]$: On choisit un seul point d'intégration dans $[a, b]$ et on remplace f par un polynôme d'interpolation P_0 de degré zéro tel que $P_0(x) = f(x_0)$. Donc :

$$\int_a^b f(x)dx \approx \widetilde{I} = \int_a^b P_0(x)dx = \int_a^b f(x_0)dx = (b-a)f(x_0)$$

Généralement, les cas les plus utilisés sont :

- $x_0 = a$: Méthode rectangle à gauche (d'ordre 0),
- $x_0 = b$: Méthode rectangle à droite (d'ordre 0),
- $x_0 = \frac{a+b}{2}$: Méthode du point au milieu (d'ordre 1).

EXEMPLES :

2) Interpolation linéaire : On choisit deux point d'intégration $x_0 = a$ et $x_1 = b$ et on remplacer f par un polynôme d'interpolation P_1 de degré un tel que :

$$P_1(x) = \frac{(x-a)f(b) - (x-b)f(a)}{b-a}$$

Donc :

$$\int_a^b f(x)dx \approx \widetilde{I} = \int_a^b P_1(x)dx = (b-a)\frac{f(a)+f(b)}{2}$$

Méthode du trapèze (un seul) d'ordre 1.

3) Méthode de Newton-Cotes (cas général) : On choisit les $n+1$ points équidistants définit par : $x_i = a + i\frac{b-a}{n}$ avec $i = 0, 1, ..., n$.

Soit $p_n(.)$ le polynôme d'interpolation de f sur la base des $n+1$ points distincts $(x_i)_{i=0,1,...,n}$, alors :

$$p_n(x) = \sum_{i=0}^{n} f(x_i)L_i(x)$$

avec $(L_i(.))_i$ les polynômes caractéristiques de Lagrange

$$L_i(x) = \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j}$$

Commençant par déterminer la quadrature sur l'intervalle $[-1, 1]$ en utilisant le changement de variable canonique :

$$x \in [a, b] \to \overline{x} = \frac{x - \frac{b+a}{2}}{\frac{b-a}{2}} \in [-1, 1]$$

et on subdivise l'intervalle $[-1, 1]$ en introduisant les $n+1$ points équidistants définit par :

$$\theta_i = \frac{x_i - \frac{b+a}{2}}{\frac{b-a}{2}} = -1 + \frac{2i}{n} \text{ avec } i = 0, 1, ..., n$$

Le polynôme d'interpolation d'une fonction $f \in C([-1, 1])$ (sur la base des θ_i) est donné par :

$$p_n(x) = \sum_{i=0}^{n} f(\theta_i)L_i(x)$$

avec $(L_i(.))_i$ les polynômes caractéristiques de Lagrange

$$L_i(x) = \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j}$$

Donc :

$$\begin{aligned}
\int_{-1}^{1} f(x)dx &\approx \int_{-1}^{1} p_n(x)dx = \int_{-1}^{1} \sum_{i=0}^{n} f(\theta_i)L_i(x)dx \\
&\approx \sum_{i=0}^{n} f(\theta_i) \int_{-1}^{1} L_i(x)dx = 2\sum_{i=0}^{n} f(\theta_i) \underbrace{\frac{1}{2}\int_{-1}^{1} L_i(x)dx}_{\omega_i} \\
&\approx 2\sum_{i=0}^{n} f(\theta_i)\omega_i
\end{aligned}$$

Les points θ_i sont distribués d'une manière symétrique autour de 0, On a : $\theta n - i = -\theta_i$, $L_{n-i}(x) = L_i(-x)$, $\omega_{n-i} = \omega_i$
Sur $[a, b]$: calculons $\int_a^b f(x)dx$ en utilisant le changement de variable :

$$\overline{x} = \frac{x - \frac{b+a}{2}}{\frac{b-a}{2}} \Rightarrow x = \frac{b-a}{2}\overline{x} + \frac{b+a}{2} = a + (\overline{x}+1)\frac{b-a}{2}$$

avec : $x = a \Rightarrow \overline{x} = -1$, $x = b \Rightarrow \overline{x} = 1$, $dx = \frac{b-a}{2}d\overline{x}$

$$\int_a^b f(x)dx = \frac{b-a}{2}\int_{-1}^{1} f(a + (\overline{x}+1)\frac{b-a}{2})d\overline{x}$$

$$\int_a^b f(x)dx \approx \frac{b-a}{2} \times 2\sum_{i=0}^{n} f(a + (\theta_i + 1)\frac{b-a}{2})\omega_i$$

