Modélisation et Simulation pour la Physique

HMPH104 - 2015

A. PALACIOS

Table des matières

Chapitre 1

Introduction

Sommaire

1.1 Présentation

La physique est la *science qui a pour objet l'étude de la matière et de ses propriétés fondamentales*[1]. Pour faire de la physique, on a recours à différentes méthodes : l'expérimentation d'abord, mais également la représentation des systèmes physiques et de leurs interactions par le biais de modèles et de lois.

Dans le cadre de l'expérimentation, les outils mis en œuvre sont des instruments de mesure (oeil, oreille, osciloscope, spectromètre, anémomètre, ...).

Dans le cadre de la modélisation l'outil principal est la mathématique qui permet de donner une représentation des systèmes physiques par des équations sous un certain nombre d'hypothèses. Depuis l'apparition des ordinateurs, l'outil mathématique est utilisé dans la limite des réalisations possibles par la machine et est mis en oeuvre par le biais de ce que l'on appelle les méthodes numériques.

Les méthodes numériques sont un ensemble de techniques de calcul et de méthodes mathématiques permettant **d'estimer** la solution d'un problème physique

1. Dictionnaire de l'Académie Française

donné. Par essence, une méthode numérique est donc indépendante du support utilisé (ordinateur, langage de programmation). Elle peut toujours être décrite par un algorithme qui sera par la suite traduit en instructions dans le langage de programmation choisi pour être mise en oeuvre sur une machine donnée.

1.2 Performance des méthodes numériques

Comme la mise en œuvre d'une méthode numérique pour résoudre un problème physique implique l'utilisation d'un modèle mathématique, **la solution numérique d'un problème est un modèle : ce n'est pas une solution exacte.**
La performance d'une méthode numérique se mesure en termes de (1) temps de calcul Δt (2) précision du résultat $\Delta\varepsilon$. Une méthode numérique adaptée à un problème physique sera celle qui minimise le produit $\Delta t \Delta\varepsilon$.
Ce produit est en réalité borné : plus on calcule vite, moins on est précis et plus on est précis, plus le calcul est long.

$$\Delta t \Delta\varepsilon \geq h_{\text{num}}$$

Les performances d'une méthode numérique ne dépendent pas seulement du schéma numérique choisi. Alors que le temps de calcul et la précision du résultat dépendent tous deux de la machine, Δt est également fonction du langage de programmation et du type d'opérations effectuées alors que la précision dépend de la représentation des nombres utilisée pour exécuter le calcul.

1.3 Représentation machine des nombres

Les nombres ne sont pas stockés avec une précision infinie dans un ordinateur, mais plutôt sous la forme d'une approximation qui peut être représentée par un nombre fini d'éléments binaires (*bits*[2]). Le choix de la représentation est laissé au programmeur. Le type de données diffère en termes de taille d'encodage (8 bits, 32 bits, 64 bits) et de représentation machine : virgule fixe (entier) ou virgule flottante (réel).

Entiers les nombres entiers sont représentés de façon *exacte* sur un domaine limité et sont généralement stockés su 32 bits (31 pour le nombre et 1 pour le signe). Comme l'arithmétique numérique se fait en mode binaire (base 2), le domaine accessible aux entiers sans perte d'information est [-2^{31}+1 ; 2^{31}-1].
À noter que le résultat d'une opération sur des entiers donne un entier (c'est en particulier vrai pour la division simple dans le langage Python.)

Réels les nombres réels ont une représentation *tronquée* sur un domaine limité. L'ordinateur utilise une représentation en virgule flottante, la plupart du temps en base 2 qui s'écrit comme suit :

$$x = s \times M \times B^e$$

2. Un *byte* est un groupe de 8 *bits*

où s est le signe, M la mantisse, B la base et e l'exposant. Le stockage se fait sur 32 bits ou sur 64 bits suivant que la représentation est en simple ou double précision.
Tout comme le nombre lui-même, la précision du résultat d'une opération entre deux nombres en virgule flottante est également limitée.

Précision	Valeur min	Valeur max
Simple	$1.175\ 10^{-38}$	$3.403\ 10^{38}$
Double	$2.2251\ 10^{-308}$	$1.7977\ 10^{308}$

1.4 Erreur numérique

Lorsque l'on analyse les résultats d'un calcul numérique il est important d'avoir conscience que le résultat n'est pas une solution mathématique exacte. La précision d'une solution numérique peut être affectée de plusieurs manières plus ou moins évidentes. Comprendre ces difficultés peut aider le programmeur dans ses choix de méthodes numériques.

Définition
Soit $\hat{p}$ une approximation de p. L'***erreur absolue*** est définie par $E_p = |p - \hat{p}|$. L'***erreur relative*** vaut $R_p = |p - \hat{p}|/|p|$, si p est différent de zéro.

L'erreur relative R_p est un meilleur indicateur de la précision de l'approximation que l'erreur absolue E_p lorsque $|p|$ est très différent de 1. Pour qualifier la précision du résultat d'une opération impliquant des nombres réels (en virgule flottante) on utilisera de préférence l'erreur relative, qui est directement reliée à la mantisse.

Définition
On dira de $\hat{p}$ que c'est une ***approximation de p à d chiffres significatifs*** si d est le plus grand entier positif ou nul pour lequel

$$\frac{|p - \hat{p}|}{|p|} < \frac{10^{1-d}}{2}$$

.

1.4.1 Erreur de troncature

Une erreur de troncature ou erreur de schéma apparaît lorsqu'une relation mathématique exacte est remplacée par une relation approchée plus facile à manipuler numériquement. Par exemple, une erreur de schéma apparaît lorsqu'on remplace un développement infini en série de Taylor de l'exponentielle :

$$e^{x^2} = \sum_{n=0}^{\infty} \frac{x^{2n}}{n!}$$

par les cinq premiers termes e cette série :

$$e^{x^2} = \sum_{n=0}^{4} \frac{x^{2n}}{n!}$$

1.4.2 Erreur d'arrondi

Une erreur d'arrondi est liée à la représentation inexacte des nombres par la machine. On définit la ***précision machine*** ε_{m} comme étant le plus petit nombre tel que $1 + \varepsilon_{\mathrm{m}} > 1$.
Pour un PC on a :

Précision	Réels positifs	Réels négatifs
Simple	$1.192\ 10^{-7}$	$-5.496\ 10^{-8}$
Double	$2.220\ 10^{-16}$	$-1.110\ 10^{-16}$

1.4.3 Perte d'information

La perte d'information est liée à l'erreur de représentation machine lors de la répétition d'opérations faisant intervenir des termes de même amplitude. Par exemple, si on considère $p = 3.1415926536$ et $q = 3.1415957341$, la différence $p - q = -0.0000030805$ ne contient que 5 chiffres de précision alors que p et q sont donnés avec 11 chiffres de précision. Cette perte de précision dans le résultat final d'une opération peut se propager et entraîner des erreurs importantes là où on ne les attend pas forcément.

1.5 Stabilité d'une méthode numérique

La stabilité d'une méthode numérique se définit par sa façon à propager les erreurs. Si une quantité x est connue avec une précision relative $\varepsilon(x) << 1$, alors son image par la fonction f sera connue arec une erreur relative

$$\varepsilon(f) = \varepsilon(x)\ \eta$$

où η est appelé ***nombre de conditionnement***. La valeur du nombre de conditionnement définit la stabilité.

Définition
Soit une méthode donnant $z = z(x)$, et deux couples de valeurs (x, z) et (x', z'). Alors on a

$$\eta = \frac{\frac{z'-z}{z}}{\frac{x'-x}{x}} \equiv \eta(x) \approx x\frac{z'(x)}{z(x)}$$

La méthode z sera instable si $eta > 1$, stable si $\eta \leq 1$ et insensible si $\eta << 1$

1.5.1 Propagation des erreurs

Cas de l'addition

Soit $\hat{p}$ une approximation de p telle que $p = \hat{p} + \varepsilon_p$, et $\hat{q}$ une approximation de q telle que $q = \hat{q} + \varepsilon_q$. Dans ce cas

$$p + q = \hat{p} + \hat{q} + \varepsilon_p + \varepsilon_q$$

L'erreur de la somme est la somme des erreurs des termes.

Cas de la multiplication

Le produit de p et q donne

$$pq = (\hat{p} + \varepsilon_p)(\hat{q} + \varepsilon_q) = \hat{p}\hat{q} + \hat{p}\varepsilon_q + \hat{q}\varepsilon_p + \varepsilon_p\varepsilon_q$$

On peut utiliser cette équation pour écrire l'erreur relative du produit R_{pq} :

$$R_p q = \frac{pq - \hat{p}\hat{q}}{pq} = \frac{\hat{p}\varepsilon_q}{pq} + \frac{\hat{q}\varepsilon_p}{pq} + \frac{\varepsilon_p\varepsilon_q}{pq}$$

Si on considère que $\hat{p}$ et $\hat{q}$ sont de bonnes approximations de p et de q, alors on a $\hat{p}/p \approx \hat{q}/q \approx 1$ et on obtient

$$R_{pq} = \frac{pq - \hat{p}\hat{q}}{pq} \approx \frac{\varepsilon_q}{q} + \frac{\varepsilon_p}{p} = R_q + R_p$$

L'erreur relative du produit peut être approximée par la somme des erreurs relatives sur p et q.

On choisira autant que possible des méthodes qui sont stables, c'est-à-dire qui n'amplifient pas les erreurs initiales sur les composants qui entrent dans la série d'opérations effectuées par la mise en œuvre de la méthode.

Définition

Soit ε une erreur initiale et $\varepsilon(n)$ la croissance de l'erreur après n itérations. Si $|\varepsilon(n)| \approx n\varepsilon$, la croissance de l'erreur est dite ***linéaire***. Si $|\varepsilon(n)| \approx K^n\varepsilon$, on parlera de croissance d'erreur ***exponentielle***. Si $K > 1$, l'erreur exponentielle croît sans limites lorsque $n \to \infty$, et si $0 < K < 1$, l'erreur exponentielle tend vers 0 lorsque $n \to \infty$.

1.6 Rappels d'algorithmique

Pour mettre en œuvre des méthodes numériques il est nécessaire d'utiliser une méthodologie qui va permettre d'écrire des programmes lisibles, fiables, maintenables, réutilisables, portables, efficaces, corrects et qui obéissent à des contraintes économiques (en terme de temps de calcul, nombre de processeurs, mémoire et espace disque nécessaires).

Cette méthodologie passe par :

- L'analyse du problème via une reformulation explicite
- La conception préliminaire, où on définit le type de structuration de programme que l'on va utiliser
- La conception détaillée, qui correspond à l'écriture de l'**algorithme**
- Le codage dans le langage choisi.

Pour spécifier un problème, on applique une ***analyse descendante*** qui consiste à structurer le problème et à le décomposer en une série d'opérations hiérarchisées dont la résolution est élémentaire. Ces sous-blocs sont ensuite combinés/enchaînés de façon à obtenir la résolution du problème initial.
Pour la conception préliminaire, il va s'agir de choisir ce que l'on appelle un paradigme de programmation, qui est en fait une type de programmation. Dans ce cours, vous aborderez la **em programmation orientée objet** et la ***programmation impérative structurée***.
Dans la programmation impérative, les opérations sont décrites en termes d'états du programme et de séquences d'instructions exécutées par l'ordinateur pour modifier l'état du programme. Lorsque la programmation est structurée, cela signifie que les programmes sont hiérarchiquement subdivisés en procédures et fonctions. Dans la programmation impérative, l'instruction de base est l'affectation (ou assignation) d'une valeur à une variable. Les instructions d'affectation peuvent être organisées en mode séquentiel, conditionnel ou itératif.

Définition
Un algorithme est une suite d'instructions qui une fois exécutée correctement, conduit à un résultat donné.

Dans le cadre de l'utilisation d'un algorithme pour la préparation á un calcul sur ordinateur, on dispose de 4 types d'instructions :
- l'affectation des variables
- la lecture écriture
- les tests
- les boucles

1.6.1 Types de variables

Il existe 3 types de variables :

Numérique
- Entiers (simple ou double précision)
- Réels (simple ou double précision)
- Complexes (n'existe pas dans tous les langages)

Alphanumérique

– Chaîne de caractères

Booléens
– Variable logique

En Python, le type des variables est implicite et est définit automatiquement lors de l'affectation.

1.6.2 Tableaux

Pour les ***tableaux*** et les ***structures***, qui sont des collections d'éléments (de même type ou pas) on parle de types composites.

Déclaration et accessibilité d'un tableau

Au premier élément d'un tableau correspond l'indice 0. On peut déclarer un tableau de façon ***statique*** :

$$vecteur = Tableau(1..MAX)\ en\ Réels$$

ou de façon ***dynamique*** :

$$vecteur = Tableau()\ en\ Réels$$

Ici, *vecteur* est un tableau de MAX+1 éléments réels dont le premier vaut $vecteur[0] = 1$. On accédera à l'élément i du tableau *vecteur* de la façon suivante :

$$vecteur[i] \leftarrow 10$$

$$x \leftarrow vecteur[i]$$

Cette notation est ce que l'on appelle du **pseudo-code**

Définition
Une expression est un ensemble de valeurs liées par des opérateurs, équivalant à une seule valeur.

Définition
Un opérateur est un signe (symbolisant une opération) qui relie deux valeurs pour produire un résultat. Comme les variables, il existe des opérateurs numériques, alphanumériques et booléens.

Type d'opérandes	Opérateurs disponibles	Type de résultat
Booléen	non, et, ou, =, $\neq$	Booléen
Entier, Naturel	+, -, $\times$, div, mod	Entier
	=, $\neq$, <, >, $\geq$, $\leq$	Booléen
	+, -, $\times$, div, mod, /	Réel
Réel	+, -, $\times$, /	Réel
	=, $\neq$, <, >, $\geq$, $\leq$	Booléen
Caractère	suiv, prec	Caractère
	ord	Naturel
Naturel (nombre associé au caractère dans une chaîne)	char	Booléen
Chaîne de caractères	+	Chaîne de caractères
	=, $\neq$, <, >, $\geq$, $\leq$	Booléen

Définition
Une Action/Instruction est un élément dont l'interprétation modifie l'état d'un programme.

Définition
L'affectation d'une variable en pseudo-code s'écrit comme suit :

$$variable \leftarrow expression$$

On affecte ainsi la valeur de l'*expression* à la *variable*. L'*expression* et la *variable* doivent être de même type.

Définition
On appelle ***lecture*** l'action de lire une expression *écrite par l'utilisateur* et affichée à l'écran (ou dans un fichier).
On appelle ***écriture*** l'action d'écrire une expression *lue á l'écran ou dans un fichier par l'utilisateur.*

1.6.3 Organisation d'un algorithme

Un algorithme consiste à décrire une suite de schémas d'actions qui serviront à décomposer le problème complexe en une série de sous-problèmes simples et que l'on sait résoudre (étapes de la méthodologie de programmation).
Il existe 3 types de schémas comme indiqué plus haut :

Schéma séquentiel
Suite d'actions enchaînées. Cet enchaînement d'actions peut aussi être considéré comme une seule action.

Schéma conditionnel
Suite d'actions enchaînées si et seulement si une condition donnée C est vérifiée.

Schéma itératif
Suite d'actions qui sont répétées un nombre de fois donné.

1.6.4 Tests

Un test simple fait intervenir un *booléen*, qui est soit une variable booléenne, soit une *condition*, c'est-à-dire la combinaison *valeur + opérateur de comparaison + autre valeur*.

Si *booléen* **Alors**
Instruction
FinSi

Si *booléen* **Alors**
Instruction
Sinon
Instruction
FinSi

Un test imbriqué sera composé d'un enchaînement de conditions :

Si *condition* **Alors**
Instruction
SinonSi *condition* **Alors**
Instruction
Sinon
Instruction
FinSi

Les parenthèses dans une condition complexe impliquant les opérateurs **em ET** et ***OU*** influent sur l'ordre d'application des conditions.

Une condition exprimée avec ***ET*** peut toujours être remplacée par une condition équivalente avec ***OU***.

On définit la table de vérité comme suit :

C_1 ***ET*** C_2	C_1 V	C_1 F
C_2 V	V	F
C_2 F	F	F

C_1 ***OU*** C_2	C_1 V	C_1 F
C_2 V	V	V
C_2 F	V	F

C_1 ***XOR*** C_2	C_1 V	C_1 F
C_2 V	F	V
C_2 F	V	F

1.6.5 Schéma Itératif

Boucles simples

Définition

Une boucle est une structure permettant l'exécution répétitive d'une instruction ou d'une série d'instructions.

Il existe deux types de boucles correspondant à deux cas de figure différents :

1. Le nombre d'itérations est connu à priori : ***Boucle déterministe***

   Pour *compteur* ← *Initial* à *Final* ***Pas*** *valeur du pas*
   ...
   instructions
   ...
   compteur ***Suivant***

2. Le nombre d'itérations n'est pas connu à priori : ***Boucle indéterministe***

   Tantque *Booléen*
   ...
   instructions
   ...
   FinTantQue

Boucles imbriquées

Lorsque l'on veut répéter des instructions à l'intérieur d'une boucle, on utilise une ***boucle imbriquée***.

Un pseudo-code pour deux boucles imbriquées s'écrira

Pour *compteur1* ← *Initial1* à *Final1* ***Pas*** *valeur du pas*
...
instructions
...
Pour *compteur2* ← *Initial2* à *Final2* ***Pas*** *valeur du pas*
...
instructions
...
compteur2 ***Suivant***

compteur1 ***Suivant***

Dans le cas des boucles imbriquées, il est particulièrement important de ne pas mélanger nom du compteur et nom des variables sur lesquelles on réalise des opérations.

En Python les instructions dites de clôture ne sont pas explicitées. Elles sont symbolisées par des retraits de ligne.

Chapitre 2

Dérivation numérique

Sommaire

2.1 Dérivation numérique

Dans la dérivation numérique, le but est d'estimer la dérivée d'une fonction non-analytique par exemple définie par un ensemble discret de N points de mesure $\{(x_i, y_i = f(x_i))\}_N$.
Pour estimer la dérivée d'une fonction décrivant le comportement d'un ensemble de points discrets, il existe deux méthodes principales :

- estimer la dérivée numériquement en utilisant directement les points de mesure
- modéliser la fonction décrivant les points de mesure par une fonction simple de dérivée connue et dériver cette fonction.

2.1.1 Rappels et définitions

Définition

Soit $f(x)$ une fonction définie sur un intervalle ouvert contenant x_0. On dira de la fonction f qu'elle est ***dérivable*** en x_0 si

$$lim_{x \to x_0} \frac{f(x) - f(x_0)}{x - x_0}$$

existe. Lorsque cette limite existe, elle est notée $f'(x_0)$ et est appelée ***dérivée*** de la fonction f en x_0.
De façon équivalente on peut définir la dérivée par la limite suivante

$$lim_{h \to 0} \frac{f(x_0 + h) - f(x_0)}{h} = f'(x_0)$$

où $h = x - x_0$.

On notera que le nombre $m = f'(x_0)$ est la pente de la droite tangeante à la courbe représentant la fonction $y = f(x)$ au point $(x_0, y_0 = f(x_0))$.

Théorème

Soit $f \in C^{n+1}[a, b]$, et $x_0 \in [a, b]$. Alors pour tout $x \in [a, b]$, il existe un nombre $c = c(x)$ compris entre x_0 et x tel que

$$f(x) = P_n(x) + R_x(x)$$

où :

$$P_n(x) = \sum_{k=0}^{n} \frac{f^{(k)}(x_0)}{k!}(x - x_0)^k$$

et :

$$R_n(x) = \frac{f^{(n+1)}(c)}{(n+1)!}(x - x_0)^{n+1})$$

Corollaire

Si $P_n(x)$ est le polynôme de Taylor de degré n défini par le théorème de Taylor, alors

$$P_n^{(k)}(x_0) = f^{(k)}(x_0) \quad \text{pour } k = 0, 1, ..., n$$

2.1.2 Différences finies

Pour pouvoir calculer des dérivées sur un ensemble de points discrets, il faut passer de la formulation continue de la dérivée à la formulation discrète de la différentielle. Pour cela on modélise la dérivée en utilisant des représentations approchées tirées du développement de Taylor de bas ordre.

Développement de Taylor d'ordre 1 de la fonction f au voisinage de x_0 :

$$f(x) = f(x_0) + (x - x_0)\frac{f'(x_0)}{1!} + \sum_{k=2}^{\infty}(x - x_0)^k \frac{f^{(k)}(x_0)}{k!}$$

En posant $h = x - x_0$, on peut représenter la dérivée de f en x_0 par la ***différence finie en avant*** :

$$f'(x_0) = \frac{f(x_0 + h) - f(x_0)}{h} + o(h)$$

De même, en posant $h' = x_0 - x$, on peut aussi définir la différence finie en arrière :

$$f'(x_0) = \frac{f(x_0) - f(x_0 - h)}{h} + o(h')$$

Dans le cas d'un maillage ***régulier***, c'est-à-dire pour des points équidistants, où $h = h'$, et en combinant ces deux expressions de différences finies à deux points, on peut estimer la dérivée de la fonction f au point x_0 par un ***schéma de différences finies centrées à 3 points***

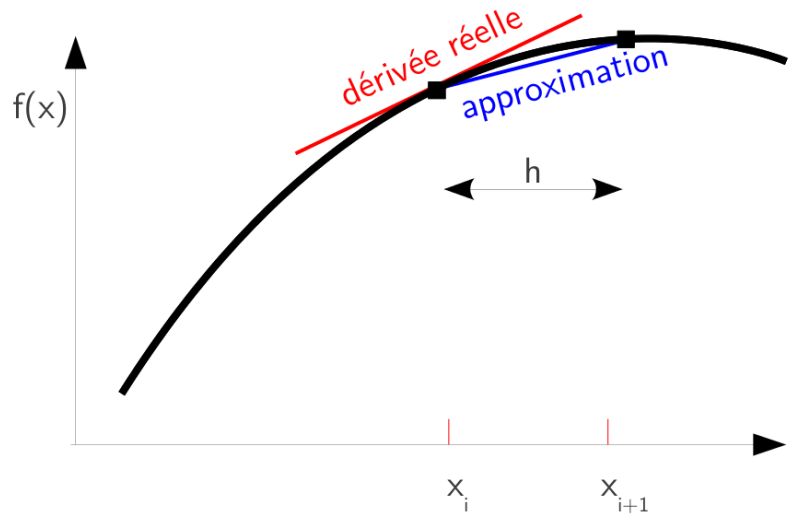


FIGURE 2.1 – Représentation graphique de l'approximation de la dérivée première par une différence finie en avant.