On en déduit :

$$\int_a^b f(x)dx \approx (b-a)\sum_{i=0}^{n} f(a + (\theta_i + 1)\frac{b-a}{2})\omega_i = (b-a)\sum_{i=0}^{n} f(x_i)\omega_i$$

Avec :

$$\omega_i = \frac{1}{2}\int_{-1}^{1} L_i(x)dx$$

Les poids de la quadrature sur $[-1, 1]$.

ORDRE D'UNE MÉTHODE D'INTÉGRATION :

L'ordre d'une méthode d'intégration est le plus grand degré des polynômes intégrés exactement ($I = \widetilde{I}$). Nous avons :

Méthode	Nombre de point	Ordre
Rectangle	1	0
Milieu	1	1
Trapèze (un)	2	1
Simpson (Newton-Cotes $n = 2$)	3	3

MÉTHODES COMPOSITES DE CALCUL DE $\int_a^b f(x)dx$

On pose $h = \frac{b-a}{n}$ et $a_i = a + ih$ avec $i = 0, 1, ..., n$

Méthode composite des rectangles à gauche :

$$\int_a^b f(x)dx \simeq \sum_{i=0}^{n-1} hf(a_i) = \sum_{i=0}^{n-1} hf(a + ih)$$

Méthode composite des rectangles à droite : $\int_a^b f(x)dx$

$$\simeq \sum_{i=1}^{n} hf(a_i) = \sum_{i=1}^{n} hf(a + ih) = \sum_{i=0}^{n-1} hf(a_{i+1}) = \sum_{i=0}^{n-1} hf(a + (i+1)h)$$

MÉTHODES COMPOSITES DE CALCUL DE $\int_a^b f(x)dx$

Méthode composite des rectangles au milieu :

$$\int_a^b f(x)dx \simeq \sum_{i=0}^{n-1} hf(\frac{(a_i + a_{i+1})}{2}) = \sum_{i=0}^{n-1} hf(a + (i + \frac{1}{2})h)$$

Méthode composite des trapèzes : $\int_a^b f(x)dx$

$$\simeq \frac{h}{2}(f(a)+f(b))+\sum_{i=1}^{n-1} hf(a_i) = \frac{h}{2}(f(a)+f(b))+\sum_{i=1}^{n-1} hf(a+ih)$$

Méthode de Simpson (Newton-cotes avec $n=2$)

$$\int_a^b f(x)dx = \frac{b-a}{6}\left[f(a)+4f(\frac{a+b}{2})+f(b)\right]$$

Démonstration : Application de la méthode Newton-cotes avec $n=2$ dans $[-1,1]$ puis $[a,b]$

Méthode composite de Simpson :

Application de la méthode de Simpson dans chaque intervalle $[a_i, a_{i+2}]$ (n est pair) :

$$\int_a^b f(x)dx = \frac{h}{3}\left[f(a)+2\sum_{j=1}^{j=\frac{n}{2}-1} f(a_{2j})+4\sum_{j=1}^{j=\frac{n}{2}} f(a_{2j-1})+f(b)\right]$$

MÉTHODES COMPOSITES DE CALCUL DE $\int_a^b f(x)dx$

Méthodes basées sur l'interpolation : On détermine P_n un polynôme d'interpolation et on a :

$$\int_a^b f(x)dx \simeq \int_a^b P_n(x)dx$$

Interpolation de Lagrange $\rightarrow$ Méthode de Newton-cotes. Dans ce cas, il n'est pas nécessaire de calculer P_n mais seulement les L_i.