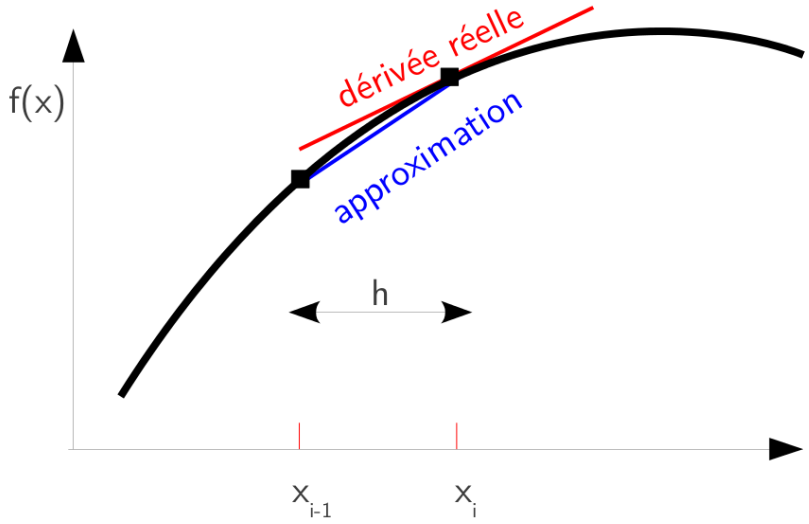


FIGURE 2.2 – Représentation graphique de l'approximation de la dérivée première par une différence finie en arrière.

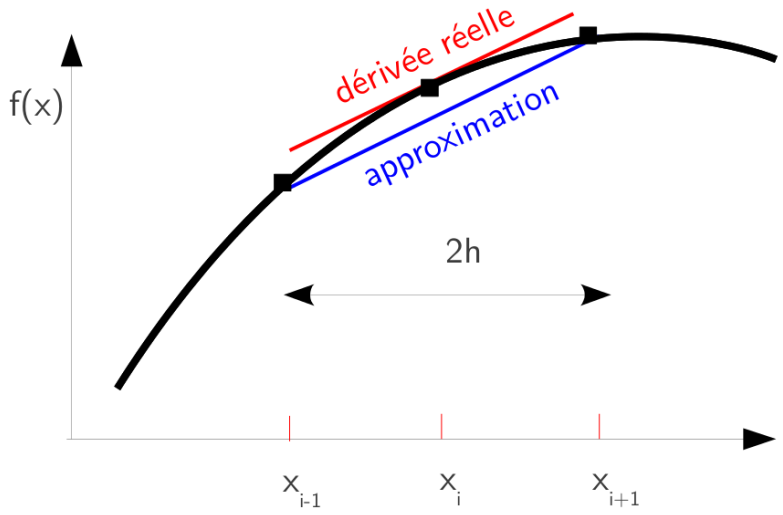


FIGURE 2.3 – Représentation graphique de l'approximation de la dérivée première par une différence finie centrée à 3 points.

$$f'(x_0) = \frac{f(x_0+h) - f(x_0-h)}{2h} + o(h^2)$$

La dérivée d'ordre supérieur est alors estimée en utilisant ce même schéma qui est maintenant appliqué à la dérivée première :

$$f''(x_0) = \frac{f(x_0+h) - 2f(x_0) + f(x_0-h)}{h^2} + o(h^2)$$

Ce schéma entraîne des effets de bords qui augmentent avec l'ordre de la dérivée.

Dans de nombreux cas, le maillage régulier n'est pas le mieux adapté et il est préférable d'utiliser un maillage adaptatif ***irrégulier***. Les données issues d'expériences sont quant à elles en général distribuées de manière non-uniforme dans l'espace consiéré, et on aura par défaut à faire à un maillage irrégulier.

Dans le cas où la fonction à dériver est connue mais où il est préférable pour des raisons techniques, scientifiques ou numériques, d'utiliser un maillage irrégulier, on pourra définir

$$h_i = x_{i+1} - x_i \text{ et } h_{i-1} = x_i - x_{i-1} \text{ avec } h_i \neq h_{i-1}$$

Une estimation de la dérivée première de f par le schéma des différences finies est alors

$$f'_i = \frac{h_{i-1} f_{i+1} + (h_i - h_{i-1}) f_i - h_i f_{i-1}}{2 h_i h_{i-1}}$$

où on a utilisé la notation $f_i = f(x_i)$.

Dans le cas de données expérimentales réparties de manière non-uniforme, une autre façon de faire est d'utiliser un polynôme d'interpolation de Lagrange d'ordre deux (voir Ch. 3) à des groupes de 3 points adjacents. On peut alors dériver le polynôme du second ordre pour obtenir une autre expression de la dérivée première :

$$\begin{aligned} f'(x) &= f(x_{i-1})\frac{2x - x_i - x_{i+1}}{(x_{i-1} - x_i)(x_{i-1} - x_{i+1})} + f(x_i)\frac{2x - x_{i-1} - x_{i+1}}{(x_i - x_{i-1})(x_i - x_{i+1})} \\ &\quad + f(x_{i+1})\frac{2x - x_{i-1} - x_i}{(x_{i+1} - x_{i-1})(x_{i+1} - x_i)} \end{aligned}$$

où x est le point auquel on veut estimer la dérivée. Cette formulation a la même précision que l'expression dérivée du schéma des différences finies centrées à 3 points.

Chapitre 3

Interpolation, Extrapolation, Ajustement

Sommaire

En physique expérimentale les systèmes sont décrits par des valeurs discrètes d'une fonction $f(x)$ en un nombre donné de points $x_1, x_2, ..., x_N$, sans pour autant que la forme analytique de cette fonction soit connue. Ceci implique que l'on ne peut pas a priori évaluer cette fonction en un point arbitraire x.

La plupart du temps, on essaiera d'estimer la valeur $f(x)$ de la fonction en un point arbitraire x en traçant une courbe qui connecte les points connus $f(x_i)$ et en la prolongeant éventuellement. On parlera d'***ajustement*** de courbe lorsqu'on connecte les points connus, d'***interpolation*** lorsqu'on cherche à estimer $f(x)$ poux

$x \in [x_1, x_N]$, et d'***extrapolation*** quand $x \notin [x_1, x_N]$.

La fonction approximant f sera sélectionnée pour avoir une forme simple à manipuler (notamment à dériver et/ou à intégrer) et pour être applicable à un grand nombre de cas.
Pour cette raison, on utilisera souvent des ***polynômes***. Les fonctions ***trigonométriques*** ou ***rationnelles*** sont aussi souvent employées.

3.1 Interpolation linéaire

Le polynôme d'interpolation le plus simple est la droite.

L'interpolation par une droite ou ***interpolation linéaire*** s'utilise en général dans un contexte très local.

Définition

Soit f une fonction continue et dérivable sur un sous-intervalle $[x_i, x_{i+1}]$ de l'intervalle de définition de f, soit [a,b] $= [x_0, x_N]$, et c un élément de cet intervalle. Alors la valeur f en c peut être estimée par la droite d'interpolation en ce point :

$$f(c) = pf_j + (1-p)f_{j+1} \quad \forall \;\; 1 \leq j \leq N-1$$

où

$$p = \frac{x_{j+1} - c}{x_{j+1} - x_j}.$$

NB : Ce type d'interpolation peut produire une fonction avec une dérivée première discontinue.

3.2 Rappels sur les polynômes

3.2.1 Approximation polynomiale

Définition

Soit $n \geq 0$ un nombre entier. Étant donné n+1 couples distincts $(x_0, y_0), (x_1, y_1), ...,$, on cherche le polynôme de degré n, p_n tel que

$$p_n(x_j) = y_j \;\; \forall j \in [0, n].$$

Si ce polynôme existe, on le note P_n et on l'appelle ***polynôme d'interpolation*** aux points x_j pour $j \in [0, n]$.

Définition

Soit f une fonction continue sur l'intervalle I, et $x_0, x_1, ..., x_n$ $n+1$ points discrets de I. Si on prend $n+1$ valeurs y_j telles que $y_j = f(x_j)$, alors le polynôme d'interpolation est noté $P_n f(x)$ et est appelé ***interpolant*** de f aux points $x_0, x_1, ..., x_n$. Ces points sont appelés ***points de collocation***.

3.2.2 Factorisation polynomiale

Lorsqu'un polynôme comporte des coefficients très différents ou qu'il est d'ordre élevé, il est difficile de l'évaluer numériquement.
Dans ce cas, on utilise préférentiellement la forme imbriquée du polynôme considéré, dite ***forme de Horner*** :

$$P_N(x) = a_0 + x(a_1 + x(a_2 + x(a_3 + ... + x(a_{N-1} + xa_N))))$$

Cette forme récursive obéit à un algorithme simple :

$$Y_0 \leftarrow x$$
$$Y_{N-k+1} \leftarrow a_{N-k} + a_{N-k+1} Y_{N-k}$$

pour tout $k \in [1, N]$.

3.3 Familles de polynômes remarquables

3.3.1 Polynômes de Lagrange

Lorsqu'on se donne $N+1$ points $(x_0, y_0), (x_1, y_1), ...(x_N, y_N)$ et qu'on cherche un polynôme $P(x)$ tel que $P(x_0) = y_0, ..., P(x_N) = y_N$, alors :

- Il existe une infinité de polynômes $P(x)$ qui satisfont ces relations.
- Le polynôme de plus bas degré est de degré N et il est ***unique***.
- Ce polynôme peut s'écrire comme une ***combinaison linéaire de polynômes de Lagrange***.

Les polynômes de Lagrange se notent L_j et sont définis par les relations suivantes :

$$L_{N,j}(x_i) = \delta_{i,j} \ \forall i, j = 0, 1, 2, ..., N$$
$$L_{N,j} = \prod_{i=0, i \neq j}^{N} \frac{(x - x_i)}{(x_j - x_i)}$$

Le polynôme

$$P_N(x) = \sum_{j=0}^{N} f(x_j) L_{N,j}(x)$$

est le seul polynôme de degré N interpolant les données y_k aux points x_k pour $k \in [0, N]$.

Erreur de l'approximation par les polynômes de Lagrange

Soient $f \in C^{N+1}[a,b]$ et $x_0, x_1, ..., x_N$ $N+1$ noeuds de cette fonction sur l'intervalle [a,b]. Si $x \in [a,b]$, alors on peut écrire

$$f(x) = P_N(x) + E_N(x)$$

où $P_N(x)$ est une combinaison de polynômes de Lagrange de degré au plus N qui peut être utilisée pour approximer $f(x)$ et qui vérifie :

$$f(x) \approx P_N(x) = \sum_{k=0}^{N} f(x_k) L_{N,k}(x)$$

Le terme d'erreur s'écrit alors :

$$E_N(x) = \frac{\prod_{j=0}^{N}(x - x_j) f^{(N+1)}(c)}{(N+1)!}$$

où $c = c(x)$ appartient à l'intervalle de définition [a,b].

3.3.2 Polynômes de Newton

Les polynômes de Newton se construisent à partir de la relation de récurrence suivante :

$$\mathcal{N}_N(x) = \mathcal{N}_{N-1}(x) + a_N \prod_{k=0}^{N-1} (x - x_k)$$

Les coefficients des polynômes de Newton sont déterminés en utilisant le formalisme des différences divisées à partir des deux premiers termes de la suite :

$$\mathcal{N}_0(x_0) = a_0 = f(x_0)$$

$$\mathcal{N}_1(x_1) = a_0 + a1(x_1 - x_0) = f(x_1)$$

soit

$$a1 = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

Si l'on note $f[x_i] \equiv f(x_i)$, on peut alors définir la ***première différence divisée*** comme

$$f[x_i, x_{i+1}] = \frac{f[x_{i+1}] - f[x_i]}{x_{i+1} - x_i}$$

Avec cette notation, après $k-1$ différences divisées, on obtient

$$f[x_i, x_{i+1}, x_{i+2}, ..., x_{i+k-1}, x_{i+k}] = \frac{f[x_{i+1}, x_{i+2}, ..., x_{i+k-1}, x_{i+k}] - f[x_i, x_{i+1}, x_{i+2}, ..., x_{i+k-1}]}{x_{i+k} - x_i}$$

et $a_k = f[x_0, x_1, ..., x_k]$.

Par exemple on aura

$$a_1 = f[x_0, x_1] = \frac{f[x_1] - f[x_0]}{x_1 - x_0} = \frac{f(x_1) - f(x_0)}{x_1 - x_0}$$

$$a_2 = f[x_0, x_1, x_2] = \frac{f[x_1, x_2] - f[x_0, x_1]}{x_2 - x_0} = \frac{\frac{(f(x_2)-f(x_1)}{x_2-x_1} - \frac{f(x_1)-f(x_0)}{x_1-x_0}}{x_2 - x_0}$$

Polynômes de Newton comme polynômes d'interpolation

Définition

Soient $x_0, x_1, ..., x_N$ $N+1$ nombres distincts de l'intervalle [a,b]. Il existe un polynôme unique $P_N(x)$ de degré au plus N vérifiant :

$$f(x_j) = P_N(x_j) \ \forall \ j = 0, 1, ..., N$$

La ***forme de Newton*** de ce polynôme est :

$$P_N(x) = a_0 + \sum_{k=1}^{N} a_k \prod_{j=1}^{N-1} (x - x_j)$$

avec $a_k = f[x_0, x_1, ..., x_k]$ pour $k = 0, 1, ..., N$.

Si $P_N(x))$ est utilisé pour approcher la fonction f sur l'intervalle $[a, b]$, et si $f \in C^{N+1}[a, b]$, alors pour chaque $x \in [a, b]$, il existe $c = c(x)$ dans ce même intervalle tel que :

$$E_N(x) = \frac{f^{(N+1)}(c) \prod_{k=0}^{N}(x - x_k)}{(N+1)!}$$

Les formes de Lagrange et de Newton permettent de trouver un polynôme $P(x)$ qui prend les mêmes valeurs $y = f(x)$ que la fonction f en N points donnés.

3.3.3 Polynômes de Taylor

La ***forme de Taylor*** permet de construire un polynôme qui coïncide en x_0 avec la valeur de $f(x_0)$ de la fonction f, et dont ***les dérivées coïncident en plus également avec celles de la fonction f jusqu'au degré N si on construit le polynôme avec $N+1$ points discrets.***

La forme de Taylor se définit comme suit :

$$P_N(x - x_0) = \sum_{i=0}^{N} a_i (x - x_0)^{(i)}$$

avec

$$a_i = \frac{f^{(i)}(x_0)}{i!} \ \forall \ i = 0, 1, ..., N$$

3.4 Polynômes d'interpolation - Analyse des erreurs

Définition

Soient $x_0, x_1, ..., x_N$ $N+1$ noeuds distincts de l'intervalle $I = [a, b]$, et soit f une fonction continue et $N+1$ fois dérivable sur I. Alors pour tout $x \in I$ on peut définir l'erreur d'interpolation de f par le polynôme $P_N f(x)$ [1] par la relation suivante :

$$f(x) - P_N f(x) = \omega_{N+1}(x) \frac{f^{(N+1)}(\xi)}{(N+1)!}$$

où $\xi \in I$ et

$$\omega_{N+1}(x) = \prod_{i=0}^{N} (x - x_i)$$

Dans l'expression de $\omega_{N+1}(x)$, on remarque que le produit est tel que l'erreur est nulle en chacun des points d'interpolation (ou des points de collocation), et que cette erreur serait égale au prochain polynôme de Newton $\mathcal{N}_{N+1}$ si on venait à rajouter un autre point de collocation.

Dans le cas particulier où les noeuds sont ***équirépartis***, on a :

$$E_N(f) = \max_{x \in I} \ |f(x) - P_N f(x)| \leq \frac{1}{4(N+1)} \left(\frac{b-a}{N} \right)^{N+1} \max_{x \in I} \left| f^{(N+1)}(x) \right|$$

Dans tous les cas, l'erreur d'interpolation dépend de la dérivée d'ordre $N+1$ de la fonction f et de la fonction $\omega_{N+1}(x)$.

Il existe globalement deux cas de figure pour lesquels l'erreur d'interpolation polynomiale peut diverger :

1. Le cas où la fonction f que l'on cherche à interpoler est une *mauvaise* fonction en ce sens par exemple qu'elle n'est pas continûment dérivable sur l'intervalle $[a, b]$ considéré. Il suffit par ailleurs d'un point où la fonction n'est pas dérivable pour que l'erreur croisse sur *tout l'intervalle*.
   Dans ce cas on peut éventuellement minimiser les effets de la *mauvaise* définition de la dérivée en choisissant bien les polynômes d'interpolation ou les points d'interpolation. En général, les *mauvaises* fonctions, qui ne satisfont pas aux pré-requis mathématiques nécessaires à l'application des formules données précédemment, résistent à l'analyse numérique quel que soit le degré de sophistication des méthodes utilisées.

1. On utilise ici la notation dans le cas d'un polynôme interpolant une fonction continue.

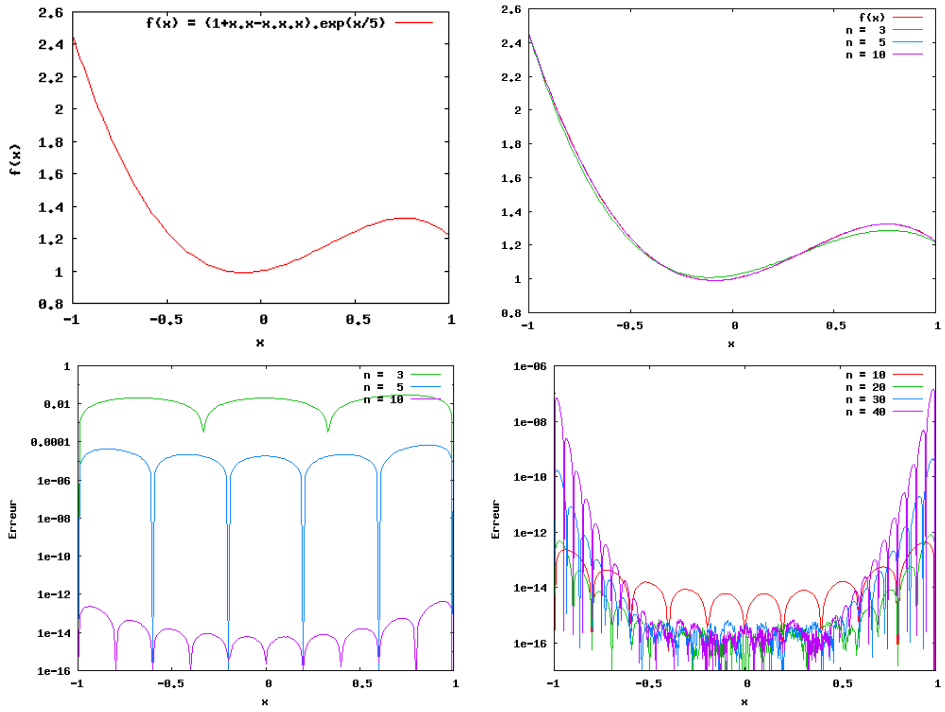


FIGURE 3.1 – Illustration du phénomène de Runge dans le cas d'une interpolation avec $n+1$ points de collocation équirépartis sur l'intervalle [-1,1].

2. Le cas où la fonction a une dérivée qui croît très vite dans les ordres élevés, c'est-à-dire une fonction pour laquelle plus on prend de points de collocation, plus grande est l'erreur. On parle alors de ***phénomène de Runge***. Ce phénomène apparaît notamment dans le cas où on a des noeuds équirépartis comme ceux utilisés pour les interpolations avec des polynômes de Lagrange, Newton et Taylor. Dans ces cas, l'erreur *diverge* lorsque le degré du polynôme d'interpolation augmente.
   La croissance de l'erreur peut être en partie contrôlée par le choix rigoureux des points d'interpolation.

3.4.1 Sélection des points d'interpolation - polynômes de Chebyshev

En examinant la formule d'erreur pour l'interpolation polynomiale donnée ci-dessus, on peut voir qu'un moyen de réduire l'erreur consiste à sélectionner les noeuds d'interpolation $x_0, x_1, ..., x_N$ de façon à minimiser le produit $\prod_{i=0}^{N}(x - x_i)$. Pour ce faire, on va plutôt chercher à maximiser le nombre de points sur les bords de l'intervalle et en prendre moins au centre.
La distribution de points qui minimise ce produit est appelée ***distribution de Chebyshev***.

Pour construire la distribution de Chebyshev, on peut procéder graphiquement de la façon suivante [2] :

1. On trace un demi-cercle sur l'intervalle $[a, b]$
2. Pour échantillonner $N+1$ points, on dispose $N+1$ points de façon équidistante sur le demi-cercle
3. On projette chaque point sur l'axe des abscisses :

$$x_j = \frac{b-a}{2} \dot{cos}\left(j\frac{\pi}{N}\right) + \frac{a+b}{2} \quad \forall \quad j = 0, 1, ..., N$$

Ces points sont associés à la base des polynômes de Chebyshev qui sont définis par la relation de récurrence suivante :

$$\begin{aligned} \mathcal{T}_0(x) &= 1 \\ \mathcal{T}_1(x) &= x \\ \mathcal{T}_k(x) &= 2x\mathcal{T}_{k-1}(x) - \mathcal{T}_{k-2}(x) \quad \forall\ k = 2, ..., N \end{aligned}$$

3.4.2 Réduction de l'erreur - Interpolation par les fonctions splines

L'interpolation polynomiale sur un ensemble important de points peut être assez insatisfaisante du fait du phénomène de Runge. Une façon de contourner ce

2. Cette méthode est décrite dans le cours de Pr Amos RON de l'Université du Wisconsin

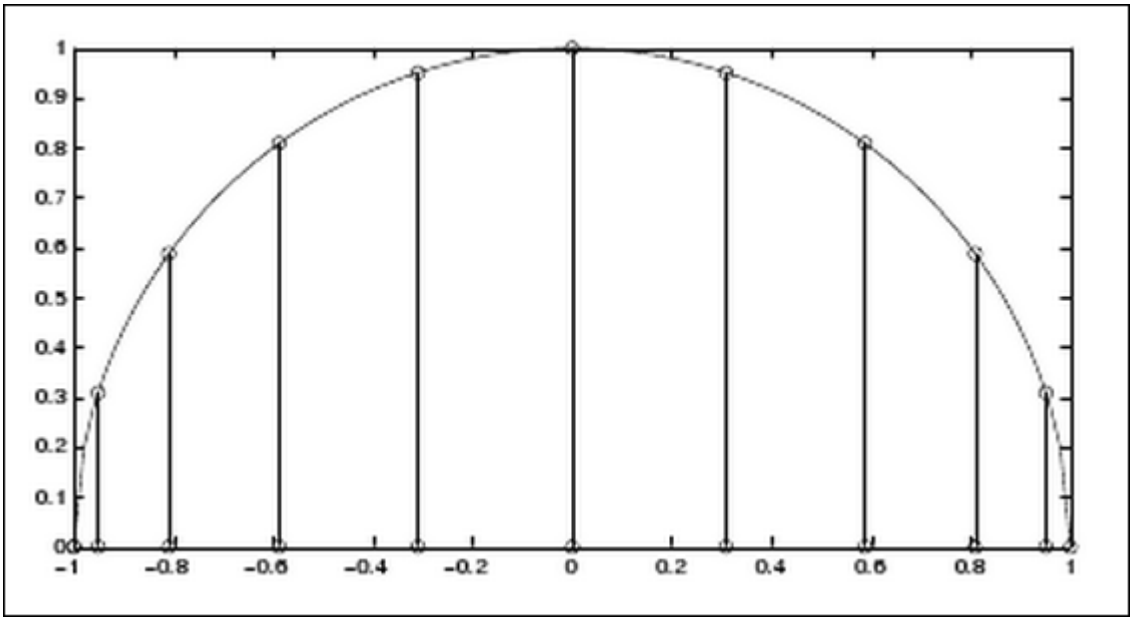


FIGURE 3.2 – Distribution de Chebyshev sur l'intervalle [-1,1].

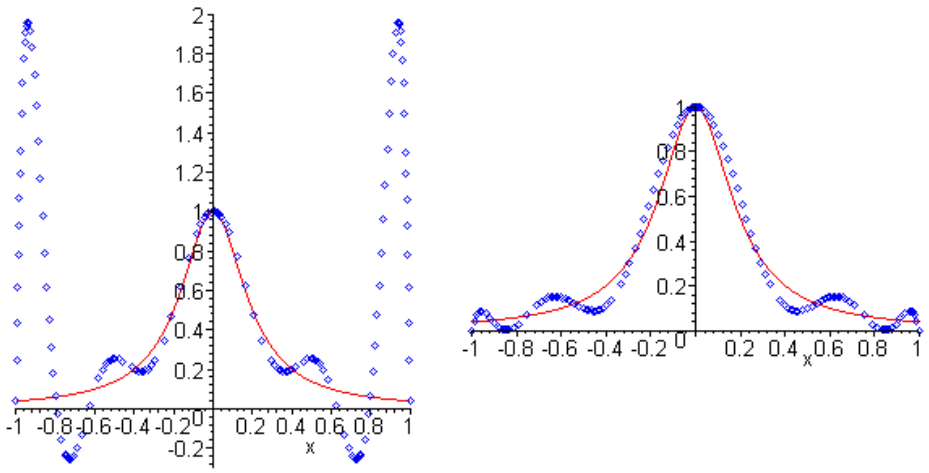


FIGURE 3.3 – Illustration de la réduction du phénomène de Runge par l'utilisation de la distribution de Chebyshev dans le choix des points de collocation. Cas de la fonction $f(x) = \frac{1}{1+25x^2}$ avec 10 points de collocation de Chebyshev.

problème peut alors être d'interpoler la fonction par morceaux, par des fonctions polynomiales de plus bas degré $S_k(x)$ sur des sous-intervalles $[x_k, x_{k+1}]$ de l'intervalle de définition d'origine. On choisira alors les fonctions de sorte qu'en chacun des noeuds on ait $y_{k+1} = S_k(x_{k+1}) = S_{k+1}(x_{k+1})$, avec $S_k(x)$ la fonction interpolante sur le sous-intervalle $[x_k, x_{k+1}]$, et $S_k(x_{k+1})$ la fonction interpolante sur le sous-intervalle $[x_{k+1}, x_{k+2}]$.
Ces fonctions sont appelées ***splines***. Les formes les plus utilisées sont les ***splines linéaires*** et les ***splines cubiques***.

3.4.2.1 Splines linéaires

Les polynômes les plus simples à utiliser pour interpoler une fonction par morceaux sont des droites (polynômes de degré 1). On peut définir les fonctions $S_k(x)$ en utilisant le polynôme de Lagrange pour représenter les droites de l'interpolation par morceaux :

$$S_k(x) = y_k \frac{x - x_{k+1}}{x_k - x_{k+1}} + y_{k+1} \frac{x - x_k}{x_{k+1} - x_k} \quad \text{pour} \quad x_k \leq x \leq x_{k+1}$$

On peut aussi utiliser une expression équivalente obtenue avec la formule de la pente pour un segment de droite :

$$S_k(x) = y_k + d_k(x - x_k) = y_k + \frac{(y_{k+1} - y_k)}{(x_{k+1} - x_k)}(x - x_k)$$

La fonction ***spline linéaire*** qui en résulte peut s'écrire de la façon suivante :

$$S(x) = \begin{cases} y_0 + d_0(x - x_0), & \text{for } x \text{ in } [x_0, x_1] \\ y_1 + d_1(x - x_1), & \text{for } x \text{ in } [x_1, x_2] \\ \vdots & \vdots \\ y_k + d_k(x - x_k)), & \text{for } x \text{ in } [x_k, x_{k+1}] \\ \vdots & \vdots \\ y_{N-1} + d_{N-1}(x - x_{N-1})), & \text{for } x \text{ in } [x_{N-1}, x_N] \end{cases}$$

3.4.2.2 Splines cubiques

Ces fonctions splines permettent d'avoir une interpolation continue par morceaux d'une fonction par des polynômes de degré 3, c'est-à-dire qui est telle que les dérivées première et seconde peuvent être continues.

Définition

Soient $n+1$ points $(x_k, y_k)_{k=0}^n$, avec $a = x_0 < x_1 < \cdots < x_n = b$.
La fonction S est appelée ***spline cubique*** interpolant une fonction f définie sur l'intervalle I s'il existe n polynômes cubiques $S_k(x)$ de coefficients $s_{k,0}, s_{k,1}, s_{k,2}, s_{k,3}$ satisfaisant les propriétés suivantes :

1. $S(x) = S_k(x) = \sum_{j=0}^{3} s_{k,j}(x - x_k)^j$ pour $x \in [x_k, x_{k+1}]$ et $k = 0, 1, ..., N-1$
2. $S(x_k) = f(x_k) = y_k \quad \forall k \in 0, 1, ..., n$
3. $S_k(x_{k+1}) = S_{k+1}(x_{k+1}) \quad \forall k \in 0, 1, ..., n-2$
4. $S'_k(x_{k+1}) = S'_{k+1}(x_{k+1}) \quad \forall k \in 0, 1, ..., n-2$
5. $S''_k(x_{k+1}) = S''_{k+1}(x_{k+1}) \quad \forall k \in 0, 1, ..., n-2$

De ces propriétés découlent 4n -2 relations reliant $4n$ inconnues. Pour déterminer la spline cuibique de façon unique il faut donc définir 2 conditions supplémentaires. Leur définition va déterminer la construction de la spline cubique et sa forme.

On pourra ainsi définir la ***spline naturelle*** en posant $S''_3(x_0)_0 = 0 = S'''_3(x_n)_{n-1}$.

Détermination de la spline

$S_3 \in \mathbb{P}_3$ implique que sa dérivée seconde est de degré 1 et est déterminée par les relations :

$$m_i = f''(x_i) \qquad m_{i+1} = f''(x_{i+1})$$

En notant $h_i = x_{i+1} - x_i$ et $I_i = [x_i, x_{i+1}]$, et si $S_i(x)$ est le polynôme de degré 3 qui coïncide avec la spline sur le sous-intervalle I_i, alors après deux intégrations

successives de la dérivée seconde de la spline, et en utilisant la notation $y_i = f(x_i)$, la spline $S_i(x)$ sur le sous-intervalle I_i s'écrit :

$$S_i(x) = \frac{m_i}{6h_i}(x_{i+1}-x)^3 + \frac{m_{i+1}}{6h_i}(x-x_i)^3 + \left(\frac{y_i}{h_i} - \frac{m_i h_i}{6}\right)(x_{i+1}-x) + \left(\frac{y_{i+1}}{h_i} - \frac{m_{i+1}h_i}{6}\right)(x-x_i)$$

Il reste à déterminer les coefficients m_i. On peut montrer que pour des points équirépartis, $h_i = h \ \ \forall i$, et en utilisant les conditions de la spline naturelle qui sont que $m_0 = m_1 = 0$, les coefficients m_i sont obtenus en résolvant le système suivant :

$$\begin{pmatrix} 4 & 1 & \cdots & 0 \\ 1 & 4 & 1 & \cdots \\ \vdots & \ddots & \ddots & \vdots \\ \cdots & 1 & 4 & 1 \\ 0 & \cdots & 1 & 4 \end{pmatrix} \begin{pmatrix} m_1 \\ m_2 \\ \vdots \\ m_{n-1} \end{pmatrix} = \begin{pmatrix} S_1 \\ S_2 \\ \vdots \\ S_{n-1} \end{pmatrix}$$

On peut aussi définir une matrice de ce type (plus compliquée) pour le cas d'intervalles de taille quelconque.

3.5 Ajustement

En physique, on aura souvent à faire à des ensembles de points entachés d'erreur résultant d'une expérience. On voudra alors établir une *relation de corrélation* entre ces points plutôt que d'utiliser une interpolation. Pour cela on réalise un ***ajustement***.

On va réaliser l'ajustement avec une fonction quelconque. Dans le cas de l'utilisation d'un polynôme, on cherchera un ***polynôme d'ajustement*** $P_k(x)$ de degré k tel que $k+1$ soit inférieur au nombre N de points de l'échantillon.

Procédure

Une procédure d'ajustement se fait en deux étapes :

1. le calcul de la distance d_i entre la fonction d'ajustement $f(x_i)$ et les points expérimentaux y_i.
2. la minimisation de la distance totale $\sum_i |d_i|$.

La fonction d'ajustement se définit comme suit :

$$f(x_k) = y_k + e_k$$

où $e_k = y_k - f(x_k)$ est le résidu que l'on cherche à minimiser.
On a différentes possibilités pour trouver le *meilleur ajustement*, basées sur une description différente de l'erreur.

– Erreur moyenne On cherche à minimiser la somme de toutes les erreurs résiduelles

$$E_1(f) = \frac{1}{N}\sum_{k=1}^{N}(f(x_k) - y_k) = \frac{1}{N}\sum_{k=1}^{N} e_k$$

Ce critère n'est pas adéquat car il ne donne pas une solution unique et ne permet donc pas de déterminer LE meilleur ajustement.
On peut tenter d'améliorer ce critère en minimisant la somme des valeurs absolues des résidus :

$$E_1(f) = \frac{1}{N}\sum_{k=1}^{N}|f(x_k) - y_k| = \frac{1}{N}\sum_{k=1}^{N}|e_k|$$

Ici encore, on peut avoir des solutions multiples pour cette minimisation, et ce critère est lui aussi inadéquat.

– Erreur maximale Une troisième stratégie peut consister à minimiser le résidu maximal :

$$E_\infty(f) = \max_{1\leq k\leq N}(|f(x_k) - y_k|) = \max_{1\leq k\leq N}(|e_k|)$$

Cette méthode n'est pas non plus adaptée pour la régression linéaire car elle peut donner beaucoup de poids aux points singuliers entaché d'une erreur importante. Cette méthode peut cependant être bien adaptée dans le cas où on veut ajuster une fonction compliquée par une fonction simple.

– Erreur des moindres carrés
Une façon d'outrepasser les faiblesses des critères mentionnés ci-avant est de minimiser la somme des carrés des résidus.

$$E_2(f) = \left(\frac{1}{N}\sum_{k=1}^{N}|f(x_k) - y_k|^2\right)^{1/2} = \left(\frac{1}{N}\sum_{k=1}^{N}|e_k|^2\right)^{1/2}$$

Ce critère a de nombreux avantages dont celui de conduire à une *solution unique* pour ensemble de points donné.

3.5.1 Méthode des moindres carrés : Régression linéaire

La fonction la plus simple pour un ajustement est une droite d'équation :

$$p_1(x) = a_0 + a_1 x$$

Dans une procédure de recherche du meilleur ajustement d'un ensemble de m points (x_k, y_k), $k = 1, ...m$ au sens es moindes carrés, il s'agit donc de minimiser :

$$E(a_0, a_1) = \sum_{k=1}^{m}|y_k - a_0 - a_1 x_k|^2$$

Pour cela on doit déterminer les coefficients a_0 et a_1, ce qui peut se faire en cherchant les valeurs qui annuelent les dérivées partielles de $E(a_0, a_1)$:

$$\begin{aligned}
\frac{\partial \Phi}{\partial a_0} = 0 \quad &\rightarrow \quad \left(\sum_{k=1}^{m} x_k\right) a_1 + m a_0 = \sum_{k=1}^{m} y_k \\
\frac{\partial \Phi}{\partial a_1} = 0 \quad &\rightarrow \quad \left(\sum_{k=1}^{m} x_k^2\right) a_1 + \left(\sum_{k=1}^{m} x_k\right) a_0 = \sum_{k=1}^{m} x_k y_k
\end{aligned}$$

Ces équations sont appelées *équations normales* et admettent pour solution :

$$\begin{aligned}
a_1 &= \frac{m \sum_{k=1}^{m} x_k y_k - \sum_{k=1}^{m} x_k \sum_{k=1}^{m} y_k}{m \sum_{k=1}^{m} x_k^2 - \left(\sum_{k=1}^{m} x_k\right)^2} \\
a_0 &= \frac{1}{m} \sum_{k=1}^{m} y_k - \frac{a_1}{m} \sum_{k=1}^{m} x_k
\end{aligned}$$

3.5.2 Méthode des moindres carrés : Linéarisation de relations non-linéaires

La régression linéaire est une méthode puissante pour trouver le meilleur ajustement à un ensemble de données si tant est qu'elles suivent un comportement à peu près linéaire. Dans le cas contraire, on peut choisir soit (a) de linéariser la fonction représentant les données (b) appliquer un ajustement polynomial (toujours au sens des moindres carrés).
Les données pouvant être décrites par des formes exponentielles, lois de puissance ou des formes rationnelles seront préférentiellement étudiés via une régression linéaire après linéarisation.

- forme exponentielle linéarisée en appliquant la fonction ln
- loi de puissance linéarisée en appliquant la fonction logarithme $\log_{10}$
- forme rationnelle (de type $y = \frac{ax}{b+x}$) linéarisée par inversion.

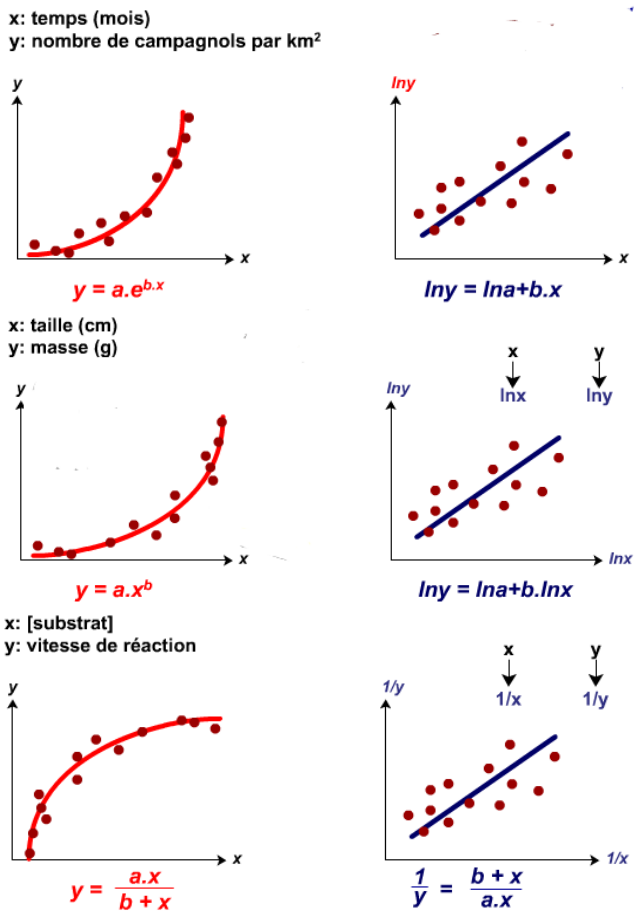


FIGURE 3.4 – Illustration de la linéarisation de lois exponentielle, de puissance et rationnelle.

3.5.3 Méthode des moindres carrés : Régression polynomiale

Faisant naturellement suite au cas particulier de la régression linéaire, on peut définir de même un ajustement polynomial au sens des moindres carrés à des ensemble de données quie ne peuvent être décrit par une loi linéaire ou linéarisable.

Définition

On appelle ***polynôme des moindres carrés*** $p_n(x)$, le polynôme de degré n tel que, pour m couples de mesures (x_i, y_i) :

$$\sum_{i=0}^{m} |y_i - p_n(x_i)|^2 \leq \sum_{i=0}^{m} |y_i - q_n(x_i)|^2 \quad \forall \quad q_n(x) \in \mathbb{P}_n$$

où $\mathbb{P}_n$ est l'espace des polynômes de degré n.

En d'autres termes, ce polynôme de degré n est celui qui, parmi tous les polynômes de degré n, minimise la distance des données.
Si on note $p_n(x) = a_0 + a_1 x + ... + a_n x^n$, et qu'on définit la fonction

$$\Phi(a_0, ..., a_n) = \sum_{i=0}^{m} |y_i - (a_0 + a_1 x_i + ... + a_n x_i^n)|^2$$

Les coefficients a_k du polynôme des moindres carrés sont alors donnés par les relations :

$$\frac{\partial \Phi}{\partial a_k} = 0$$

Cela donne n relations linéaires entre les coefficients a_k :

$$\frac{\partial \Phi}{\partial a_k} = -2 \left[\sum_{i=0}^{m} x_i^k y_i - \left(a_0 \sum_{i=0}^{m} x_i^k + a_1 \sum_{i=0}^{m} x_i^{k+1} + ... + a_n \sum_{i=0}^{m} x_i^{k+n} \right) \right] \quad \forall \ k = 0, 1, ..., n$$

En combinant ces deux relations, on obtient un système linéaire qui peut se mettre sous forme matricielle $Aa = y$, soit encore en forme développée

$$\begin{pmatrix} m+1 & \cdots & \sum_{i=0}^{m} x_i^m \\ \vdots & \ddots & \vdots \\ \sum_{i=0}^{m} x_i^n & \cdots & \sum_{i=0}^{m} x_i^{2n} \end{pmatrix} \begin{pmatrix} a_0 \\ \vdots \\ a_n \end{pmatrix} = \begin{pmatrix} \sum_{i=0}^{m} y_i \\ \vdots \\ \sum_{i=0}^{m} y_i x_i^n \end{pmatrix}$$

Si on prend $n = 1$, on retrouve la ***régression linéaire***.

3.6 Rappels de notions probabilistes

Définition

Une ***variable aléatoire X*** est une application sur l'espace des événements associés à une expérience. Elle prendra les valeurs discrètes $x_1, x_2, ..., x_N$ associées aux événements $e_1, e_2, ..., e_N$ de probabilité $p_1, p_2, ..., p_N$.

Définition

On définit l' ***espérance mathématique*** d'une variable aléatoire X par la relation suivante :

$$E(X) = \sum_{i=1}^{N} p_i x_i = \overline{X}$$

L'espérance est le premier moment de la variable aléatoire X.

Définition

On définit la ***variance*** d'une variable aléatoire X par la relation suivante :

$$V(X) = \sigma^2 = \sum_{i=1}^{N} p_i x_i^2 - \left(\sum_{i=1}^{N} p_i x_i \right)^2 = \sum_{i=1}^{N} p_i x_i^2 - E(X)$$

Il s'agit là du second moment de la variable aléatoire X.

$\sigma = \sqrt{V(X)}$ est l'***écart-type*** de la variable aléatoire X.

3.7 Test d'ajustement du χ^2

Le test du d'ajustement du χ^2 est basé sur la statistique du χ^2 établie par Pearson. Cette statistique consiste à mesurer l'écart qui existe entre la distribution des effectifs théoriques t_i et la distribution des effectifs observés n_i , et à voir si cet écart est suffisamment faible pour être imputable aux fluctuations d'échantillonage.
Dans le cas du test d'ajustement du χ^2, la statistique est la suivante :

$$\chi^2_{obs} = \sum_{i=1}^{k} \frac{(n_i - t_i)^2}{t_i}$$

Si n est l'effectif total étudié, alors la valeur théorique attendue t_i pour la $i$ème réalisation de la variable aléatoire X est

$$t_i = n * p_i$$

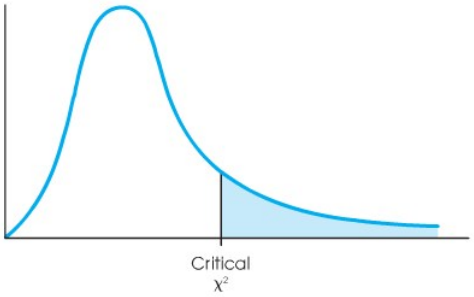
.

La stratégie d'un test du χ^2 est la suivante :

On fait l'hypothèse H_0 que les valeurs de l'échantillon observé suivent une loi F et on combine valeurs observées et valeurs théoriques (suivant effectivement la loi F) de sorte que suivant le résultat obtenu, on puisse décider avec un niveau de confiance $1-\alpha \in]0,1[$ donné, **soit de rejeter l'hypothèse H_0, soit de l'accepter.** α est en général petit et choisi par l'expérimentateur . Ce nombre représente la probabilité d'accepter l'hypothèse H_0 alors qu'elle est fausse : c'est le *risque d'erreur*.

La statistique du χ^2 calculée χ^2_{obs} est comparée à la valeur seuil χ^2_{seuil} lue dans la table du χ^2 pour $k-1$ degrés de liberté et pour un risque d'erreur α fixé.

*The table entries are critical values of χ^2.



k-1 *df*	Risque d'erreur 0.10	0.05	0.025	0.01	0.005
1	2.71	3.84	5.02	6.63	7.88
2	4.61	5.99	7.38	9.21	10.60
3	6.25	7.81	9.35	11.34	12.84
4	7.78	9.49	11.14	13.28	14.86
5	9.24	11.07	12.83	15.09	16.75
6	10.64	12.59	14.45	16.81	18.55
7	12.02	14.07	16.01	18.48	20.28
8	13.36	15.51	17.53	20.09	21.96
9	14.68	16.92	19.02	21.67	23.59
10	15.99	18.31	20.48	23.21	25.19
11	17.28	19.68	21.92	24.72	26.76
12	18.55	21.03	23.34	26.22	28.30
18	25.99	28.87	31.53	34.81	37.16
19	27.20	30.14	32.85	36.19	38.58
20	28.41	31.41	34.17	37.57	40.00
21	29.62	32.67	35.48	38.93	41.40
22	30.81	33.92	36.78	40.29	42.80
23	32.01	35.17	38.08	41.64	44.18
24	33.20	36.42	39.36	42.98	45.56
25	34.38	37.65	40.65	44.31	46.93
26	35.56	38.89	41.92	45.64	48.29
27	36.74	40.11	43.19	46.96	49.64
28	37.92	41.34	44.46	48.28	50.99
29	39.09	42.56	45.72	49.59	52.34
30	40.26	43.77	46.98	50.89	53.67
40	51.81	55.76	59.34	63.69	66.77
50	63.17	67.50	71.42	76.15	79.49
60	74.40	79.08	83.30	88.38	91.95
70	85.53	90.53	95.02	100.42	104.22
80	96.58	101.88	106.63	112.33	116.32
90	107.56	113.14	118.14	124.12	128.30
100	118.50	124.34	129.56	135.81	140.17

FIGURE 3.5 – Table du χ^2. Les colonnes correspondent au risque d'erreur α, et les lignes au nombre de degrés de liberté $k-1$

3.7.1 Conditions d'application du test

Le test du χ^2 est valable sous certaines restrictions :
- le nombre de valeurs théoriques doit être égal au nombre de valeurs observées.
- L'échantillon étudié doit être de grande taille ($n \geq 50$)
- il est conseillé qu les produits $t_i = np_i$, c'est-à-dire les valeurs théoriques, soient égales ou supérieures à 5.