Interpolation de Newton utilisant les différences divisées : Cette Méthode utilise une base différente de l'espace $\mathbb{P}_n[X]$, à savoir : MÉTHODES COMPOSITES DE CALCUL DE $\int_a^b f(x)dx$

$$\begin{cases} Q_0(x)=1 \\ Q_k(x)=(x-x_0)(x-x_1)...(x-x_{k-1})=\prod_{j=0}^{j=k-1}(x-x_j), \quad \text{k=1,2,...,n} \end{cases}$$

Dans cette base, le coefficient de $Q_k(x)$ est donné par la différence divisée $f[x_0,x_1,...,x_k]$ définie par récurrence :

$$\begin{cases} f[x_0]=f(x_0) \\ f[x_0,x_1,...,x_k]=\frac{f[x_1,...,x_k]-f[x_0,...,x_{k-1}]}{x_k-x_0} \quad \text{k=1,2,...,n} \end{cases}$$

MÉTHODES COMPOSITES DE CALCUL DE $\int_a^b f(x)dx$

Une méthode pratique :

x	$f[x]=f(x)$			
x_0	$f[x_0]$			
x_1	$f[x_1]$	$f[x_0,x_1]$		
x_2	$f[x_2]$	$f[x_1,x_2]$	$f[x_0,x_1,x_2]$	
x_3	$f[x_3]$	$f[x_2,x_3]$	$f[x_1,x_2,x_3]$	$f[x_0,x_2,x_2,x_3]$

$$P_3(x)=f[x_0]+f[x_0,x_1](x-x_0)+f[x_0,x_1,x_2](x-x_0)(x-x_1)$$

x	0.1	0.2	0.3	0.4	0.5
$f(x)$	1.40	1.56	1.76	2.00	2.28

x	$f[x]=f(x)$				
0.1	1.40				
0.2	1.56	$\frac{1.56-1.40}{0.2-0.1}=1.6$			
0.3	1.76	$\frac{1.76-1.56}{0.3-0.2}=2.0$	$\frac{2.0-1.6}{0.3-0.1}=2.0$		
0.4	2.00	$\frac{2.0-1.76}{0.4-0.3}=2.4$	$\frac{2.4-2.}{0.4-0.2}=2.0$	0.	
0.5	2.28	$\frac{2.28-2.0}{0.5-0.4}=2.8$	$\frac{2.8-2.4}{0.5-0.3}=2.0$	0.	0.

$$+f[x_0,x_1,x_2,x_3](x-x_0)(x-x_1)(x-x_2)$$

$$P_4(x)=1.4+1.6(x-0.1)+2(x-0.1)(x-0.2)+0(x-0.1)(x-0.2)(x-0.3)$$
$$+0(x-0.1)(x-0.2)(x-0.3)(x-0.4)$$

ESTIMATION DE L'ERREUR

Méthode	Simple	Combinée
Rectangle au milieu	$E=\frac{(\frac{b-a}{2})^3}{3}\lvert f''(\eta)\rvert$	$E_n=\frac{b-a}{24}(\frac{b-a}{n})^2\lvert f''(\eta)\rvert$
Trapèzes	$E=\frac{(b-a)^3}{12}\lvert f''(\eta)\rvert$	$E_n=\frac{(b-a)^3}{12}(\frac{b-a}{n})^2\lvert f''(\eta)\rvert$
Simpson	$E=\frac{(\frac{b-a}{2})^5}{90}\lvert f^{(4)}(\eta)\rvert$	$E_n=\frac{b-a}{180}(\frac{b-a}{4n})^4\lvert f^{(4)}(\eta)\rvert$

Les méthodes précédentes fixent d'abord les points x_i et cherchent les poids ω_i. Pour améliorer la méthode, on peut chercher à optimiser le choix des x_i pour obtenir la meilleure approximation (de maximiser l'ordre de la méthode, $n+1$ au maximum).
On se place sur $[-1,1]$ et on cherche les points x_i et les poids ω_i pour minimiser la différence :

$$\int_{-1}^{1} f(x)dx-\sum_{i=0}^{n} f(x_i)\omega_i$$

MÉTHODE DE GAUSS-LEGENDRE :

Théorème :

Il existe un choix et un seul des points x_i et des poids ω_i de sorte que la méthode soit d'ordre $p=2n+1$. Les points x_i sont les zéros du polynôme de Legendre $\mathbb{L}_{n+1}$. Les poids ω_i sont donnés par plusieurs formules.
D'autre part, l'erreur est donnée par :

$$E(f)=c\frac{f^{(2n+2)}(\xi)}{(2n+2)!} \text{ où } \xi\in[-1,1] \text{ à condition : } f \text{ suffisamment régulière}$$

MÉTHODE DE GAUSS-LEGENDRE :

$\mathbb{L}_{n+1}$ définit par la relation de recurrence :

$$(n+1)\mathbb{L}_{n+1}(x)=(2n+1)x\mathbb{L}_n(x)-n\mathbb{L}_{n-1}(x), \text{ avec } \mathbb{L}_0=1 \text{ et } \mathbb{L}_1=x$$

Pour déterminer les racines, $(x_i)_{i=0,\dots,n}$, de $\mathbb{L}_{n+1}(x)$ nous pouvons utiliser la formule équivalente suivante :

$$x\mathbb{L}_n(x)=\frac{n+1}{2n+1}\mathbb{L}_{n+1}(x)+\frac{n}{2n+1}\mathbb{L}_{n-1}(x)$$

et x est une valeur propre d'une matrice tridiagonale (à déterminer).
En pratique, nous pouvons utiliser une méthode numérique pour déterminer des valeurs approchées des racines $(x_i)_{i=0,...,n}$.

$$\mathbb{L}_0 = 1, \mathbb{L}_1 = x \text{ et } \mathbb{L}_2 = \frac{3}{2}x^2 - \frac{1}{2} = \frac{1}{2}(3x^2 - 1)$$

Un seul point d'intégration : $x_0 = 0$ et $\omega_0 = 2$
Deux points $\mp\sqrt{\frac{1}{3}}$ et les poids : $\omega_i = 1, i = 0, 1$
Trois points d'intégration : $\mathbb{L}_3 = \frac{1}{2}(5x^3 - 3x) \Rightarrow x_i = \mp\sqrt{\frac{3}{5}}, 0$ et $\omega_i = \frac{5}{9}, \frac{8}{9}, \frac{5}{9}$

MÉTHODE DE GAUSS-LEGENDRE :

1) Les points d'intégration, $(x_i)_{i=0,...,n-1}$, racines du polynôme de Legendre $\mathbb{L}_n$,
2) Les poids $(\omega_i)_{i=0,...,n-1}$ sont aussi donnés par :

$$\omega_i = \frac{-2}{(n+1)\mathbb{L}'_n(x_i)\mathbb{L}_{n+1}(x_i)} = \frac{2}{(1-x_i^2)(\mathbb{L}'_n(x_i))^2} \text{ pour } i = 0, ..., n$$

3) $\mathbb{L}_n$ est définit aussi comme solution de l'équation de Legendre :

$$\frac{d}{dx}[(1-x^2)\frac{dy}{dx}] + n(n+1)y = 0$$

C'est à dire que $\mathbb{L}_n$ vérifie : $\frac{d}{dx}[(1-x^2)\frac{d\mathbb{L}_n}{dx}] + n(n+1)\mathbb{L}_n = 0$.

Chapitre 4

Résolution numérique des équations différentielles

4.1 Dérivation numérique et résolution des Équations différentielles

INTRODUCTION ET POSITION DU PROBLÈME

La dérivée d'une fonction f en un point $x \in \mathbb{R}$ est définie par :

$$\lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

Le calcul du dérivée n'est pas toujours possible :

- Le calcul analytique est long et compliqué,
- f n'est pas donnée par une forme explicite mais seulement par un nombre fini de couple $(x_i, y_i)_{0 \leq i \leq n}$ (suite à une expérience).

Une équation différentielle d'inconnu y (une fonction) s'écrit :

$$y'(t) = f(y(t), t)$$

Lorsque on ne peut pas appliquer les méthodes usuelles (forme explicite compliquée), on cherche à appliquer des méthodes numériques.

En particulier, nous considérons le **problème de Cauchy** :

$$y'(t) = f(y(t), t) \text{ si } t > 0 \text{ avec la condition initiale } y(0) = y_0$$

$f : \mathbb{R}^+ \times \mathbb{R} \to \mathbb{R}$ une fonction continue et $y : \mathbb{R}^+ \to \mathbb{R}$.

Théorème (Cauchy-Lipschitz) Si $f : \mathbb{R}^+ \times \mathbb{R} \to \mathbb{R}$ est une fonction continue et Lipschitzienne par rapport à la deuxième variable : Il existe $L > 0$ tel que :

$$|f(t, x_1) - f(t, x_2)| \leq L|x_1 - x_2| \text{ pour tout } t > 0 \text{ et tout } x_1, x_2 \in \mathbb{R}$$

Alors, le problème de Cauchy admet une solution globale (définie pour tout $t > 0$) et cette solution est unique.