3.7.2 Utilisation du test d'ajustement du χ^2

En ce qui concerne le test lui-même il se met en œuvre de la façon suivante :

L'hypothèse nulle testée est :
- H_0 : la distribution observée est conforme à la istribution théorique
- H_1 : la distribution observée ne suit pas la distribution théorique

On calcule :

$$\chi^2_{obs} = \sum_{i=1}^{k} \frac{(n_i - t_i)^2}{t_i}$$

avec k le nombre de valeurs de l'échantillon, n_i les valeurs observées, et t_i les valeurs théoriques attendues sous l'hypothèse H_0.

La valeur obtenue est comparée à la valeur seuil χ^2_{seuil} donnée dans la table de χ^2 pour $k-1$ et pour le risque d'erreur choisi α.

Si $\chi^2_{obs} \leq \chi^2_{seuil}$, l'hypothèse H_0 ne peut être rejetée : les distributions observée et théorique ne sont pas significativement différentes.
Si $\chi^2_{obs} > \chi^2_{seuil}$, l'hypothèse H_0 est rejetée au seuil de signification α, et l'hypothèse H_1 est acceptée.

3.7.3 Ajustement à des lois de probabilité connues

Loi binomiale Si on considère que la distribution théorique suit une loi binomiale $B(n,p)$ avec n le nombre d'épreuces, p la probabilité de succès et k le nombre de réalisations de la variable aléatoire X , alors on a :

$$p_k = P(X = k) = C_n^k p^k q^{n-k}$$

d'où on peut calculer les valeurs théoriques $t_i = np_i$.

Loi de Poisson Si on considère une série d'événements rares se produisant en moyenne un nombre λ de fois pendant une période T, alors la variable aléatoire X déterminant le nombre de fois où se produit l'événement sur la période T, X prenant des valeurs entières positives, obéit à une loi de

probabilité de Poisson de paramètre λ :

$$p(k) = P(X = k) = e^{-\lambda}\frac{\lambda^k}{k!} \quad \forall\ k \geq 0$$

L'espérance associée vaut λ et l'écart-type $\sqrt{\lambda}$.

Pour une série de données supposées obéir à une loi de Poisson, on cherchera donc à minimiser :

$$\chi^2 = \sum_{i=0}^{n} \frac{(x_i - \lambda)^2}{\lambda}$$

Chapitre 4

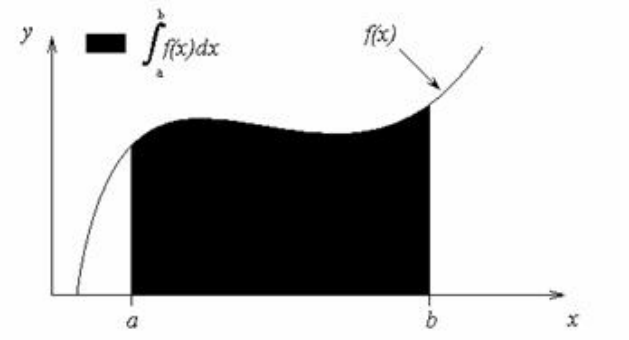
Intégration numérique

Sommaire

Le but de ce chapitre est de pouvoir résoudre le plus précisément possible l'équation

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

– Si I n'a pas de forme analytique, on utilisera une procédure de ***quadrature*** : on calcule *l'aire sous la courbe*.



– Si la primitive de la fonction f est connue, on préférera reformuler le problème et résoudre une équation différentielle. Dans ce cas, on cherchera une solution de l'équation

$$\frac{dy}{dx} = f(x)$$

au point $I \equiv y(b)$ avec la condition limite $y(a) = 0$.

4.1 Quadratures

Définition

Soit un ensemble de points discrets sur l'intervalle [a,b], avec $a = x_0 < x_1 < \cdots < x_n = b$. Alors une formule de type

$$Q[f] = \sum_{k=0}^{n} \omega_k f(x_k)$$

avec la propriété que

$$\int_a^b f(x)dx = Q[f] + E[f]$$

est appelée ***intégration numérique*** ou ***formule de quadrature***.

Le terme $E[f]$ est appelé ***erreur de troncature*** pour l'intégration.
Les points $\{x_i\}_{i=0}^n$ sont appelés les ***noeuds*** de la quadrature.
Les points $\{omega_i\}_{i=0}^n$ sont appelés les ***poids*** de la quadrature.

Les noeuds seront choisis en fonction de l'application Q. Dans le cas des ***quadratures de Newton-Cotes***, les noeuds sont équirépartis.

4.1.1 Quadratures de Newton-Cotes

Définition

On appelle ***quadratures de Newton-Cotes*** les formules de quadrature qui utilisent l'intégrale du polynôme d'interpolation de Lagrange pour évaluer l'intégrale d'une fonction $f(x)$ définie par $n+1$ couples $(x_k, f(x_k))$, avec x_k équirépartis sur l'intervalle $[a, b]$.
Lorsque les points $x_0 = a$ et $x_n = b$ sont utilisés, on parle de formule de Newton-Cotes ***fermée***.

Théorème

Pour un ensemble de $n+1$ points x_k équirépartis de sorte que $x_k = x_0 + hk$ avec $h = (x_n - x_0)/n$, et pour les valeurs $f_k = f(x_k)$ de la fonction f associées, les quatre premières formules de quadratures fermées de Newton-Cotes sont :

– Méthode des trapèzes

$$\int_{x_0}^{x_1} f(x)dx = \frac{h}{2}\left[f_0 + f_1\right] + o\left(h^3 f^{(2)}\right)$$

– Méthode de Simpson

$$\int_{x_0}^{x_2} f(x)dx = \frac{h}{3}\left[f_0 + 4f_1 + f_2\right] + o\left(h^5 f^{(4)}\right)$$

– Méthode de Simpson 3/8

$$\int_{x_0}^{x_3} f(x)dx = \frac{3h}{8}\left[f_0 + 3f_1 + 3f_2 + f_3\right] + o\left(h^5 f^{(4)}\right)$$

– Méthode de Boole

$$\int_{x_0}^{x_4} f(x)dx = \frac{2h}{45}\left[7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4\right] + o\left(h^7 f^{(6)}\right)$$

4.1.2 Degré d'exactitude, ordre et erreur de troncature

Définition

On dira que la formule de quadrature $Q_n(f)$ est ***d'ordre*** $q \geq 1$ ***par rapport à la largeur*** $h = b - a$ ***de l'intervalle d'intégration*** $I = [a, b]$, si pour toute fonction f régulière on a

$$\left| \int_a^b f(x)dx - Q_n(f)) \right| = o\left(h^q\right)$$

Pour la formule de Simpson, $q = 5$ si $f \in C^4([a, b])$.

Définition

Le ***degré de précision*** d'une formule de quadrature est l'entier positif n tel que $E[P_i] = 0$ pour tous les polynômes $P_i(x)$ de degré $i \leq n$, mais pour lesquels $E[P_{n+1}] \neq 0$ pour un polynôme $P_{n+1}(x)$ de degré $n + 1$.

la forme de $E[P_i]$ peut s'anticiper en étudiant ce qu'il se passe lorsque $f(x)$ est un polynôme.
On considère un polynôme arbitraire

$$P_i(x) = a_i x^i + \cdots + a_2 x^2 + a_1 x + a_0$$

de degré i. Si $i \leq n$, alors $P_i^{(n+1)} \equiv 0$ pour tout i, et $P_{n+1}^{(n+1)}(x) = (n+1)!a_{n-1}$ pour tout x. Il en découle que la forme générale pour le terme d'erreur de troncature est

$$E[f] = Kf^{(n+1)}(c)$$

où K est une constante et n est le degré de précision.

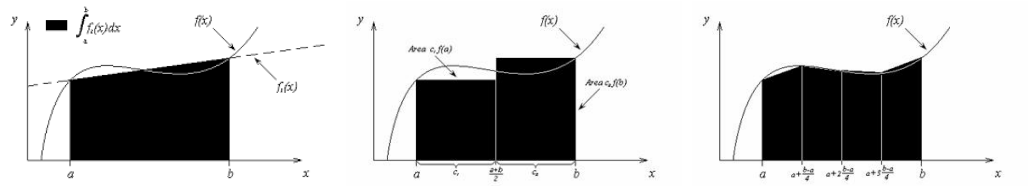


FIGURE 4.1 – Illustration de l'augmentation de la précision de l'intégration numérique par la décomposition de l'intervalle d'intégration en sous-intervalles de même largeur.

Plus précisément, pour une formule de quadrature $Q_n[f]$ telle que

$$\int_a^b f(x)dx = Q_n[f] + E_n[f] = Q_n(x) + E_n(x)$$

l'erreur de troncature $E_n[f] = E_n(x)$ est obtenue en intégrant l'erreur d'interpolation $e_n[f] = e_n(x)$, soit

$$E_n(x) = \int_a^b e_n(x)dx = \int_a^b \frac{f^{(n+1)}(\xi)}{(n+1)!} \prod_{i=0}^{n}(x - x_i)dx \quad \text{pour} \quad \xi \in [a, b]$$

En posant $x - a = uh$ avec $dx = hdu$, et en tenant compte du fait que $x_i = a + ih$ (noeuds équirépartis), on obtient :

$$E_n[f] = \frac{h^{(n+2)}}{(n+1)!} f^{(n+1)}(\xi) \int_0^{u_b} \prod_{i=0}^{n}(u - i)du \quad \text{pour} \quad \xi \in [a, b]$$

4.2 Méthodes composites

Pour réduire l'erreur d'intégration d'une fonction par une formule de quadrature, on peut décomposer l'intervalle d'intégration en une somme de sous-intervalles contigus dans lesquels l'approximation de la fonction par un polynôme de bas degré(par exemple une droite) est plus réaliste.

4.2.1 Quadratures de Newton-Cotes composites

4.2.1.1 Méthode des trapèzes composites

Théorème

Soit un intervalle $[a, b]$ subdivisé en N sous-intervalles $[x_k, x_{k+1}]$ de largeur $h = (b - a)/N$ par des noeuds équirépartis $x_k = a + kh \ \forall k \in 0, 1, ..., N$.
La méthode composite des trapèzes sur ces N sous-intervalles peut alors s'exprimer de 3 façons différentes :

$$T(f,h) = \frac{h}{2}\sum_{k=0}^{N-1}\left(f(x_k) + f(x_{k+1})\right)$$

$$T(f,h) = \frac{h}{2}\left(f_0 + 2f_1 + ... + 2f_{N-1} + f_N\right)$$

$$T(f,h) = \frac{h}{2}\left(f(a) + f(b)\right) + h\sum_{k=1}^{N-1} f(x_k)$$

avec toujours

$$\int_a^b f(x)dx \approx T(f,h) = T(f,h) + E_T(f,h) = T(f,h) + o(h^2)$$

4.2.1.2 Méthode de Simpson composite

Théorème

Soit un intervalle $[a,b]$ subdivisé en $2N$ sous-intervalles $[x_k, x_{k+1}]$ de largeur $h = (b-a)/2N$ par des noeuds équirépartis $x_k = a + kh \ \forall k \in 0, 1, ..., 2N$.
La méthode composite de Simpson sur ces $2N$ sous-intervalles peut alors s'exprimer de 3 façons différentes :

$$S(f,h) = \frac{h}{3}\sum_{k=1}^{N} f(x_{2k-2}) + 4f(x_{2k-1}) + f(x_{2k})$$

$$S(f,h) = \frac{h}{3}\left(f_0 + 4f_1 + 2f_2 + 4f_3 + ... + 2f_{2N-2} + 4f_{2N-1} + f_{2N}\right)$$

$$S(f,h) = \frac{h}{3}\left(f(a) + f(b)\right) + \frac{2h}{3}\sum_{k=1}^{N-1} f(x_{2k}) + \frac{4h}{3}\sum_{k=1}^{N} f(x_{2k-1})$$

avec toujours

$$\int_a^b f(x)dx \approx S(f,h) = S(f,h) + E_S(f,h) = S(f,h) + o(h^4)$$

4.2.2 Ordre des formules composites

Soit un intervalle $I = [a,b]$ subdivisé en m sous-intervalles de largeur $h = (b-a)/m$. On note $Q_{n,m}(f)$ la formule composite de la quadrature $Q_n(f)$:

$$Q_{n,m}(f) = \sum_{k=0}^{m-1} Q_n(f,h)$$

, où $Q_n(f,h)$ est la forme de quadrature utilisée sur chacun des sous-intervalles $[x_k, x_{k+1}]$.

Si dans chaque sous-intervalle , la formule de quadrature est d'ordre q, c'est-à-dire

$$\left| \int_{x_k}^{x_{k+1}} f(x)dx - Q_n(f,h) \right| = Ch^q$$

alors la formule de quadrature composite est d'ordre $q-1$ l'erreur globale sur la formule composite est donnée par :

$$\left| \int_a^b f(x)dx - Q_{n,m}(f) \right| = \left| \sum_{k=0}^{m-1} \left(\int_{x_k}^{x_{k+1}} f(x)dx - Q_n(f,h) \right) \right| \leq \sum_{k=0}^{m-1} Ch^q = Cmh^q = C(b-a)h^{q-1}$$

4.3 Formules récursives et intégration de Romberg

Pour les quadratures de Newton-Cotes fermées, on voit que l'estimation de l'intégrale est d'autant meilleure qu'on utilise un nombre important de sous-intervalles.

4.3.1 Formules récursives des trapèzes

Théorème

Soit $J \geq 1$ tel que les points $x_k = a + kh$ subdivisent l'intervalle $[a,b]$ en $2^J = 2M$ sous-intervalles de même largeur $h = (b-a)/2^J$.
Les formules des trapèzes $T(f,h)$ et $T(f,2h)$ obéissent à la relation

$$T(f,h) = \frac{T(f,2h)}{2} + h \sum_{k=1}^{M} f(x_{2k-1})$$

Définition

On définit

$$T(0) = \frac{h}{2}\left(f(a) + f(b)\right)$$

qui représente la formule des trapèzes pour un pas d'intégration $h = b-a$. Alors pour chaque $J \geq 1$, on peut définir $T(J) = T(f,h)$ où, $T(f,h)$ est la formule des trapèzes avec un pas d'intégration $h = (b-a)/2^J$. La séquence des formules des trapèzes $\{T(J)\}$ est générée de façon récursive par la formule :

$$T(J) = \frac{T(J-1)}{2} + h \sum_{k=1}^{M} f(x_{2k-1}) \quad \text{pour} \quad J = 1, 2, ...$$

où $h = (b-a)/2^J$ et $\{x_k = a + kh\}$

4.3.2 Formules récursives de Simpson

Théorème

Soit $\{T(J)\}$ la séquence des formules récursives des trapèzes. Si $J \geq 1$ t que l'on note $S(J)$ la formule de Simpson pou 2^J sous-intervalles de $[a, b]$, alors on peut définir $S(J)$ à partir des formules des trapèzes $T(J)$ et $T(J-1)$

$$S(J) = \frac{4T(J) - T(J-1)}{3} \quad \text{pour} \quad J = 1, 2, \cdots$$

4.3.3 Intégration de Romberg

Nous avons vu précédemment que les formes composites des quadratures de Newton-Cotes étaient d'ordre $o(h^2)$ et $o(h^4)$ pour la formule des trapèzes et la formule de Simpson respectivement, et par généralisation d'ordre $o(h^{2N})$ pour une quadrature de Newton-Cotes.
L'intégration de Romberg est utilisée pour accroître l'ordre de l'erreur de $o(h^{2N})$ à $o(h^{2N+2})$, et ainsi améliorer les quadratures de Newton-Cotes. Cette méthode repose sur le ***schéma d'extrapolation de Richardson***, qui est une méthode d'accélération de la convergence.

Cette méthode d'intégration a des avantages et des inconvénients : l'avantage principal est que dans l'intégration de Romberg, tous les poids sont positifs et que les abscisses équidistantes sont simples à calculer. L'inconvénient principal est que pour augmenter l'ordre de l'erreur, on doit procéder à 2 fois plus d'opérations.
Lemme

Étant donné deux approximations $R(2h, K-1)$ et $R(h, K-1)$ d'une quantité (fonction) Q qui satisfont respectivement les relations :

$$Q = R(h, K-1) + c_1 h^{2K} + c_2 h^{2K+2} + \cdots$$

et

$$Q = R(2h, K-1) + c_1 4^K h^{2K} + c_2 4^{K+1} h^{2K+2} + \cdots$$

alors une amélioration de la formule d'approximation est donnée par :

$$Q = \frac{4^K R(h, K-1) - R(2h, K-1)}{4^K - 1} + o(h^{2K+2})$$

Définition

On peut définir la séquence $\{R(J, K) \mho J \geq K\}_{J=0}^{\infty}$ des formules de quadrature pour $f(x)$ sur l'intervalle $[a, b]$ comme suit :

$$R(J,0) = T(J) \quad \text{pour } J \geq 0 \quad \text{est la formule des trapèzes séquentielle}$$

$$R(J,1) = S(J) \quad \text{pour } J \geq 1 \quad \text{est la formule de Simpson séquentielle}$$

$$R(J,2) = B(J) \quad \text{pour } J \geq 2 \quad \text{est la formule de Boole séquentielle}$$

Une formule de quadrature améliorée est alors donnée par :

$$R(J,K) = \frac{4^K R(J,K-1) - R(J-1,K-1)}{4^K - 1} \quad \text{pour } J \geq K$$

On peut représenter l'intégration de Romberg par un tableau :

J	$R(J,0)$ Trapezoidal rule	$R(J,1)$ Simpson's rule	$R(J,2)$ Boole's rule	$R(J,3)$ Third improvement	$R(J,4)$ Fourth improvement
0	$R(0,0)$				
1	$R(1,0)$	$R(1,1)$			
2	$R(2,0)$	$R(2,1)$	$R(2,2)$		
3	$R(3,0)$	$R(3,1)$	$R(3,2)$	$R(3,3)$	
4	$R(4,0)$	$R(4,1)$	$R(4,2)$	$R(4,3)$	$R(4,4)$

FIGURE 4.2 – Tableau des intégrations de Romberg

Pour une fonction $f \in C^{2K+2}([a,,b])$, l'erreur de troncature de l'intégration de Romberg est donnée par

$$\int_a^b f(x)dx = R(J,K) + b_k h^{2K+2} f^{(2K+2)}(c_{J,K}) = R(J,K) + o(h^{2K+2})$$

avec b_k une constante fonction de K, $c_{J,K}$ un point de l'intervalle $[a,b]$ et $h = (b-a)/2^J$.

4.4 Quadratures de Gauss

Les quadratures de Newton-Cotes sont valables pour des noeuds équirépartis. Cependant dans certains cas on préférera utiliser des quadratures qui se basent sur des noeuds répartis non-uniformément. c'est le cas des quadratures de Gauss.

L'idée des quadratures de Gauss est d'obtenir des formules d'intégration d'ordre élevé en se donnant la liberté de choisir non seulement les poids, mais également

les noeuds de la quadrature, que l'on ne choisira plus également espacés.

Les poids et les noeuds des quadratures de Gauss sont tels que **l'intégration numérique sera exacte pour des intégrants de type** $P_N(x)W(x)$, et non plus seulement pour ceux du type $P_N(x)$, comme c'est le cas des quadratures de Newton-Cotes.

Définition

Soit $P_N(x)$ un polynôme de degré N tel que

$$\int_a^b P_N(x)W(x)x^k dx = 0$$

où k est un entier quelconque sur l'intervalle [0,N-1] et W est une ***fonction de pondération***.
Soient $\{x_i\}$ les racines du polynôme $P_N(x)$.
En construisant les formules d'intégration

$$\int_a^b P_N(x)W(x)dx \approx \sum_{i=1}^{N} w_i f(x_i)$$

on est assuré que pour des ensembles donnés de $\{w_i\}$, l'approximation est exacte si $f(x)$ est un polynôme de degré inférieur à $2N$.
Les ***noeuds*** $\{x_i\}$ de la quadrature de Gauss à N points sont ***les racines du polynôme orthogonal à*** f sur le ***même intervalle*** avec les ***mêmes poids***.

Pour l'ensemble des quadratures de Gauss, il existe des tables donnant les noeuds et les poids, qui peuvent être trouvées dans des ouvrages de référence.

Intervalle	Poids $W(x)$	Récurrence	Nom
$[-1,1]$	1	$(i+1)P_{i+1} = (2i+1)xP_i - iP_{i-1}$	Gauss-Legendre
$[-1,1]$	$\sqrt{(1-x^2)}$	$T_{i+1} = 2XT_i - T_{i-1}$	Gauss-Chebyschev
$[0,\infty[$	$x_c e^{-x}$	$(i+1)L_{i+1}^c = (-x+2i+c+1)L_i^c$	Gauss-Laguerre (c = 0,1,...)
$]-\infty,\infty[$	e^{-x^2}	$H_{i+1} = 2xH_i - 2iH_{i-1}$	Gauss-Hermite

Parmi les quadratures de Gauss, les plus simples et communes sont les les quadratures de Gauss-Legendre.
La quadrature de Gauss-Legendre à N points comporte une fonction de pondération $W(x) = 1$ et des noeuds $\{x_i\}$ qui sont les racines du polynôme de Legendre de degré N sur l'intervalle d'intégration.

4.4.1 Quadrature de Gauss-Legendre à 3 points

$$\int_{-1}^{1} f(x)dx \approx \sum_{i=1}^{3} w_i f(x_i)$$

On cherche le polynôme de degré 3 tel que

$$\int_{-1}^{1} P_3(x)dx = \int_{-1}^{1} P_3(x)xdx = \int_{-1}^{1} P_3(x)x^2dx = 0$$

avec $P_3(x) = c_0 + c_1x + c_2x^2 + c_3x^3$.

En combinant ces deux relations on obtient le système suivant :

$$\begin{aligned} 2c_0 + \frac{2}{3}c_2 &= 0 \\ \frac{2}{3}c_1 + \frac{2}{5}c_3 &= 0 \\ \frac{2}{3}c_0 + \frac{2}{5}c_2 &= 0 \end{aligned} \tag{4.1}$$

Ce système est sous-déterminé (3 équations pour 4 inconnues), et il faut se donner l'un des coefficients pour pouvoir le résoudre.
On se donne donc $c_1 = 3/2$. Les deux premières équations de ce système impliquent $c_0 = c_2 = 0$. En faisant les substitutions pour c_1, c_2 et c_0, on obtient ainsi le polynôme de Legendre de degré 3 :

$$P_3(x) = -\frac{3}{2}x + \frac{5}{2}x^3$$

Les noeuds $\{x_i\}$ de la quadrature de Gauss-Legendre sont les racines de $P_3(x)$:

$$x_1 = 0, x_2 = \sqrt{\frac{3}{5}}, x_3 = -\sqrt{\frac{3}{5}}$$

On a donc la formule d'intégration suivante :

$$\int_{-1}^{1} f(x)dx = w_1 f\left(-\sqrt{\frac{3}{5}}\right) + w_2 f(0) + w_3 f\left(\sqrt{\frac{3}{5}}\right)$$

Il faut à présent trouver les poids.
Par définition de la quadrature de Gauss, on sait qu'elle doit être exacte pour tout polynôme de degré < 6, en particulier donc $f(x) = 1, x, x^2, x^3, x^4$ ou x^5. Comme on

cherche 3 poids, on n'utilisera que $f(x) = 1, f(x) = x$ et $f(x) = x^2$.

En résolvant ce système, on obtient finalement

$$w_1 = \frac{5}{9},\ w_2 = \frac{8}{9},\ w_3 = \frac{5}{9}$$

d'où

$$\int_{-1}^{1} f(x)dx = \frac{5f\left(-\sqrt{\frac{3}{5}}\right) + 8f(0) + 5f\left(\sqrt{\frac{3}{5}}\right)}{9} + E_3(f)$$

Pour une intégrale définie sur un intervalle $[a, b]$ quelconque,

$$\int_a^b f(t)dt$$

on peut toujours appliquer un changement de variable de sorte à la définir sur l'intervalle [-1,1]. En posant

$$t =\equiv t(u) = \frac{b+a}{2} + \frac{b-a}{2}u$$

on a aussi

$$dt = \frac{b-a}{2}du$$

et on obtient on obtient

$$\int_a^b f(t)dt = \frac{b-a}{2}\int_{-1}^{1} f(t(u))du$$

L'évaluation de l'intégrale par la méthode de Gauss-Legendre à N points donne alors

$$\int_a^b f(t)dt = \frac{b-a}{2}\int_{-1}^{1} f(u)du \approx \frac{b-a}{2}\sum_{k=1}^{N}\omega_{N,k}f(t(x_{N,k})) \tag{4.2}$$

où $\omega_{N,k}$ et $x_{N,k}$ sont respectivement les poids et les noeuds de la quadrature de Gauss-Legendre à N points.

Chapitre 5

Résolution d'équations aux dérivées ordinaires

Sommaire

5.1 Introduction

Une équation différentielle est une équation impliquant des dérivées. Une ***équation différentielle ordinaire*** est une équation qui ne fait intervenir des dérivées que par rapport à une variable indépendante. Des exemples d'***EDO*** sont

$$\frac{d^2y}{dx^2} + 2\frac{dy}{dx} + y = 0$$

$$\frac{d^3y}{dx^3} + 3\frac{d^2y}{dx^2} + 5\frac{dy}{dx} + y = sinx$$

Ces équations sont respectivement d'ordre 2 et 3, ce qui correspond à l'ordre de la dérivée d'ordre le plus élevé.

Les équations différentielles ordinaires comme celles-ci, qui sont d'ordre N supérieur à 1, peuvent toujours s'écrire sous la forme d'un *système de N équations différentielles ordinaires couplées d'ordre 1.* Ainsi :

$$\frac{d^2y}{dx^2} + q(x)\frac{dy}{dx} = r(x)$$

peut aussi s'écrire :

$$\begin{aligned} \frac{dy}{dx} &= z(x) \\ \frac{dz}{dx} &= r(x) - q(x)z(x) \end{aligned}$$

où z est une nouvelle variable. En général la ou les ***variables auxiliaires*** seront simplement définies comme étant des dérivées les unes des autres (et/ou de la variable d'origine).

Le problème générique des ***EDO*** d'ordre élevé se réduit donc à l'étude d'un système de N ***équations différentielles du premier ordre couplées*** pour les fonctions $y_i,\ i = 1, 2, ..., N$ ayant la forme générale :

$$\frac{dy_i(x)}{dx} = f_i(x, y1, y2, ..., y_N) \quad i = 1, ..., N$$

où les fonctions f_i du membre de droite sont connues.

Ces équations ne déterminent pourtant pas complètement le problème, et doivent être complétées par des conditions limites pour que l'on puisse effectivement résoudre le problème. Les ***conditions limites*** sont des conditions algébriques pour les valeurs des fonctions y_i.
En particulier, pour reprendre le premier exemple, l'équation

$$\frac{d^2y}{dx^2} + 2\frac{dy}{dx} + y = 0$$

doit être complétée par autant de conditions limites qu'il y aura d'équations d'ordre 1 dans le système résultant de la décomposition de cette équation d'ordre 2 :

$$\frac{dy}{dx}(0) = 2 \ \text{ et } \ y(0) = 4$$

En général la nature des conditions limites déterminera le type de méthode numérique à utiliser pour résoudre une ***EDO***. Il en existe de deux types :

– dans les problèmes avec ***conditions initiales***, toutes les fonctions y_i sont connues pour une valeur initiale x_d et on cherche à déterminer les valeurs des y_i en un point final x_f, ou en une série de points discrets

– dans les problèmes avec des ***conditions aux limites***, les fonctions y_i sont connues aux bords du domaine d'intégration, et c'est plutôt les valeurs de ces fonctions à l'intérieur du domaine qui sont recherchées.

5.2 Résolution générale d'une équation différentielle ordinaire

Une équation différentielle ordinaire générale d'ordre n

$$\frac{d^n y}{dx^n} + k_n \frac{d^{n-1} y}{dx^{n-1}} + \cdots + k_3 \frac{d^2 y}{dx^2} + k_2 \frac{dy}{dx} + k_1 y = F(x)$$

admet une solution générale du type

$$y = \underbrace{y_H}_{\text{partie homologue}} + \underbrace{y_P}_{\text{partie particulière}}$$

5.2.1 Partie homologue

La partie homologue représente la solution de l'équation différentielle ordinaire lorsque le membre de droite est nul ($F(x) = 0$) :

$$(D^n + k_n D^{n-1} + \cdots + k_2 D + k_1) y = 0$$

que l'on peut aussi écrire

$$(D - r_n)(D - r_{n-1} \cdots (D - r_1) y = 0$$

où on a adopté la notation $D^n \equiv \frac{d^n}{dx^n}$.

y_H s'exprimera différemment selon que les racines de l'EDO sont réelles distinctes, réelles identiques ou complexes.

Racines réelles distinctes

Une solution de l'EDO dans le cas $F(x) = 0$ est obtenue pour $(D - r_i)y = 0$. Ceci est une ***équation différentielle linéaire du premier ordre de Leibniz***, et la partie homologue y_H pourra alors s'écrire comme la somme des solutions individuelles de Leibniz :

$$y_H = C_1 e^{r_1 x} + C_2 e^{r_2 x} + \cdots + C_{n-1} e^{r_{n-1} x} + C_n e^{r_n x}$$

Racines réelles identiques

Pour m racines identiques de l'EDO, on aura à nouveau une combinaison linéaire de solutions individuelles de Leibniz pour décrire la partie homologue :

$$y_H = \left(C_1 + C_2 x + C_3 x^2 + \cdots + C_m x^{m-1}\right) e^{r_m x} + C_{m+1} e^{r_{m+1} x} + \cdots + C_n e^{r_n x}$$

Racines complexes

Pour une paire de racines complexes de l'EDO, $r_1 = \alpha + i\beta$ et $r_2 = \alpha - i\beta$ la partie homologue s'écrira

$$y_H = e^{\alpha x} \left(A cos(\beta x) + B sin(\beta x)\right) + C_3 e^{r_3 x} + \cdots + C_n e^{r_n x}$$

avec $A = C_1 + iC_2$ et $B = C_1 - iC_2$.

5.2.2 Partie particulière

Pour une équation différentielle ordinaire de forme générale

$$(D^n + k_n D^{n-1} + \cdots + k_2 D + k_1) y = F(x)$$

la partie particulière de la solution, soit y_P, donne $F(x)$ quand on la substitue à y dans cette équation. En général, la partie particulière n'est pas connue et difficile à déterminer analytiquement. Il existe néanmoins quelques cas remarquables :
Dans le cas général où on se sait pas déterminer de façon simple la partie particulière d'une équation différentielle ordinaire, on utilisera des méthodes numériques pour en estimer la solution.

5.3 Équations différentielles ordinaires d'ordre 1

5.3.1 Méthode d'Euler

Soit l'équation différentielle ordinaire du premier ordre

$$\frac{dy}{dx} = f(x, y)$$

assortie de la condition limite de type condition initiale $y(x_0) = y_0$.

En écrivant que la valeur de y au point $x_1 > x_0$ peut-être approximée par un développement limité de Taylor d'ordre 1, on a

$$y_1 = y_0 + \frac{dy}{dx}_{x=x_0} (x_1 - x_0) + o\left((x_1 - x_0)^2\right)$$

En posant $h = x_1 - x_0$, on retrouve

$$y_1 = y_0 + f(x_0, y_0)h + o(h^2)$$

et par la suite, si on se place dans le cas de points équirépartis :

$$y_{i+1} = y_i + f(x_i, y_i)h + o(h^2)$$

Cette relation de récurrence définit la ***méthode d'Euler***.

Précision de la méthode d'Euler - Théorème

Soit $y(x)$ la solution du problème aux conditions initiales suivant

$$\frac{dy}{dx} = f(x, y) \quad \text{avec} \quad y(x_0) = y_0 \quad \text{donn}$$

Si $y(x) \in C^2[x_0, b]$, et que $\{(x_k, y_k)\}_{k=0}^{M}$ est la séquence d'approximations générées par la méthode d'Euler sur l'intervalle d'intégration $[t_0 = a, b]$, alors pour chaque pas d'intégration d'Euler on peut définir

$$|e_k| = |y(x_k) - y_k| = o(h)$$

et

$$|\varepsilon_{k+1}| = |y(x_{k+1}) - y_k - hf(x_k, y_k)| = o(h^2)$$

où $h = (b - a)/M$ est le pas d'intégration.

L'erreur à la fin de l'intervalle d'intégration est appelée ***erreur finale globale*** et vaut :

$$E(y(b), h) = |y(b) - y_M)| = o(h)$$

5.3.2 Méthode de Heun

Pour résoudre l'équation

$$\frac{dy}{dx} = f(x, y(x)) \quad \text{avec} \quad y(x_0) = y_0 \quad \text{sur l}'\text{intervalle } [a, b]$$

on peut également simplement intégrer le membre de gauche :

$$\int_{x_0=a}^{x_1=b} \frac{dy}{dx} dx = \int_{x_0=a}^{x_1=b} f(x, y(x))dx = y(b) - y(a)$$

soit encore :

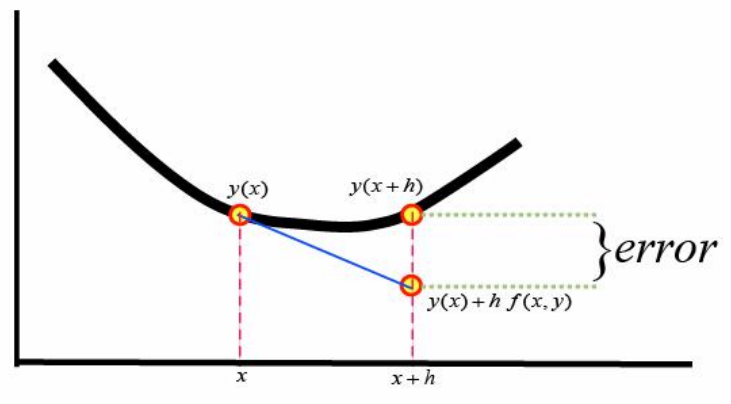


FIGURE 5.1 – Illustration graphique de la résolution par la méthode d'Euler et de l'erreur associée.

$$y(x_1 = b) = y(x_0 = a) + \int_{x_0=a}^{x_1=b} f(x, y(x))dx \tag{5.1}$$

On peut alors utiliser une des méthodes classiques d'intégration numérique pour résoudre l'intégrale du membre de droite. En utilisant la méthode des trapèzes on obtient :

$$y(x_1) \approx y(x_0) + \frac{h}{2}\left(f(x_0, y(x_0)) + f(x_1, y(x_1))\right)$$

Dans cette formulation, on voit encore apparaître dans le membre de droite $y(x_1)$, qui est l'inconnue. On peut l'évaluer ici en utilisant la méthode d'Euler :

$$y(x_1) \approx y(x_0) + \frac{h}{2}\left(f(x_0, y(x_0)) + f(x_1, y_0 + hf(x_0, y_0))\right)$$

Cette formulation est appelée ***méthode de Heun***. Elle constitue également l'une des formulations pour la méthode Runge-Kutta d'ordre 2 qui sont détaillées ci-dessous.

5.3.3 Méthode de Runge Kutta

Avec la méthode d'Euler, on extrapole la dérivée au point x_n pour estimer la valeur de la fonction au point x_{n+1}. C'est une méthode d'ordre 1 et l'erreur est de l'ordre du pas d'intégration.

Si d'un autre côté, on commence par se placer au milieu de l'intervalle de résolution $[x_n, x_{n+1}]$ pour atteindre le point x_{n+1}, on peut diminuer l'erreur :

$$\begin{cases} k_1 = f(x_n, y_n) \\ k_2 = f(x_n + \frac{1}{2}h, y_n + \frac{1}{2}k_1 h) \end{cases} \Rightarrow y_{n+1} = y_n + k_2 h + o(h^3)$$

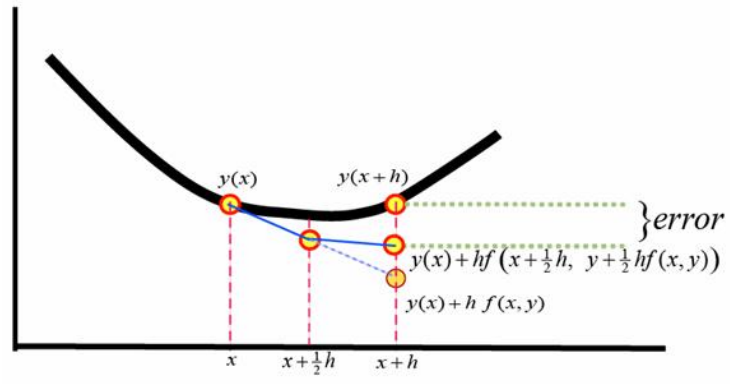


FIGURE 5.2 – Ilustration graphique de la résolution par la méthode du point milieu et de l'erreur associée.

Cette formulation, dite ***méthode du point milieu***, est l'une des formulations de la méthode de Runge-Kutta d'ordre 2.

5.3.3.1 Runge-Kutta d'ordre 2

Pour les méthodes de Runge-Kutta d'ordre 2, on posera

$$y_{i+1} = y_i + a_1 k_1 h + a_2 k_2 h + o(h^3)$$

avec

$$\begin{cases} k_1 = f(x_i, y_i) \\ k_2 = f(x_i + p_1 h, y_i + q_{11} k_1 h) \end{cases}$$

L'expression de k_2 peut s'interpréter en termes d'un développement de Taylor à 2 dimensions où l'incrément (habituellement noté h) en abscisse vaut $p_1 h$ et l'incrément en ordonnée vaut $q_{11} h k_1$, soit encore

$$k_2 = f(x_i, y_i) + p_1 h \frac{\partial}{\partial x_i} f(x_i, y_i) + q_{11} h k_1 \frac{\partial}{\partial y_i} f(x_i, y_i)$$

alors en utilisant la définition de k_1, on a pour y_{i+1}

$$\begin{aligned} y_{i+1} &= a_1 h k_1 + a_2 h k_1 + a_2 h^2 p_1 \frac{\partial}{\partial x_i} f(x_i, y_i) + a_2 q_{11} h^2 k_1 \frac{\partial}{\partial y_i} f(x_i, y_i) \\ &= (a1 + a_2) h k_1 + a_2 h^2 \left(p_1 \frac{\partial}{\partial x_i} f(x_i, y_i) + q_{11} k_1 \frac{\partial}{\partial y_i} f(x_i, y_i) \right) \\ &= (a1 + a_2) h f(x_i, y_i) + a_2 h^2 \left(p_1 \frac{\partial}{\partial x_i} + q_{11} f(x_i, y_i) \frac{\partial}{\partial y_i} \right) f(x_i, y_i) \quad (5.2) \end{aligned}$$

Si par ailleurs l'on écrit le développement en série de Taylor de y autour de x_0 on a

$$y_{i+1} = y_i + y_i' h + + \frac{h^2}{2!} y_i'' + o(h^3)$$

où on rappelle que $y_i \equiv y(x_i)$.
En notant que

$$y_i'' = \frac{d}{dx_i} f_i = \frac{\partial}{\partial x_i} f_i + y_i' \frac{\partial}{\partial y_i} f_i = \frac{\partial}{\partial x_i} f_i + f_i \frac{\partial}{\partial y_i} f_i$$

on obtient

$$y_{i+1} = y_i + h f_i + \frac{h^2}{2!} \left[\frac{\partial}{\partial x_i} + f_i \frac{\partial}{\partial y_i} \right] f_i \tag{5.3}$$

On peut maintenant faire l'analogie entre les équations (5.2) et (5.3), ce qui nous donne le système suivant

$$\begin{cases} a_1 + a_2 = 1 \\ a_2 p_1 = \frac{1}{2} \\ a_2 q_{11} = \frac{1}{2} \end{cases}$$

Ce système est sous-déterminé et on doit se donner l'une des variables, par exemple a_2 pour pouvoir le résoudre. Selon la valeur choisie, on peut définir diverses méthodes de Runge-Kutta d'ordre 2 :

Méthode de Ralston - $a_2 = \frac{2}{3}$

$$\begin{cases} y_{i+1} \approx y_i + \left(\frac{1}{3} k_1 + \frac{2}{3} k_2 \right) h \\ k_1 = f(x_i, y_i) \\ k_2 = f(x_i + \frac{3}{4} h, y_i + \frac{3}{4} k_1 h) \end{cases}$$

Méthode du point milieu - $a_2 = 1$

$$\begin{cases} y_{i+1} \approx y_i + k_2 h \\ k_1 = f(x_i, y_i) \\ k_2 = f(x_i + \frac{h}{2}, y_i + k_1 \frac{h}{2}) \end{cases}$$

5.3.3.2 Runge-Kutta d'ordre 4

Pour une méthode de Runge-Kutta d'ordre 4, on peut procéder de la même façon en posant cette fois-ci :

$$y_{i+1} = y_i + (a_1 k_1 + a_2 k_2 + a_3 k_3 + a_4 k_4) h + o(h^5)$$

En combinant cette expression au développement en série de Taylor d'ordre 4 de y au voisinage de x_i,

$$y_{i+1} = y_i + f(x_i, y_i)h + \frac{h^2}{2!}f'(x_i, y_i) + \frac{h^3}{3!}f''(x_i, y_i) + \frac{h^4}{4!}f^{(3)}(x_i, y_i) + o(h^5)$$

on obtient un système de 11 équations avec 13 inconnues.
Comme dans le cas des méthodes Runge-Kutta d'ordre 2, on va se donner les valeurs de deux coefficients et de là dériver tous les autres. Il existe deux solutions principales : celle de Runge et celle de Kutta.

Une autre façon de voir les choses est de repartir de l'équation (5.1) :

$$y(x_{i+1}) = y(x_i) + \int_{x_i}^{x_{i+1}} f(x, y(x))dx$$

et d'appliquer une quadrature de Newton-Cotes d'ordre plus élevé que la méthode des trapèzes pour résoudre l'intégrale du membre de droite.

Solution de Runge

Si on applique une quadrature de Simpson avec un pas d'intégration $h/2$ pour calculer l'intégrale $\int_{x_i}^{x_{i+1}} f(x, y)dx$ on a :

$$\int_{x_i}^{x_{i+1}} f(x, y(x))dx \approx \frac{h}{6}\left(f(x_i, y_i) + 4f(x_{i+\frac{1}{2}}, y_{i+\frac{1}{2}}) + f(x_{i+1}, y_{i+1})\right)$$

Ici il faut encore définir ce que l'on entend par l'indice $i + \frac{1}{2}$. En choisissant les notations $k_1 = f(x_i, y_i)$, $k_4 = f(x_{i+1}, y_{i+1})$ et $f(x_{i+\frac{1}{2}}, y_{i+\frac{1}{2}}) = \frac{k_2+k_3}{2}$, on obtient finalement

$$y_{i+1} \approx y_i + \frac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$$

k_1, k_2, k_3, k_4 peuvent être définis par analogie avec le développement de Taylor et on a

$$\begin{cases} k_1 = f(x_i, y_i) \\ k_2 = f(x_i + \frac{h}{2}, y_i + k_1\frac{h}{2}) \\ k_3 = f(x_i + \frac{h}{2}, y_i + k_2\frac{h}{2}) \\ k_4 = f(x_i + h, y_i + k_3 h) \end{cases}$$

Solution de Kutta

De la même manière, on peut trouver les coefficients de Runge en utilisant la quadrature de Simpson 3/8 pour l'estimation de l'intégrale du second membre dans l'expression (5.1) :

$$y_{i+1} \approx y_i + \frac{h}{8}(k_1 + 3k_2 + 3k_3 + k_4)$$

avec

$$\begin{cases} k_1 = f(x_i, y_i) \\ k_2 = f(x_i + \frac{h}{3}, y_i + k_1\frac{h}{3}) \\ k_3 = f(x_i + \frac{2}{3}h, y_i - k_1\frac{h}{3} + k_2 h) \\ k_4 = f(x_i + h, y_i + k_1 h - k_2 h + k_3 h) \end{cases}$$

5.3.3.3 Précision de la méthode de Runge-Kutta d'ordre 4

Théorème

Soit $y(x))$ la solution du problème aux conditions initiales suivant

$$\frac{dy}{dx} = f(x, y) \quad \text{avec} \quad y(x_0) = y_0 \quad \text{donn}$$

Si $y(x) \in C^5[x_0, b]$, et que $\{(x_k, y_k)\}_{k=0}^{M}$ est la séquence d'approximations générées par la méthode d'Euler sur l'intervalle d'intégration $[t_0 = a, b]$, alors pour chaque pas d'intégration de Runge-Kutta d'ordre 4 on peut définir

$$|e_k| = |y(x_k) - y_k| = o(h^4)$$

et

$$|\varepsilon_{k+1}| = |y(x_{k+1}) - y_k - hT_4(x_k, y_k)| = o(h^5)$$

où $h = (b - a)/M$ est le pas d'intégration et $T_4(x_k, y_k)$ est le polynôme de Taylor d'ordre 4.
L'erreur à la fin de l'intervalle d'intégration est appelée ***erreur finale globale*** et vaut :

$$E(y(b), h) = |y(b) - y_M)| = o(h^4)$$

5.4 Schémas multipas

5.4.1 Schéma implicites et schémas explicites

On dira qu'un schéma numérique est ***explicite*** lorsque la recherche de la solution $y(x_{n+1})$ de l'équation différentielle ordinaire évaluée au point x_{n+1} ne dépend que des valeurs que la fonction prend aux points précédents : $y(x_n)$ au temps x_n, etc... Ces schémas sont faciles à mettre en oeuvre et peu gourmands en termes de ressources de calcul. il sont cependant ***conditionnellement stables*** et peuvent devenir instable en fonction notamment du pas d'intégration choisi.