Soit y une fonction de classe $C^1([a,b])$ et $a = t_0 < t_1 < ... < t_n = b$, une partition de $n+1$ points équidistants dans $[a,b]$ et $h = \frac{b-a}{n}$ la distance entre deux points consécutifs.
La dérivée est donnée par l'une des trois formules :

$$y'(t_i) = \lim_{h\to 0^+} \frac{y(t_i+h) - y(t_i)}{h}$$

$$y'(t_i) = \lim_{h\to 0^+} \frac{y(t_i) - y(t_i-h)}{h}$$

$$y'(t_i) = \lim_{h\to 0^+} \frac{y(t_i+h) - y(t_i-h)}{2h}$$

DÉRIVÉE NUMÉRIQUE : Une approximation numérique $(Dy)_i$ de $y'(t_i)$ peut être définie :
- Différence finie progressive (ou décentrée à droite) :

$$(Dy)_i^P = \frac{y(t_i+h) - y(t_i)}{h} == \frac{y(t_{i+1}) - y(t_i)}{h} \text{ avec } i = 0, 1, ..., n-1$$

- Différence finie rétrograde (ou décentrée à gauche) :

$$(Dy)_i^R = \frac{y(t_i) - y(t_i-h)}{h} = \frac{y(t_i) - y(t_{i-1})}{h} \text{ avec } i = 1, ..., n$$

- Différence finie centrée :

$$(Dy)_i^C = \frac{y(t_i+h) - y(t_i-h)}{2h} = \frac{y(t_{i+1}) - y(t_{i-1})}{2h} \text{ avec } i = 1, ..., n-1$$

Si y une fonction de classe $C^2(\mathbb{R})$, il existe θ entre t_i et t avec (Formule de Taylor) :

$$y(t) = y(t_i) + y'(t_i)(t - ti) + \frac{y''(\theta)}{2}(t - t_i)^2$$

- Pour $t = t_{i+1}$ on obtient :

$$y(t_{i+1}) = y(t_i) + y'(t_i)(t_{i+1} - ti) + \frac{y''(\theta)}{2}(t_{i+1} - t_i)^2$$

Soit :

$$(Dy)_i^P = y'(t_i) + \frac{h}{2}y''(\theta)$$

D'ou :

$$|y'(t_i) - (Dy)_i^P| \leq Kh \text{ avec } K = \frac{1}{2}\max_{t\in[t_i,t_{i+1}]} y''(t)$$

- Pour $t = t_{i-1}$ on obtient :

$$y(t_{i-1}) = y(t_i) + y'(t_i)(t_{i-1} - ti) + \frac{y''(\theta)}{2}(t_{i-1} - t_i)^2$$

Soit : $(Dy)_i^R = y'(t_i) - \frac{h}{2}y''(\theta)$. D'ou :

$$|y'(t_i) - (Dy)_i^R| \leq Kh \text{ avec } K = \frac{1}{2}\max_{t\in[t_{i-1},t_i]} y''(t)$$

Si y une fonction de classe $C^3(\mathbb{R})$, il existe θ entre t_i et t avec (Formule de Taylor) :

$$y(t) = y(t_i) + y'(t_i)(t - ti) + \frac{y''(t_i)}{2}(t - t_i)^2 \frac{y'''(\theta)}{6}(t - t_i)^3$$

On obtient pour $t = t_{i+1}$et$t = t_{i-1}$:

$$y(t_{i+1}) = y(t_i) + y'(t_i)(t_{i+1} - ti) + \frac{y''(t_i)}{2}(t_{i+1} - t_i)^2 + \frac{y'''(\theta_1)}{6}(t_{i+1} - t_i)^3$$

$$y(t_{i-1}) = y(t_i) + y'(t_i)(t_{i-1} - ti) + \frac{y''(t_i)}{2}(t_{i-1} - t_i)^2 + \frac{y'''(\theta_2)}{6}(t_{i-1} - t_i)^3$$

Soit :

$$y(t_{i+1}) - y(t_i) = y'(t_i)(h) + \frac{y''(t_i)}{2}(h)^2 + \frac{y'''(\theta_1)}{6}(h)^3$$

$$y(t_{i-1}) - y(t_i) = y'(t_i)(-h) + \frac{y''(t_i)}{2}(-h)^2 + \frac{y'''(\theta_2)}{6}(-h)^3$$