On dira qu'un schéma numérique est ***implicite*** lorsque la recherche de la solution $y(x_{n+1})$ de l'équation différentielle ordinaire évaluée au point x_{n+1} dépend des valeurs que la fonction prend aux points précédents : $y(x_n)$ au point x_n, etc... mais aussi de la valeur recherchée au point courant x_{n+1}.
Ces schémas sont plus difficiles à mettre en oeuvre et demandent plus de ressources car ils impliquent un processus itératif et des inversions matricielles. Ils ont cependant l'avantage d'être ***inconditionnellement stables***, ce qui les rend supérieurs aux schémas explicites.
Il existe également des schémas dits ***semi-implicites*** mais qui sont plutôt adaptés à la résolution d'équations aux dérivées partielles.

5.4.2 Schéma d'Euler retardé

On peut mettre la méthode d'Euler présentée en début de chapitre en oeuvre en écrivant la dérivée sous la forme de différences finies vers l'arrière plutôt que vers l'avant. On aura alors :

$$\frac{dy}{dt} = f(t, y)$$

qui peut aussi s'écrire

$$\frac{dy}{dt} \approx \frac{y(t_i) - y(t_{i-1})}{h}$$

soit encore

$$y(t_i) \approx h\frac{dy}{dt} + y(t_{i-1})$$

et finalement en décalant d'un cran vers l'avant

$$y(t_{i+1}) \approx y(t_i) + hf(t_{i+1}, y_{i+1})$$

Ce schéma à 1 pas peut ensuite être étendu à plusieurs pas, et on obtient ainsi les schémas d'Euler retardé d'ordre 2, 3 etc :

$$\begin{aligned}
\frac{dy(t)}{dt} &= \frac{3}{2\delta t}y_{i+1} - \frac{2}{\delta t}y_i + \frac{1}{2\delta t}y_{i-1} \quad \text{Schéma d}'\text{ordre 2} \\
\frac{dy(t)}{dt} &= \frac{11}{6\delta t}y_{i+1} - \frac{3}{\delta t}y_i + \frac{3}{2\delta t}y_{i-1} - \frac{1}{3\delta t}y_{i-2} \quad \text{Schéma d}'\text{ordre 3}
\end{aligned}$$

Ces schémas sont tous **implicites**.

5.4.3 Méthodes de prédicteur/correcteur

Les méthodes de prédicteur/correcteur sont des méthodes multipas dans lesquelles on utilise l'information ***de plusieurs pas précédents*** pour évaluer la fonction au

pas suivant. Ces méthodes ne sont pas auto-suffisantes en ce sens qu'il faut commencer par utiliser des méthodes à 1 pas pour construire l'information sur laquelle on va s'appuyer.
De façon générale dans une méthode multipas on aura

$$y_{n+1} = y_n + h\left(\beta_0 y'_{n+1} + \beta_1 y'_n + \beta_2 y'_{n-1} + \beta_3 y'_{n-2} + \cdots\right) \tag{5.4}$$

Lorsque $\beta_0 \neq 0$, on aura un schéma implicite.
L'ordre de la méthode multipas dépend du nombre de points *précédents* sur lesquels on a besoin de s'appuyer pour faire une évaluation de y au point x_{n+1}.
Les méthodes de prédicteur/correcteur font intervenir un ***prédicteur explicite*** qui sert à initier le processus itératif mis en oeuvre pour évaluer $y(x_{n+1})$. L'étape du prédicteur est donc une étape utilisant un schéma explicite qui sert à faire une première évaluation de $y(x_{n+1})$ qui est ensuite injectée dans le second membre de l'équation (5.4) pour réévaluer $y(x_{n+1})$. Cette nouvelle valeur sera ensuite réinjectée dans le membre de droite, et ainsi de suite, dans un processus itératif qui s'achève lorsque la différence entre membre de droite et membre de gauche devient suffisamment petite. La réévaluation de la fonction $y(x - n + 1)$ par ré-injection de la prédiction dans l'équation d'origine est l'étape du ***correcteur***.
La différence entre les valeurs de la fonction prédite et corrigée fournit l'information sur l'erreur de troncature locale qui peut être utilisée pour contrôler la précision et ajuster le pas de temps.

La méthode de prédicteur/correcteur la plus utilisée est la méthode d'Adams-Bashford-Moulton.

5.4.3.1 Méthode d'Adams-Bashford-Moulton

Dans cette méthode, on utilise le prédicteur d'Adams-Bashford et le correcteur d'Adams-Moulton de la façon suivante.

On repart de la formulation donnée par l'équation (5.1)

$$y(x_{k+1}) = y(x_k) + \int_{x_k}^{x_{k+1}} f(x, y(x))dx$$

Le prédicteur d'Adams-Bashford intègre entre x_k et x_{k+1} le polynôme d'interpolation de Lagrange de degré 4 pour évaluer la fonction $f(x, y(x)))$, qui est évalué aux points (x_{k-3}, f_{k-3}), (x_{k-2}, f_{k-2}),(x_{k-1}, f_{k-1}) et (x_k, f_k). On réalise donc une ***extrapolation*** à ce stade de la méthode. Le prédicteur ainsi construit est donné par

$$p_{k+1} = y_k + \frac{h}{24}\left(-9f_{k-3} + 37f_{k-2} - 59f_{k-1} + 55f_k\right) \tag{5.5}$$

Le correcteur d'Adams-Moulton est construit de la même façon mais consiste maintenant en une ***interpolation***. On construit ainsi un second polynôme d'interpolation

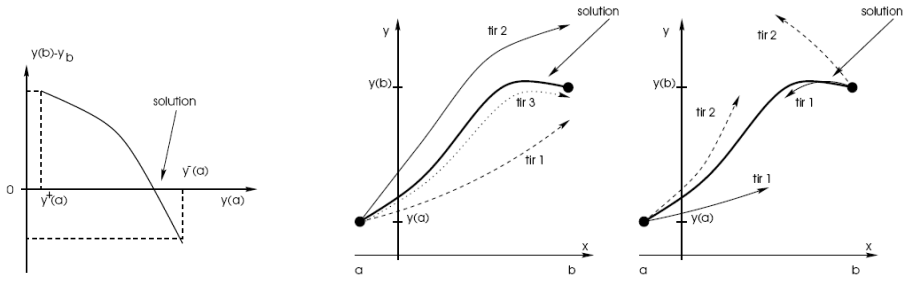


FIGURE 5.3 – Illustration de la méthode de tir.

de Lagrange pour $f(x, y(x)))$ qui est évalué aux noeuds (x_{k-2}, f_{k-2}),(x_{k-1}, f_{k-1}) , (x_k, f_k) et $(x_{k+1}, f_{k+1}) = (x_{k+1}, f(x_{k+1}, p_{k+1}))$:

$$y_{k+1} = y_k + \frac{h}{24}\left(f_{k-2} - 5f_{k-1} + 19f_k + 9f_{k+1}\right) \tag{5.6}$$

L'erreur de troncature locale associée au prédicteur et au correcteur est de l'ordre de $o(h^5)$.

On notera que le prédicteur et le correcteur peuvent être construits sur la base de polynômes d'interpolation de Lagrange de degrés inférieur ou supérieur à 4 qui est l'exemple choisi ici.

5.5 Équations différentielles ordinaires d'ordre > 1 - Problèmes aux conditions aux limites

Comme expliqué dans l'introduction de ce chapitre, une équation différentielle ordinaire d'ordre $N > 1$ peut toujours se décomposer en un système de N équations différentielles ordinaires d'ordre 1 auxquelles ont pourra appliquer l'un des méthodes décrites au paragraphe 5.3.

Lorsque le problème considéré est un problème aux conditions aux limites plutôt qu'aux conditions initiales, on pourra utiliser des méthodes spécifiques. Les méthodes décrites ci-après s'appliquent à des problèmes du type

$$\begin{cases} y''(x) = f(x, y(x), y'(x)) \quad \text{pour} \quad a \leq x \leq b \\ y(a) = \alpha \\ y(b) = \beta \end{cases} \tag{5.7}$$

5.5.1 Méthode de tir

Dans la méthode de tir il s'agit de transformer un problème d'équation différentielle ordinaire d'ordre > 1 aux conditions aux limites en un problème aux conditions

initiales.

Soit le problème aux conditions limites suivant :

$$\begin{aligned} A(x)\frac{d^2y}{dx^2} + B(x)\frac{dy}{dx} + C(x)y &= D(x) \\ y(0) &= y_0 \\ y(x=L) &= y_L \end{aligned} \tag{5.8}$$

On remplace ces conditions aux limites par les conditions initiales suivantes

$$\begin{cases} y(0) = y_0 \\ \left(\frac{dy}{dx}\right)_{x=0} = u \end{cases} \quad \text{où } u \text{ est arbitraire.}$$

On peut ainsi appliquer l'une des méthodes présentées précédemment (par exemple une méthode de Runge-Kutta) pour résoudre l'équation (5.10). La solution obtenue $y_u(x)$ donnera pour $x = L$, $y_u(L) = y_L(u)$. Cette solution est fonction de u, on va donc chercher u de sorte que $y_u(L) = y_L$.

On se ramène ainsi à un problème où il s'agit de trouver la racine de l'équation $y_u(L) - y_L = 0$. Le processus de résolution sera ***itératif***.

Pour résoudre l'équation (5.10), on se donne 2 valeurs arbitraires u_1 et u_2 qui mènent à $y_L(u_1)$ et $y_L(u_2)$. En faisant une interpolation linéaire entre ces deux solutions approchées, on obtient une troisième solution $y_L(u_3)$.

De façon plus générale, on déduit une estimation de u_{n+1} par interpolation linéaire des solutions approchées $y_L(u_{n-1})$ et $y_L(u_n)$ obtenues pour les valeurs de u_{n-1} et u_n :

$$u_{n+1} = \frac{(y_L - y_{n-1})u_n - (y_L - y)u_{n-1}}{y_n - y_{n-1}}$$

où on note y_n la solution approchée $y_L(u_n)$ à l'itération n.

On trouve la solution lorsque cette suite converge, c'est-à-dire lorsque u_{n+1} et u_n sont suffisamment proches l'un de l'autre.

5.5.2 Méthode des différences finies

On considère une équation différentielle ordinaire d'ordre 2 aux conditions aux limites sur l'intervalle $[a, b]$, soit :

$$\begin{aligned} y'' &= p(x)y'(x) + q(x)y(x) + r(x) \\ y(a) &= \alpha \\ y(b) &= \beta \end{aligned} \tag{5.9}$$

Au lieu de séparer cette équations en deux équations d'ordre 1, on va directement appliquer ce qui a été vu au chapitre 2 concernant la dérivation numérique.
Pour cela on considère N+1 points équirépartis sur l'intervalle $[a,b]$, $x_0 = a, x_1, \cdots, x_N = b$, avec $h = (b-a)/N$ et $x_j = a + hj$ pour j entier compris entre 0 et N. En utilisant les formules de différences finies centrées à deux points pour les dérivées première et seconde

$$y'(x_j) = \frac{y(x_{j+1}) - y(x_{j-1})}{2h} + o(h^2)$$

et

$$y''(x_j) = \frac{y(x_{j+1}) - 2y(x_j) + y(x_{j-1})}{h^2} + o(h^2)$$

on peut réécrire l'équation différentielle comme suit :

$$\frac{y(x_{j+1}) - 2y(x_j) + y(x_{j-1})}{h^2} + o(h^2) = p(x_j)\left(\frac{y(x_{j+1}) - y(x_{j-1})}{2h} + o(h^2)\right) + q(x_j)y(x_j) + r(x_j)$$

On peut se débarrasser des termes $o(h^2)$, et en adoptant la notation $y(x_j) = y_j$ on arrive à

$$\frac{y_{j+1} - 2y_j + y_{j-1}}{h^2} = p_j\left(\frac{y_{j+1} - y_{j-1}}{2h}\right) + q_j y_j + r_j$$

En rassemblant les termes de même indice, et en écrivant cette équation pour chaque point de l'intervalle $[a,b]$, on arrive finalement au système d'équations suivant :

$$\left(-\frac{h}{2}p_j - 1\right) y_{j-1} + \left(2 + h^2 q_j\right) y_j + \left(\frac{h}{2}p_j - 1\right) y_{j+1} = -h^2 r_j$$

que l'on peut mettre sous forme matricielle en prenant bien en compte le fait que $y_0 = \alpha$ et $y_1 = \beta$:

$$\begin{pmatrix} 2+h^2q_1 & \frac{h}{2}p_1 - 1 & 0 & \cdots & 0 \\ -\frac{h}{2}p_2 - 1 & 2+h^2q_2 & \frac{h}{2}p_2 - 1 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & -\frac{h}{2}p_j - 1 & 2+h^2q_j & \frac{h}{2}p_j - 1 & \\ 0 & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & -\frac{h}{2}p_{N-2} - 1 & 2+h^2q_{N-2} & \frac{h}{2}p_{N-2} - 1 \\ 0 & \cdots & 0 & -\frac{h}{2}p_j N - 1 - 1 & 2+h^2q_{N-1} \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_{N-2} \\ y_{N-1} \end{pmatrix}$$

$$= \begin{pmatrix} -h^2 r_1 + e_0 \\ -h^2 r_2 \\ \vdots \\ -h^2 r_j \\ \vdots \\ -h^2 r_{N-1} + e_N \end{pmatrix}$$

où $e_0 = \left(\frac{h}{2}p_1 + 1\right)\alpha$ et $e_N = \left(-\frac{h}{2}p_{N-1} + 1\right)\beta$

Pour résoudre le problème il suffit alors de procéder à une inversion matricielle :

$$A \cdot Y = B \qquad \Rightarrow \qquad Y = A^{-1} \cdot B$$

Chapitre 6

Résolution d'équations aux dérivées partielles

Sommaire

6.1 Introduction

Une équation aux dérivées partielles (EDP) est une équation différentielle impliquant des dérivées par rapport à plusieurs variables indépendantes. C'est en ce sens que ce type d'équations se différencie des équations différentielles ordinaires vues au chapitre précédent.

On se concentre sur les équations dites quasi-linéaires qui sont de la forme :

$$A\Phi_{xx} + B\Phi_{xy} + C\Phi_{yy} = f(x, y, \Phi, \Phi_x, \Phi_y) \tag{6.1}$$

où on a adopté la notation $\Phi_x \equiv \frac{\partial}{\partial x}$ et $\Phi_{xy} \equiv \frac{\partial^2}{\partial x \partial y}$.

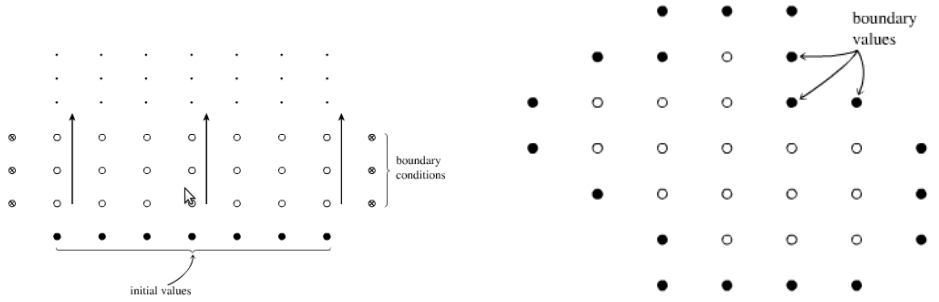


FIGURE 6.1 – Différence entre problèmes aux conditions initiales (à gauche) et problèmes aux conditions de bord (à droite). Dans un problème aux conditions initiales, les valeurs initiales sont données sur une tranche temporelle, et on avance la solution en temps en calculant les lignes successives de points ouverts dans la direction indiquée par les flèches. Les conditions de bord à gauche et à droite symbolisées par $\otimes$ doivent aussi être données, mais on n'a besoin que d'une paire par tranche temporelle. Dans le cas de problèmes à conditions de bord, les conditions limites sont spécifiées sur le pourtour de la grille d'intégration, et on utilise un processus itératif pour trouver les valeurs de tous les points ouverts à l'intérieur de ce domaine. *Extrait de Numerical Recipes.*

En assimilant les opérateurs de dérivation à des variables, on peut faire l'analogie entre l'équation (6.1) et un polynôme du second degré, et c'est cette analogie qui est utilisée pour classer les différents types d'équations aux dérivées partielles.
On compte 3 types distincts d'équations aux dérivées partielles quasi-linéaires qui sont définis comme suit :

- $B^2 - 4AC < 0$ ***équation elliptique***
  équation de Laplace, équation de Poisson

- $B^2 - 4AC = 0$ ***équation parabolique***
  équation de la chaleur, équation de Schrödinger

- $B^2 - 4AC > 0$ ***équation hyperbolique***
  équation de propagation des ondes, équation d'advection

On peut également classer les équations aux dérivées partielles en deux types de familles

- **Problèmes aux valeurs initiales / problèmes de Cauchy**
  C'est le cas des équations paraboliques et hyperboliques pour lesquels l'une des variables indépendantes est le temps

$$\frac{\partial^2 u}{\partial t^2} = \nu^2 \frac{\partial^2 u}{\partial x^2} \quad \text{éq. propagation d}'\text{onde}$$

$$\frac{\partial u}{\partial t} = -\frac{\partial}{\partial x}\left(D\frac{\partial u}{\partial x}\right) \quad \text{éq. de diffusion}$$

– **Problèmes aux conditions de bord**
C'est le cas des équations elliptiques qui concernent des problèmes indépendants du temps

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = \rho(x, y) \quad \text{éq. de Poisson}$$

Pour résoudre les EDP quasi-linéaires, on fait appel à deux types de méthodes de résolution :
– les méthodes des différences finies (problèmes de Cauchy)
– les méthodes spectrales (problèmes aux conditions de bord)
Dans un premier temps on ne s'intéresse qu'aux problèmes de Cauchy.

6.1.1 Discrétisation des EDP

Dans le cadre d'une résolution d'EDP par une méthode de différences finies, on va utiliser une discrétisation directe des équations en écrivant chacune des dérivées sous la forme de différentielles (voir chapitre 2).

Pour une fonction $u(x,t)$ et un maillage tel que $\Delta x = x_{j+1} - x_j$ et $\Delta t = t_{n+1} - tn$, on a les relations suivantes :

– Dérivées premières :

$$\left(\frac{\partial u}{\partial x}\right)_{j,n} = \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} \tag{6.2}$$

$$\left(\frac{\partial u}{\partial t}\right)_{j,n} = \frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} \tag{6.3}$$

– Dérivées secondes :

$$\left(\frac{\partial^2 u}{\partial x^2}\right)_{j,n} = \frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta x^2} \tag{6.4}$$

$$\left(\frac{\partial^2 u}{\partial t^2}\right)_{j,n} = \frac{u_j^{n+1} - 2u_j^n + u_j^{n-1}}{\Delta t^2} \tag{6.5}$$

– Dérivées croisées :

$$\left(\frac{\partial^2 u}{\partial x \partial t}\right)_{j,n} = \left(\frac{\partial^2 u}{\partial t \partial x}\right)_{j,n} = \frac{u_{j+1}^{n+1} - u_{j+1}^{n-1} - u_{j-1}^{n+1} + u_{j-1}^{n-1}}{4\Delta x \Delta t} \tag{6.6}$$

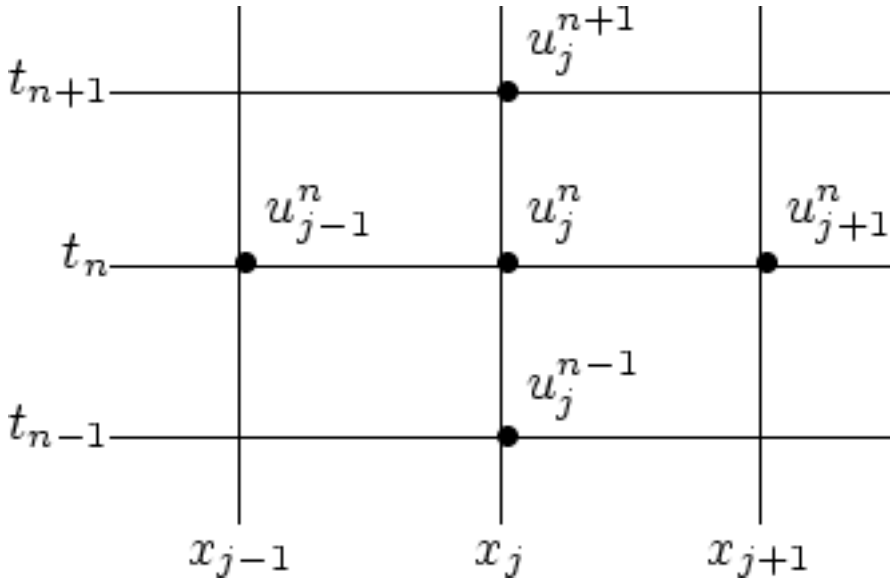


FIGURE 6.2 – Maillage à deux dimensions pour les différences finies

6.2 Analyse de stabilité de Von Neumann

Pour chacune des méthodes de différences finies que nous détaillons ci-après, on peut réaliser une analyse de stabilité afin de déterminer la qualité de la méthode et sa propension (ou pas) à propager des erreurs. Pour cela on applique une analyse de stabilité de Von Neumann qui consiste en ceci :

Hypothèse Les coefficients des équations différentielles varient très lentement. Ceci nous amène à supposer que sur les intervalles d'intégration considérés, les coefficients sont supposés constants par rapport à chacune des variables indépendantes.

Dans ce cas, les modes propres des équations sont tous de la forme

$$u_j^n = \xi^n e^{ikj\Delta x}$$

où $x_j = j\Delta x$, $k \in \mathbb{R}$ est le nombre d'onde spatial et $\xi = \xi(k) \in \mathbb{C}$ est le facteur d'amplification.

La dépendance en temps d'un mode propre donné est définie par les puissances entières successives du nombre ξ, $\xi^{n+1} = g(k)\xi^n$.

On dira que le schéma de discrétisation d'une EDP est instable s'il existe des valeurs de k pour lesquelles

$$\|g(k)\| > 1$$

6.3 Équations hyperboliques

L'équation d'advection est le prototype des équations hyperboliques. Cette équation décrit le fait qu'une quantité u est transportée par un flot fluide à la vitesse υ. A une dimension, et pour une vitesse constante, cette équation s'écrit

$$\frac{\partial u}{\partial t} = -\upsilon \frac{\partial u}{\partial x} \tag{6.7}$$

La solution de cette équation est une onde qui se propage dans la direction des x positifs $u = f(x - \upsilon t)$.

Pour résoudre numériquement cette équation, on choisit dans un premier temps une discrétisation équirépartie en temps et en espace :

$$\begin{cases} x_j = x_0 + j\Delta x, & j = 0, 1, ...J \\ t_n = x_0 + n\Delta t, & n = 0, 1, ...N \end{cases}$$

6.3.1 Schéma FTCS

Définition

Ce premier schéma est un schéma ***explicite en temps*** : la dérivée temporelle est approximée par une différence finie vers l'avant :

$$\left.\frac{\partial u}{\partial t}\right|_{j,n} = \frac{u_j^{n+1} - u_j^n}{\Delta t} + o(\Delta t)$$

Pour la dérivée spatiale on utilise par contre un schéma centré d'ordre supérieur ne faisant intervenir que des valeurs au pas de temps n :

$$\left.\frac{\partial u}{\partial x}\right|_{j,n} = \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} + o((\Delta x)^2)$$

En adoptant ces discrétisations, l'équation d'advection (6.7) s'écrit :

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -\upsilon \ \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} \tag{6.8}$$

Cette méthode met en oeuvre un ***schéma explicite en temps***, et elle est ***instable***.

Stabilité

Pour établir la stabilité du schéma FTCS, on applique l'analyse de Von Neumann à l'équation d'advection discrétisée. Cela revient à remplacer dans l'équation (6.8), les u_j^n par l'expression des modes propres $u_j^n = \xi^n e^{ikj\Delta x}$.

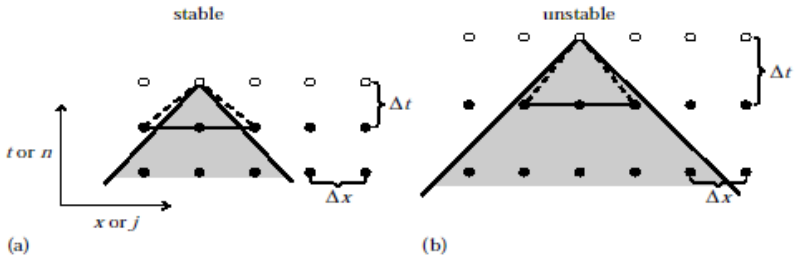


FIGURE 6.3 – Illustration de la condition de stabilité CFL.

$$\begin{aligned}
\frac{u_j^{n+1} - u_j^n}{\Delta t} &= -\upsilon \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} \\
\frac{\xi^{n+1} e^{ikj\Delta x} - \xi^n e^{ikj\Delta x}}{\Delta t} &= -\upsilon \frac{\xi^n e^{ik(j+1)\Delta x} - \xi^n e^{ik(j-1)\Delta x}}{2\Delta x} \\
\xi^n e^{ikj\Delta x} \left(g(k) - 1\right) &= -\upsilon \frac{\Delta t}{2\Delta x} e^{ikj\Delta x} \xi^n \left(e^{ik\Delta x} - e^{-ik\Delta x}\right) \\
g(k) &= 1 - \upsilon \frac{\Delta t}{2\Delta x} \left(e^{ik\Delta x} - e^{-ik\Delta x}\right) \\
g(k) &= 1 - i\upsilon \frac{\Delta t}{\Delta x} sin(k\Delta x)
\end{aligned}$$

On a bien $\|g(k)\| > 1$ pour toutes les valeurs de k, ce qui implique que le schéma est ***inconditionnellement instable***.

6.3.2 Schéma de Lax

Le schéma de Lax est une amélioration de la stabilité du schéma FTCS. Il s'obtient en remplaçant u_j^n par la moyenne $\frac{1}{2}\left(u_{j+1}^n + u_{j-1}^n\right)$ dans l'équation (6.8) :

$$u_j^{n+1} = \frac{1}{2}\left(u_{j+1}^n + u_{j-1}^n\right) - \upsilon \frac{\Delta t}{2\Delta x}\left(u_{j+1}^n + u_{j-1}^n\right) \tag{6.9}$$

Comme pour le schéma FTCS, on peut faire une analyse de stabilité de Von Neumann, qui donne le résultat suivant pour la dépendance en temps du facteur d'amplification

$$g(k) = cos(\Delta x) - i\upsilon \frac{\Delta t}{\Delta x} sin(k\Delta x)$$

Contrairement au schéma FTCS, on voit ici que ce schéma peut être stable sous certaines conditions. En fait, la condition de stabilité $\|g(k)\| \leq 1$ est assurée ici lorsque $\upsilon \frac{\Delta t}{\Delta x} \leq 1$.

Cette condition de stabilité est appelée condition de Courant-Friedrichs-Lewy, ou condition CFL. Une représentation graphique de la condition CFL est donnée en

figure 6.3. Si le pas de temps choisi est tel que la vitesse d'accroissement numérique $\Delta x/\Delta t$ est supérieure à la vitesse du flot fluide v, alors le schéma devient instable car on utilise de l'information provenant d'une zone où elle n'est pas connue (délimitée par le triangle en pointillés sur la figure).
On peut se demander en quoi le remplacement de u_j^n par la valeur moyenne $\frac{1}{2}\left(u_{j+1}^n + u_{j-1}^n\right)$ permet de passer d'un ***schéma inconditionnellement instable*** (FTCS) à un ***schéma conditionnellement stable*** (Lax).
Pour cela, on réécrit l'équation (6.9) sous la même forme que l'équation (6.8) :

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -v\ \frac{u_{j+1}^n - u_{j-1}^n}{2\Delta x} + \frac{1}{2}\left(\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{\Delta t}\right) \tag{6.10}$$

Cette équation est en fait la représentation par le schéma FTCS de l'équation suivante :

$$\frac{\partial u}{\partial t} = -v\ \frac{\partial u}{\partial x} + \frac{(\Delta x)^2}{\Delta t}\nabla^2 u \tag{6.11}$$

L'amélioration apportée par le schéma de Lax consiste donc en réalité à introduire un terme du second ordre en espace ($\nabla^2 u$) qui est un terme de ***diffusion ou de dissipation numérique***.

6.3.3 Schéma explicite excentré d'ordre 1 dit schéma "upwind"

Le schéma "upwind" permet de stabiliser le schéma explicite FTCS en utilisant une discrétisation décentrée du second membre de l'équation d'advection (6.7). Ce terme, $v\frac{\partial u}{\partial x}$ représente l'advection de u par un champ de vitesse v, et si $v > 0$, la matière est advectée de gauche à droite. On a donc bien un cas de figure qui se prête par nature à une description excentrée de la dérivée.

si $v > 0$, la dérivée de u ne dépend que des valeurs en $i-1$ et en i :

$$v\frac{\partial u}{\partial x} \approx v_i \frac{u_i - u_{i-1}}{\Delta x}$$

si $v < 0$, la dérivée de u ne dépend que des valeurs en $i+1$ et en i :

$$v\frac{\partial u}{\partial x} \approx v_i \frac{u_{i+1} - u_i}{\Delta x}$$

On assure donc une meilleure modélisation des propriétés du transport advectif avec la discrétisation suivante :

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = -v_j^n \begin{cases} \frac{u_j^n - u_{j-1}^n}{\Delta x}, & v_j^n > 0 \\ \\ \frac{u_{j+1}^n - u_j^n}{\Delta x}, & v_j^n < 0 \end{cases} \tag{6.12}$$

Le schéma upwind est d'ordre inférieur au schéma de Lax, mais présente l'avantage de conserver la forme de la fonction u au passage d'une discontinuité (par exemple

en présence d'un choc).

Stabilité de Von Neumann

Le critère de stabilité de ce schéma est donné en écrivant u sous la forme des modes propres $u_j^n = \xi^n e^{ikj\Delta x}$, avec $\xi^n = g(k)\xi^{n-1}$.
On obtient

$$g(k) = 1 - \left|\frac{v\Delta t}{\Delta x}\right| (1 - cos(k\Delta x)) - i\frac{v\Delta t}{\Delta x} sin(k\Delta x)$$

Le schéma upwind est stable sous pour la condition CFL, c-à-d que $\|g\| < 1$ si $\frac{v\Delta t}{\Delta x} \leq 1$

6.3.4 Schéma saute-mouton ou "leapfrog"

Le schéma upwind présenté ci-avant comporte une forte dissipation numérique, qui entraîne, selon le choix de la condition CFL, une dissipation d'amplitude de la solution (nulle lorsque $\frac{v\Delta t}{\Delta x} = 1$). Afin de conserver l'amplitude de la solution quelle que soit la condition CFL adoptée, on peut passer à un schéma de discrétisation centré (donc d'ordre 2) en temps et en espace.

Ainsi l'équation d'advection (6.7) peut se discrétiser de la façon suivante :

$$\frac{u_j^{n+1} - u_j^{n-1}}{2\Delta t} = -v_j^n \frac{u_j^{n+1} - u_j^{n-1}}{2\Delta x} \tag{6.13}$$

Ce schéma est explicite en temps mais ***implicite*** dans la dimension spatiale.

En réécrivant l'équation (6.13) sous la forme

$$u_j^{n+1} = u_j^{n-1} - \frac{v\Delta t}{\Delta x}\left(u_j^{n+1} - u_j^{n-1}\right)$$

on voit clairement apparaître une décorrélation dans le temps entre points d'indice temporel pair et points d'indice temporelle impair. Tout se passe comme si on divisait la grille de discrétisation en deux sous-grilles décorrélées.
Afin de conserver une cohérence entre l'évolution des deux sous-grilles, il va là aussi falloir se plier à une condition de type CFL comme nous le montre l'analyse de stabilité de Von Neumann pour ce schéma saute-mouton.

Stabilité de Von Neumann

$$g(k)^2 - 1 = 2ig(k)\frac{v\Delta t}{\Delta x} sin(k\Delta x)$$

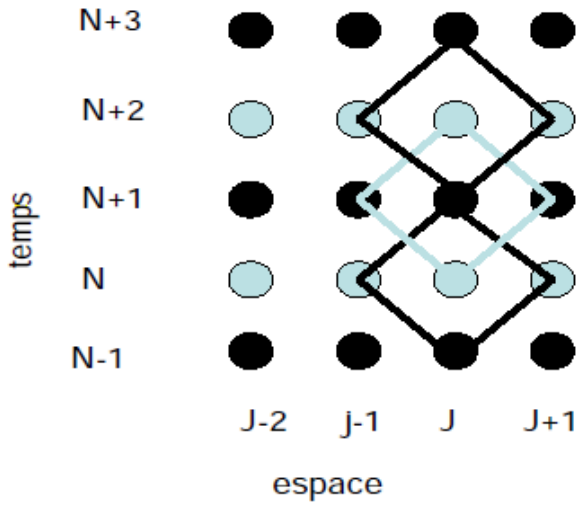


FIGURE 6.4 – Illustration du schéma saute-mouton et de la déconnexion entre points de grille pairs et points de grille impairs

Soit encore

$$g(k) = -i\frac{\upsilon\Delta t}{\Delta x}sin(k\Delta x) \pm \sqrt{1 - \left(\frac{\upsilon\Delta t}{\Delta x}sin(k\Delta x)\right)^2}$$

Contrairement au schéma upwind , ce schéma conserve l'amplitude de u dans la mesure où la condition CFL est respectée.

6.3.5 Schéma de Crank-Nicholson

Comme déjà mentionné dans le chapitre sur les équations différentielles ordinaires, une des façons d'assurer la stabilité d'un schéma est d'adopter une discrétisation implicite en temps.
C'est ce que l'on fait avec le ***schéma de Crank-Nichoslon***, qui est ***d'ordre 2 en temps et en espace***.

Le schéma de Crank-Nicholson est obtenu en prenant des moyennes temporelles des dérivées spatiales :

$$\frac{\partial u}{\partial x} = \frac{1}{2}\left[\left.\frac{\partial u}{\partial x}\right|_{n+1} + \left.\frac{\partial u}{\partial x}\right|_{n}\right] = \frac{1}{2}\frac{\left(u_{j+1}^{n+1} - u_{j-1}^{n+1}\right) + \left(u_{j+1}^{n} - u_{j-1}^{n}\right)}{2\Delta x}$$

En remplaçant cette expression dans l'équation (6.7), on obtient

$$u_j^{n+1} = u_j^n - \frac{\upsilon\Delta t}{4\Delta x}\left[\left(u_{j+1}^{n+1} - u_{j-1}^{n+1}\right) + \left(u_{j+1}^{n} - u_{j-1}^{n}\right)\right] \tag{6.14}$$

En adoptant la notation $\sigma = \frac{v\Delta t}{\Delta x}$ et en regroupant les termes de même rang en temps on obtient :

$$u_j^{n+1} + \frac{\sigma}{4}\left(u_{j+1}^{n+1} - u_{j-1}^{n+1}\right) = u_j^n - \frac{\sigma}{4}\left(u_{j+1}^n - u_{j-1}^n\right)$$

Lorsqu'on cherche l'état du système , c'est-à-dire la valeur de u à l'instant $n+1$ sur tout le domaine spatial, on se retrouve avec un système de N équations, si on a N points de discrétisation spatiale qui peut se mettre sous forme matricielle. La fonction u à l'instant $n+1$ est obtenue par inversion matricielle.

$$u^{n+1}A = Bu^n \rightarrow u^{n+1} = A^{-1}Bu^n$$

avec

$$\mathrm{A} = \begin{pmatrix} 1 & \sigma/4 & 0 & 0 & \cdots & 0 \\ -\sigma/4 & 1 & \sigma/4 & 0 & \cdots & 0 \\ 0 & -\sigma/4 & 1 & \sigma/4 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & -\sigma/4 & 1 & \sigma/4 \\ 0 & 0 & 0 & 0 & -\sigma/4 & 1 \end{pmatrix}$$

$$\mathrm{B} = \begin{pmatrix} 1 & -\sigma/4 & 0 & 0 & \cdots & 0 \\ \sigma/4 & 1 & -\sigma/4 & 0 & \cdots & 0 \\ 0 & \sigma/4 & 1 & -\sigma/4 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \sigma/4 & 1 & -\sigma/4 \\ 0 & 0 & 0 & 0 & \sigma/4 & 1 \end{pmatrix}$$

Les conditions aux limites sont des conditions de type Dirichlet et sont données par $u_{j=0} = 0$ et $u_{j=jmax} = 0$.

Stabilité de Von Neumann

L'analyse de stabilité de Von Neumann donne

$$g(k) = \frac{1 - \sigma/(2isin(\Delta x))}{1 + \sigma/(2isin(\Delta x))}$$

Ici on a $\|g\| = 1$ pour tout nombre d'onde k : ***ce schéma est inconditionnellement stable***. Il est également précis car d'ordre 2, cependant, il peut produire des oscillations en présence de discontinuités (n'est pas adapté au traitement des chocs).

6.4 Équations paraboliques

L'équation de diffusion est le prototype des équations aux dérivées partielles dites paraboliques

$$\frac{\partial u}{\partial t} = -\frac{\partial}{\partial x}\left(D\frac{\partial u}{\partial x}\right) \tag{6.15}$$

Lorsque le coefficient de diffusion $D(x,t) = cte$, cette équation se ramène à une équation de conservation du flux en posant $F = D\frac{\partial u}{\partial x}$, et l'équation de diffusion à 1D s'écrit alors

$$\frac{\partial u}{\partial t} - D\frac{\partial^2 u}{\partial x^2} = 0 \tag{6.16}$$

6.4.1 Schéma FTCS

On peut appliquer aux équations paraboliques le même type de schémas que ceux présentés pour les équations hyperboliques.
Le premier de ces schéma est le schéma FTCS.

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = D\frac{u_{j+1}^n - 2u_j^n + u_{j-1}^n}{(\Delta x)^2} \tag{6.17}$$

Le fait que le second membre est une dérivée seconde change le résultat de l'analyse de stabilité de Von Neumann.

Stabilité de Von Neumann

Le taux d'accroissement de l'erreur au sens de Von Neumann vaut ici

$$g(k) = 1 - \frac{4D\Delta t}{(\Delta x)^2} sin^2\left(\frac{k\Delta x}{2}\right)$$

Contrairement aux cas des équations hyperboliques, le schéma FTCS appliqué à une équation parabolique est stable si $\|g\| \leq 1$, ce qui est le cas ici lorsque

$$2\frac{D\Delta t}{(\Delta x)^2} \leq 1$$

.
Cette condition est similaire à une condition CFL.
Elle traduit le fait que pour avoir un schéma stable il faut que ***le pas de temps d'intégration soit inférieur au temps caractéristique de diffusion sur une cellule de largeur*** Δx***,***

$$\tau_{diff} \propto \frac{(\Delta x)^2}{D}$$

6.4.2 Schéma implicite en temps

On voit que le schéma FTCS nous donne déjà la possibilité d'avoir une stabilité numérique, mais il demande un pas de temps bridé à la taille de la résolution spatiale utilisée, ce qui peut être un facteur limitant. On cherche donc un nouveau schéma qui nous permet d'utiliser des pas de temps plus grands pour suivre l'évolution de u aux grandes échelles spatiales sans pour autant perdre l'information aux petites échelles.

Une façon de procéder est de prendre un schéma implicite en temps pour la discrétisation spatiale, c-à-d que dans le second membre de l'équation (6.16), les éléments de u sont évalués non plus au pas de temps t_n, mais en t_{n+1} :

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = D\ \frac{u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}}{(\Delta x)^2} \tag{6.18}$$

Comme à chaque fois pour les schémas implicites en temps, trouver la solution de l'équation à l'instant t_{n+1} requiert de résoudre l'équation pour tous les points de grille spatiaux en même temps.
On doit donc résoudre un système d'équations linéaires couplées du type

$$-\alpha u_{j-1}^{n+1} + u_j^{n+1}\,(1+2\alpha) - \alpha u_{j+1}^{n+1} = u_j^n \quad \forall j = 1, 2, ..., J-1$$

où $J+1$ est le nombre de points de la grille dans la direction spatiale et

$$\alpha = \frac{D\Delta t}{(\Delta x)^2}$$

Les conditions limites sont des conditions de type Dirichlet ou Neumann, c-à-d correspondent à des conditions de bord pour la fonction elle-même (CL de Dirichlet) ou pour les dérivées de la fonction (CL de Neumann).
La résolution du système passe par l'inversion de la matrice tridiagonale du membre de gauche.

Stabilité de Von Neumann

On vérifie la stabilité du schéma implicite en temps en remplaçant les u_j^n par $u_j^n = \xi^n e^{ikj\Delta x}$, avec $\xi^n = g(k)\xi^{n-1}$:

$$g(k) = \frac{1}{1 + 4\alpha sin^2\left(\frac{k\Delta x}{2}\right)}$$

Ce taux d'accroissement est de module toujours inférieur à 1 quel que soit le pas de temps Δt choisi.
Le schéma implicite en temps est inconditionnellement stable.

Les détails de l'évolution des petites échelles spatiales seront imprécis pour les grandes valeurs de Δt, mais on obtiendra néanmoins la bonne solution d'équilibre pour ces échelles. Ceci est l'une des caractéristiques des schémas implicites.

6.4.3 Schéma de Crank-Nicholson

On applique à l'équation de diffusion (6.16) la même méthode que pour l'équation d'advection, c-à-d qu'on utilise le schéma des différences finies centrées pour la dérivée seconde spatiale, et on fait une moyenne sur les instants t_{n+1} et t_n pour cette dérivée seconde :

$$\frac{u_j^{n+1} - u_j^n}{\Delta t} = \frac{D}{2} \frac{\left(u_{j+1}^{n+1} - 2u_j^{n+1} + u_{j-1}^{n+1}\right) + \left(u_{j+1}^n - 2u_j^n + u_{j-1}^n\right)}{(\Delta x)^2} \tag{6.19}$$

Tout se passe comme si on faisait l'évaluation de la dérivée au pas de temps $t_{n+\frac{1}{2}}$. Cette discrétisation est ***implicite en temps***, et donc à chaque pas de temps, on doit résoudre un système d'équations caractérisé par :

$$-\alpha\, u_{j-1}^{n+1} + u_j^{n+1}\,(1+2\alpha) - \alpha\, u_{j+1}^{n+1} = \alpha\, u_{j-1}^n + u_j^n\,(1+2\alpha) + \alpha\, u_{j+1}^n \quad \forall j = 1, 2, ..., J-1$$

où

$$\alpha = \frac{D\Delta t}{2(\Delta x)^2}$$

La fonction u à l'instant t_{n+1} sur l'ensemble du domaine spatial est donnée par la résolution d'un système matriciel de la forme

$$U^{n+1} = A^{-1} \cdot B \cdot U^n$$

Stabilité de Von Neumann

L'analyse de stabilité de Von Neumann donne l'expression suivante pour le taux d'accroissement de l'erreur :

$$g(k) = \frac{1 - 2\alpha sin^2\left(\frac{k\Delta x}{2}\right)}{1 + 2\alpha sin^2\left(\frac{k\Delta x}{2}\right)}$$

Le module de $g(k)$ est inférieur ou égal à 1 pour tout k : comme pour le cas des équations hyperboliques, ***le schéma de Crank-Nicholson appliqué aux équations paraboliques est inconditionnellement stable***.

6.5 Équations elliptiques - Problèmes indépendants du temps

Les problèmes impliquant des équations aux dérivées partielles indépendantes du temps avec des conditions de bord associées sont décrits par des équations elliptiques.

C'est le cas de l'équation de Poisson, que l'on peut utiliser comme prototype.
Pour un potentiel électrique U et une distribution spatiale des charges électriques $\rho(x, y, z)$ l'équation de Poisson s'écrit

$$\nabla^2 U = \frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} + \frac{\partial^2 U}{\partial z^2} = -\frac{\rho(x, y, z)}{\varepsilon_0} \tag{6.20}$$

Dans ce problème, on doit résoudre un champ dans une portion donnée de l'espace , et non plus une évolution de ce champ au cours du temps. On peut résoudre ce problème soit en utilisant la méthode directe (déconseillée), soit en employant une méthode dite ***spectrale***.