On obtient donc :

$$(Dy)_i^C = y'(t_i) + \frac{y'''(\theta_1) + y'''(\theta_2)}{12}h^2$$

et

$$|y'(t_i) - (Dy)_i^C| \leq Kh^2 \text{ avec } K = \frac{1}{6}\max_{t\in[t_{i-1},t_{i+1}]} y'''(t)$$

L'erreur lié à ce calcul :

$$e_i^P = |y'(t_i) - (Dy)_i^P|$$

$$e_i^R = |y'(t_i) - (Dy)_i^R|$$

$$e_i^C = |y'(t_i) - (Dy)_i^C|$$

Cet erreur est appelé : **Erreur de troncature au point t_i en utilisant la différence progressive** (resp. Rétrograde) (resp. centrée).

Remarque : Les méthodes progressive et rétrograde sont d'ordre 1 et la méthode centrée est d'ordre 2.

MÉTHODES D'EULER :

Soit $a = t_0 < t_1 < ... < t_i < t_{n+1} < .... < t_n = b$ une partition de points équidistants de $[a, b]$ avec $h = t_{i+1} - t_i$ le pas de la partition.

On note y_i une approximation de $y(t_i)$, le problème de Cauchy s'écrit pour $t = t_i$ comme suit :

$$y'(t_i) = f(t_i, y(t_i) = f(t_i, y_i)$$

Par la suite, on utilise l'une des formules de dérivée numérique pour $y'(t_i)$ et on obtient les approximations suivantes :

Schéma d'Euler Progressif :

$$\begin{cases} \frac{y_{i+1}-y_i}{h} = f(t_i, y_i) \quad \text{pour } i = 0, 1, 2, ... \\ y_0 \text{ donnée,} \end{cases}$$

Schéma d'Euler retrograde :

$$\begin{cases} \frac{y_{i+1}-y_i}{h} = f(t_{i+1}, y_{i+1}) \quad \text{pour } i = 0, 1, 2, ... \\ y_0 \text{ donnée,} \end{cases}$$

ÉTUDE GÉNÉRALE DES MÉTHODES À UN PAS : Les méthodes à un pas sont les méthodes de résolution numériques qui peuvent s'écrire sous la forme :

$$y_{i+1} = y_i + h_i\Phi(t_i, y_i, h_i) \text{ pour } i = 0, 1, 2....$$

Où Φ est une fonction supposée continue.

CONSISTANCE, STABILITÉ ET CONVERGENCE :

Définition : L'erreur de consistance e_i relative à une solution exacte y est donnée par :

$$e_i = y(t_{i+1}) - y_{i+1} \text{ pour } i = 0, 1, ...$$

$$e_i = y(t_{i+1}) - y_i + h_i\Phi(t_i, y_i, h_i) = y(t_{i+1}) - y(t_i) + h_i\Phi(t_i, y_i, h_i) \text{ pour } i = 0, 1, ...$$

En supposant $y_i = y(t_i)$ (c'est à dire la valeur exacte au rang i).
La méthode est dite consistante lorsque la somme des erreurs relatives à y, soit $\sum_i |e_i|$, tend vers 0 lorsque le pas

$$h_{max} = \max_{i=0,1,...} h_i \to 0$$

Dans le calcul récurent (ou itératif) des y_i d'autre erreurs viendront s'ajouter à l'erreur de l'approximation, en particulier les erreurs d'arrondis numériques ε_i.
La notion de stabilité d'une méthode joue un role important dans le contrôle de la propagation des erreurs pour obtenir des résultats significatifs.

Définition : Une méthode d'un pas est dite stable s'il existe une constante $S \geq 0$ telle que pour les suites y_i et $\widetilde{y}_i$ définies respectivement par :

$$y_{i+1} = y_i + h_i\Phi(t_i, y_i, h_i) \text{ pour } i = 0, 1, 2....$$

$$\widetilde{y}_{i+1} = \widetilde{y}_i + h_i\Phi(t_i, \widetilde{y}_i, h_i) + \varepsilon_i \text{ pour } i = 0, 1, 2....$$

On ait :

$$\max_{i=0,1,..} |\widetilde{y}_i - y_i| \leq S(|\widetilde{y}_0 - y_0| + \sum_{i=0,1,..} \varepsilon_n)$$

La notion de convergence pour les méthodes d'un pas est définie comme suit :

Définition : Une méthode d'un pas est dite convergeante, si pour toute solution exacte y, la suite y_i vérifie :

$$\max_{i=0,1,..} |y_i - y(t_i)| \to 0 \text{ lorsque } h_{max} \to 0 \text{ et } y_0 \to y(0)$$

Méthode stable et consistante $\Rightarrow$ méthode convergente.

Condition nécessaire et suffisante de consistance : Une méthode à un pas définie par une fonction Φ est consistante Si et seulement si :

$$\Phi(t, y, 0) = f(t, y) \text{ pour } t \in \mathbb{R}^+ \text{ et } y \in \mathbb{R}$$

Condition suffisante de stabilité : Pour q'une méthode à un pas définie par une fonction Φ soit stable, il suffit que Φ soit Lipschitzienne, de constante λ, en y :

$$|\Phi(t, y_1, h) - \Phi(t, y_2, h)| \leq \lambda|y_1 - y_2| \text{ pour } t \in [0, T], y_1, y_2 \in \mathbb{R} \text{ et } h \in \mathbb{R}$$

Dans ce cas, la constante de stabilité peut être choisie comme suit : $S = e^{\lambda T}$.

Ordre d'une méthode : Une méthode à un pas est dite d'ordre supérieur ou égal à p si pour toute solution exacte y de l'équation différentielle

$$y' = f(t, y) \text{ où } f \in C^p$$

Il existe une constante $C \geq 0$ telle que l'erreur de consistance relative à y vérifie :

$$|e_i| \leq Ch_i^{p+1} \text{ pour } i = 0, 1, ...$$

MÉTHODE DE RUNGE-KUTTA D'ORDRE 2 :
L'intégration de l'équation $y'(t) = f(t, y(t))$ entre t_i et t_{i+1}, on obtient :

$$y(t_{i+1}) - y(t_i) = \int_{t_i}^{t_{i+1}} f(t, y(t)dt$$

En calculant l'intégrale du second membre par la méthode du trapèze, on obtient le schéma implicite suivant (appelée : Méthode de **Crank-Nicolson** ou du trapèze) :

$$y_{i+1} - y_i = \frac{h}{2}[f(t_i, y_i) - f(t_{i+1}, y_{i+1})] \text{ pour } i = 0, 1, ...$$

Cette méthode implicite peut être adaptée à un schéma explicite (appelée : Méthode de **Heun**)

$$y_{i+1} - y_i = \frac{h}{2}[f(t_i, y_i) - f(t_{i+1}, y_i + hf(t_i, y_i))] \text{ pour } i = 0, 1, ...$$

Remarque : Les deux méthodes sont d'ordre 2 par rapport à h.

Si l'intégrale est approchée par la méthode du point milieu, on obtient :

$$y_{i+1} - y_i = hf(t_{i+\frac{1}{2}}, y_{i+\frac{1}{2}})$$

et si on utilise l'approximation :

$$y_{i+\frac{1}{2}} = y_i + \frac{1}{2}f(t_i, y_i)$$

On trouve la méthode d'Euler modifiée :

$$y_{i+1} - y_i = hf(t_{i+\frac{1}{2}}, y_i + \frac{1}{2}f(t_i, y_i))$$

Les méthodes de **Heun** et d'**Euler modifiée** sont des cas particuliers dans la famille des méthodes de Runge-Kutta d'ordre2. il existe d'autres méthodes plus avancées, comme par exemple, la méthode d'ordre 4 suivante (Obtenue en calculant l'intégrale par la méthode de Simpson) :

$$y_{i+1} = y_i + \frac{h}{6}(K_1 + 2K_2 + 2K_3 + K_4)$$

avec :

$$K_1 = f(t_i, y_i),\ K_2 = f(t_i + \frac{h}{2}, y_i + \frac{h}{2}K_1)$$

$$K_3 = f(t_i + \frac{h}{2}, y_i + \frac{h}{2}K_2),\ K_4 = f(t_i + \frac{h}{2}, y_i + \frac{h}{2}K_3)$$