6.5.1 Différences finies

On se place dans le cas d'un champ engendré par une charge électrique à 2D :

$$\frac{\partial^2 U}{\partial x^2} + \frac{\partial^2 U}{\partial y^2} = -\frac{\rho(x, y)}{\varepsilon_0}$$

On discrétise avec le schéma des différences finies centrées :

$$\frac{u_{j-1,l} - 2u_{j,l} + u_{j+1,l}}{\partial x^2} + \frac{u_{j,l-1} - 2u_{j,l} + u_{j,l+1}}{\partial y^2} = -\frac{\rho_{j,l}}{\varepsilon_0}$$

Dans le cas d'une grille symétrique en x et y, avec $\Delta l = \varepsilon_0 jx\Delta x = \varepsilon_0 ly\Delta y$, cette équation devient

$$\frac{1}{(\Delta l)^2}\left(u_{j-1,l} - 2u_{j,l} + u_{j+1,l} + u_{j,l-1} - 2u_{j,l} + u_{j,l+1}\right) = -\rho_{j,l}$$

Soit encore :

$$\frac{1}{(\Delta l)^2}\left(u_{j-1,l} + u_{j+1,l} + u_{j,l-1} + u_{j,l+1} - 4u_{j,l}\right) = -\rho_{j,l} \tag{6.21}$$

Pour résoudre l'équation (6.21) par la méthode directe des différences finies, on doit inverser une matrice tridiagonale avec des franges (montrée en figure 6.5), qui comporte beaucoup d'éléments nuls. L'inversion de ce type de matrices est très lourde et coûteuse en temps de calcul.
On préférera résoudre l'équation (6.21) en appliquant une méthode spectrale.

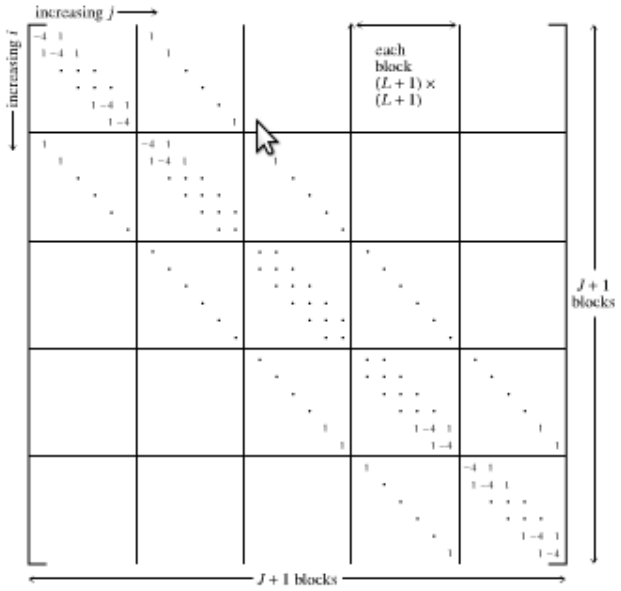


FIGURE 6.5 – Structure de la matrice obtenue en discrétisant une équation elliptique d'ordre 2 suivant le schéma des différences finies centrées. Les éléments matriciels qui ne sont pas montrés sont nuls. La matrice comporte des blocs diagonaux qui sont eux-mêmes tridiagonaux, et des blocs super et sous-diagonaux qui sont eux-mêmes diagonaux. Cette forme particulière de matrices est appelée " matrice tridiagonale avec des franges". *Tiré de Numerical Recipes.*

6.5.2 Méthode spectrale

On parlera de méthode spectrale pour la résolution des équations elliptiques lorsqu'on utilise une décomposition en série de Fourrier des variables physiques u et ρ.

La méthode spectrale consiste à utiliser les propriétés de la transformée de Fourier en particulier en ce qui concerne le produit de convolution.
En effet, on peut remarquer que l'équation (6.21) est un produit de convolution du type

$$(f * g)(m) = \sum_n f(n)g(m-n)$$

Si on considère les transformées de Fourier inverse discrètes pour u et ρ

$$u_{j,l} = \frac{1}{JL} \sum_{m=0}^{J-1} \sum_{n=0}^{L-1} \hat{u}_{m,n} e^{-(2\pi ijm)/J} e^{-(2\pi iln)/L}$$

et

$$\rho_{j,l} = \frac{1}{JL} \sum_{m=0}^{J-1} \sum_{n=0}^{L-1} \hat{\rho}_{m,n} e^{-(2\pi ijm)/J} e^{-(2\pi iln)/L}$$

on obtient en remplaçant dans (6.21) :

$$\hat{u}_{m,n} \left(e^{(2\pi ijm)/J} + e^{-(2\pi ijm)/J} + e^{(2\pi iln)/L} + e^{-(2\pi iln)/L} - 4 \right) = \hat{\rho}_{m,n} (\Delta l)^2$$

soit encore

$$\hat{u}_{m,n} = \frac{\hat{\rho}_{m,n} (\Delta l)^2}{2 \left(cos \left(\frac{2\pi m}{J} \right) + cos \left(\frac{2\pi n}{L} \right) - 2 \right)} \tag{6.22}$$

Pour résoudre cette équation, on utilisera des techniques de Transformée de Fourier Rapide (FFT) :

1. On calcule la transformée de Fourier de ρ :

$$\hat{\rho}_{m,n} = \sum_{j=0}^{J-1} \sum_{l=0}^{L-1} \rho_{j,l} e^{-(2\pi ijm)/J} e^{-(2\pi iln)/L}$$

2. On calcule $\hat{u}_{m,n}$ à partir de l'équation (6.22)
3. On calcule $u_{j,l}$ en utilisant la transformée de Fourier inverse.

Cette procédure est valide pour des conditions de bord périodiques. Elle sera modifiée dans le cas de conditions limites de type Dirichlet (voir *Numerical Recipes* , chapitre *Partial Differential Equations*).

Chapitre 7

Résolution d'équations non-linéaires

Sommaire

7.1 Introduction

Dans ce chapitre il s'agit de caractériser numériquement un état d'équilibre pour une fonction donnée. Cela passe par la recherche des points fixes d'une équation de type $x = g(x)$, ou par la recherche des racines d'une équation de type $f(x) = 0$.
Il s'agira aussi de trouver des solutions pour des systèmes d'équations non-linéaires $F(\mathbf{x}) = \mathbf{0}$, ce qui est par essence plus compliqué que la recherche d'une racine pour une équation à une dimension. En effet, en présence de N dimensions la solution n'est plus donnée par un point d'intersection d'une courbe avec un axe qui peut facilement être pressentie, mais par une hypersurface dont l'existence n'est avérée qu'une fois effectivement trouvée.

Dans tous les cas, il faudra utiliser des méthodes itératives qui vont consister à approcher la solution par itérations successives jusqu'à l'accepter sur la base d'une

condition de tolérance qu'on aura prédéfinie.

Il est nécessaire d'établir un critère de convergence de la suite construite pour approcher la solution de l'équation (ou du système d'équations) non-linéaire étudiée.

Un critère sur le nombre d'itérations à considérer avant d'estimer que la suite a convergé est à proscrire.
On adoptera plutôt un critère de convergence basé sur l'évolution de la ***racine***

$$\Delta_n(p) \equiv |p_n - p_{n-1}| \leq C_p \quad \text{ou} \quad \varepsilon_n(p) \equiv \left| \frac{p_n - p_{n-1}}{p_{n-1}} \right| \leq E_p$$

ou un critère sur la ***fonction elle-même***

$$\Delta_n(f) \equiv |f(p_n) - f(p_{n-1})| \leq C_f$$

7.2 Ordre et convergence d'une méthode

Définition : Ordre d'une racine

Pour une fonction $f(x)$ et ses dérivées $f^{(M)}(x)$ définies et continues sur un intervalle autour de $x = p$, on dira que l'équation $f(x) = 0$ possède une racine d'ordre M en $x = p$ si et seulement si la fonction et toutes ses dérivées sont nulles en ce point sauf la dérivée M -ième $f^{(M)}(p) \neq 0$.

Pour $M > 1$, on parlera de ***racines multiples***.

Si la fonction $f(x)$ a une racine d'ordre M en $x = r$, alors il existe une fonction continue $h(x)$ telle que

$$f(x) = (x - r)^M h(x)$$

Définition : Ordre de convergence

Soit la suite $\{c_n\}_{n=0}^{\infty}$ qui converge vers r et $E_n = r - c_n$ pour $n \in \mathbb{N}$.
S'il existe deux constantes $A \neq 0$ et $R > 0$ telles que

$$lim_{n\to\infty} \frac{|r - c_{n+1}|}{|r - c_n|^R} = lim_{n\to\infty} \frac{|E_{n+1}|}{|E_n|^R} = A \tag{7.1}$$

on dira que la suite converge vers r avec un ***ordre de convergence*** R. A est appelée ***constante d'erreur asymptotique***.

7.3 Points fixes de l'équation $x = g(x)$

On se place dans le cas de fonctions $g(x)$ pour lesquelles la suite p_k telle $g(p_k) = p_{k+1}$ converge vers une valeur finie.

Définition

Un ***point fixe*** de la fonction $g(x)$ est un nombre réel P tel que $P = g(P)$.

Théorème

Soit g une fonction continue et $\{p_n\}_{n=0}^{\infty}$ une suite générée par l'itération de point fixe $p_{n+1} = g(p_n), \quad \forall n \in \mathbb{N}$. Si $lim_{n\to\infty} p_n = P$, alors P est un point fixe de la fonction $g(x)$.

Théorème

Soient
- g et $g' \in C([a, b])$,
- K une constante positive,
- $p_0 \in (a, b)$,
- $g(x) \in [a, b] \ \ \forall\ x \in [a, b]$

Alors

1. Si $|g'(x)| \leq K < 1$ pour tout $x \in [a, b]$, la suite $\{p_k\}$ définie par l'itération $p_n = g(p_{n-1})$ converge vers un unique point fixe P $\in [a, b]$. Dans ce cas on dira que P est un ***point fixe attractif***.
   Dans ce cas, l'erreur faite lors de l'approximation de P par l'élément p_n de la suite $\{p_k\}$ est bornée et on a

   $$|P - p_n| \leq K^n |P - p_0| \quad \forall\ n \geq 1$$

   et

   $$|P - p_n| \leq \frac{K^n |p_1 - p_0|}{1 - K} \quad \forall\ n \geq 1$$

2. Si $|g'(x)| > 1$ pour tout $x \in [a, b]$, alors la suite $\{p_k\}$ définie par l'itération $p_n = g(p_{n-1})$ ne converge pas vers P. Dans ce cas, on dira que P est un ***point fixe répulsif***, et que la suite présente une ***divergence locale***.

7.4 Méthodes d'encadrement d'une racine

On se concentre à présent sur la résolution d'une équation à une variable du type $f(x) = 0$, c-à-d sur la recherche des zéros de la fonction f.
Si ces zéros existent, graphiquement cela signifie que sur un intervalle de définition donné $[a, b]$, la fonction change de signe.
Les méthodes d'encadrement de la racine décrites ci-après sont basées sur ce fait.

7.4.1 Méthode de dichotomie ou Bissection

Le principe de la méthode de dichotomie consiste à construire une suite d'intervalles $[a_n, b_n]$ contenant une racine p de l'équation $f(x) = 0$, de sorte que l'une des deux bornes soit le milieu de l'intervalle précédent $[a_{n-1}, b_{n-1}]$:

$$a_0 \leq a_1 \leq \cdots \leq a_n \leq \cdots \leq r \leq \cdots \leq b_n \leq \cdots \leq b_1 \leq b_0 \tag{7.2}$$

On définit $c_n = \frac{a_n + b_n}{2}$ et si $f(a_{n+1})f(b_{n+1}) < 0$ alors on posera

$$[a_{n+1}, b_{n+1}] = [a_n, c_n] \quad \text{ou} \quad [a_{n+1}, b_{n+1}] = [c_n, b_n] \quad \forall n \tag{7.3}$$

Théorème

Soit $f \in C([a, b])$, et $r \in [a, b]$ tel que $f(r) = 0$. Si $f(a)$ et $f(b)$ sont de signe opposé, et que $\{c_n\}_{n=0}^{\infty}$ représente une suite de points médians générée par une procédure de bissection telle que décrite ci-dessus par les équations (7.2) et (7.3), alors on aura

$$|r - c_n| \leq \frac{b - a}{2^{n+1}} \quad \text{pour tout } n \in \mathbb{N} \tag{7.4}$$

et la suite $\{c_n\}_{n=0}^{\infty}$ converge vers le zéro de la fonction f soit $x = r$

$$\lim_{n \to \infty} c_n = r$$

Le critère de convergence de la méthode de dichotomie porte sur la racine et non pas sur la valeur de la fonction évaluée en $x = c_n$:

$$|p_{n+1} - p_n| < \varepsilon$$

L'algorithme pour la bissection donne l'enchaînement d'opérations suivant :

Soient a et b deux points tels que $f(a)f(b) < 0$ et r une racine de l'équation $f(x) = 0$ dans l'intervalle $]a, b[$.
A l'itération 0, on a

$$a_0 = a \quad \text{et} \quad b_0 = b$$

A chaque itération n on aura

- $p \in]a_{n-1}, b_{n-1}[$ et $c_n = \frac{a_{n-1} + b_{n-1}}{2}$
- Si $f(a_{n-1})f(c_n) > 0$ alors $a_n = c_n$ et $b_n = b_{n-1}$
- Sinon si $f(a_{n-1})f(c_n) < 0$ alors $b_n = c_n$ et $a_n = a_{n-1}$
- Sinon si $f(a_{n-1})f(c_n) = 0$ alors $p = c_n \quad \Rightarrow \quad$ **FIN**

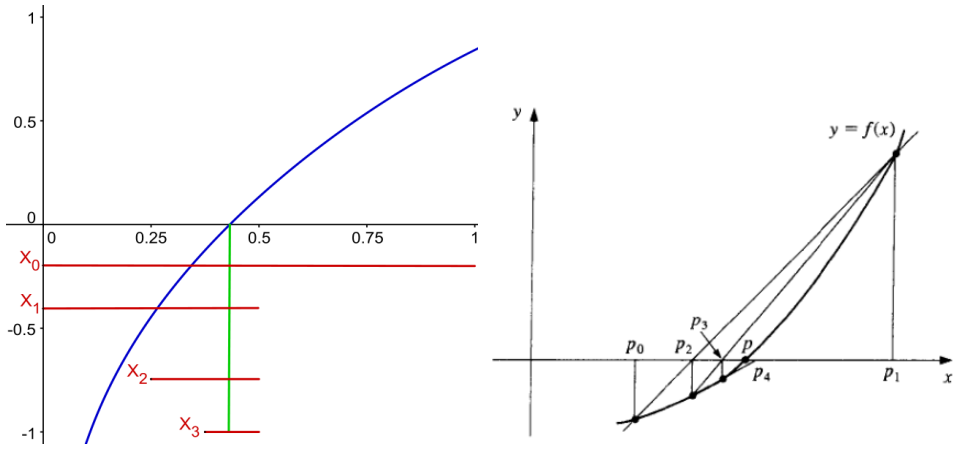


FIGURE 7.1 – Illustration de la méthode de dichotomie (à gauche) et de la méthode de la fausse position (à droite).

– Sinon si $|b_n - a_n| < \varepsilon$ alors $p \in]a_n, b_n[$ $\Rightarrow$ **FIN**

L'erreur sur l'estimation de la racine est de la largeur du dernier intervalle $|(b_n - a_n)|$.

7.4.2 Méthode de la fausse position ou *regula falsi*

Cet algorithme est une version améliorée de l'algorithme de bissection qui a une convergence assurée mais lente.

Comme pour la dichotomie, on suppose que $f(a)$ et $f(b)$ sont de signe opposé. Au lieu de se placer au centre de l'intervalle comme pour la dichotomie, on accélère le processus de convergence en prenant c le point d'intersection entre l'axe des abscisses et la droite reliant les points $(a, f(a))$ et $(b, f(b))$. La pente de cette droite peut s'écrire de deux façons :

$$\begin{cases} m = \frac{f(b)-f(a)}{b-a} \\ m = \frac{0-f(a)}{c-a} \end{cases}$$

La combinaison de ces deux expressions donne

$$c = b - \frac{f(b)(b-a)}{f(b) - f(a)} \tag{7.5}$$

L'algorithme que l'on applique est ensuite le même que celui décrit pour la bissection.
On notera cependant que le critère de convergence de cette méthode n'est plus seulement un critère sur la racine mais qu'on a aussi besoin d'un critère sur la fonction elle-même.

On doit donc à la fois se donner une tolérance pour la racine ε mais aussi pour la fonction au voisinage du zéro δ.

A chaque itération n on aura

- $p \in]a_{n-1}, b_{n-1}[$ et $c_n = b_{n-1} - \frac{f(b_{n-1})(b_{n-1}-a_{n-1})}{f(b_{n-1})-f(a_{n-1})}$
- Si $f(a_{n-1})f(c_n) > 0$ alors $a_n = c_n$ et $b_n = b_{n-1}$
- Sinon si $f(a_{n-1})f(c_n) < 0$ alors $b_n = c_n$ et $a_n = a_{n-1}$
- Sinon si $f(a_{n-1})f(c_n) = 0$ alors $p = c_n \quad \Rightarrow$ **FIN**
- Sinon si $|b_n - a_n| < \varepsilon$ alors $p \in]a_n, b_n[\quad \Rightarrow$ **FIN**
- Sinon si $|f(c_n)| < \delta$ alors $p \in]a_n, b_n[\quad \Rightarrow$ **FIN**

L'erreur sur l'estimation de la racine est de la moitié de la largeur du dernier intervalle $|(b_n - a_n)|/2$.

Les méthodes de la fausse position et de la dichotomie sont dites méthodes ***globalement convergentes***.
Leur convergence autour d'une des racines de l'équation $f(x) = 0$ est toujours vérifiée mais elle peut être lente.
Il existe une autre famille de méthodes que l'on appelle ***localement convergentes***.
Elles convergent plus vite que les méthodes à convergence globale mais elles nécessitent un choix éclairé des valeurs initiales pour la récurrence.

7.5 Méthodes à convergence locale

7.5.1 Méthode de Newton-Raphson

La méthode de Newton consiste à construire une suite $\{c_n\}_{n=0}^{\infty}$ telle que le terme c_{n+1} est l'intersection de l'axe des abscisses avec la droite tangente à la fonction f au point $(c_n, f(c_n))$. Si cette suite converge, elle tend vers une limite p qui est racine de l'équation $f(x) = 0$.

Théorème

Soit $f \in C^2([a, b])$ telle qu'il existe un nombre $r \in [a, b]$ qui vérifie $f(r) = 0$. Si $f'(r) \neq 0$, alors il existe un nombre $\delta > 0$ tel que la suite $\{c_n\}_{n=0}^{\infty}$ définie par l'itération

$$c_k = g(c_{k-1}) = c_{k-1} - \frac{f(c_{k-1})}{f'(c_{k-1})} \quad \text{pour } k = 1, 2, ... \tag{7.6}$$

converge vers r quelle que soit la valeur initiale de l'approximation de cette racine $c_0 \in [r - \delta, r + \delta]$.

La fonction $g(x) = x - \frac{f(x)}{f'(x)}$ est appelée ***fonction d'itération de Newton-Raphson.***

Convergence

Pour la méthode de Newton-Raphson, le critère de convergence s'applique sur la racine.

Théorème

Soit la suite $\{c_n\}_{n=0}^{\infty}$ caractérisant la méthode de Newton-Raphson, et qui converge vers le zéro r de la fonction $f(x)$.

Si r est une racine simple, la convergence est quadratique ($R = 2$ dans l'équation (7.1)) et pour n assez grand on a

$$|E_{n+1}| \approx \frac{|f''(r)|}{2|f'(r)|}|E_n|^2$$

Si r est une racine multiple, la convergence est linéaire ($R = 1$ dans l'équation (7.1)) et pour n assez grand on a

$$|E_{n+1}| \approx \frac{M-1}{M}|E_n|$$

Divergence

Comme toutes les méthodes ***localement convergentes***, la méthode de Newton-Raphson ne converge pas systématiquement, et la divergence peut être due à la forme de la fonction à laquelle on l'applique aussi bien qu'à un mauvais choix de la valeur d'initialisation de la suite $\{c_n\}_{n=0}^{\infty}$.

Voici un petit florilège de configurations dans lesquelles la méthode de Newton-Raphson ne converge pas.

- Cas des fonctions n'admettant pas de racine réelle

$$f(x) = x^2 - 4x + 5$$

- Cas où la dérivée au voisinage du point d'initialisation c_0 de la suite est très petite

$$f(x) = cos(x) \text{ avec } c_0 = 3 \text{ ou } c_0 = \pi/2$$

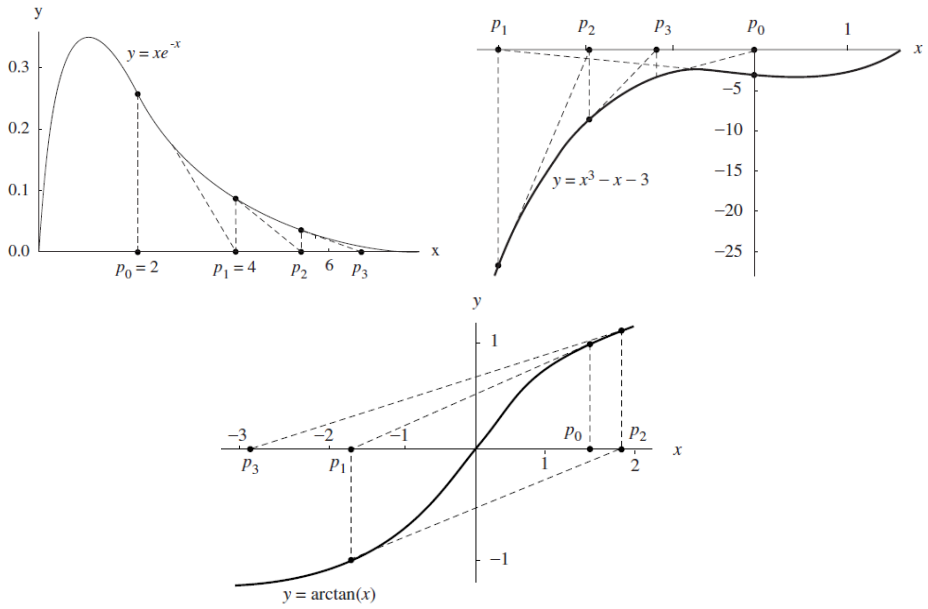


FIGURE 7.2 – Itérations de Newton-Raphson pour la fonction xe^{-x} avec $c_0 = 2$ *(en haut à gauche)*, pour la fonction $f(x) = x^3 - x - 3$ avec une initialisation en $c_0 = 0$ *(en haut à droite)*, pour la fonction $f(x) = arctan(x)$ avec une initialisation en $c_0 = 1.45$ *(en bas)*

- Cas des fonctions avec une décroissance monotone sur un intervalle $[a, +\infty[]$ si le point d'initialisation de la suite $c_0 > a$

$$f(x) = xe^{-x} \quad \text{avec} \quad c_0 = 2$$

- Cas de fonctions pour lesquelles le choix du point d'initialisation de la suite implique une répétition quasi à l'identique des termes suivants

$$f(x) = x^3 - x - 3 \quad \text{avec } c_0 = 0 \quad \text{ou} \quad c_0 = 1$$

- Cas où la dérivée de la fonction d'itération $g(x)$ est telle que $|g(x)| \geq 1$ sur un intervalle contenant la racine r de la fonction f

$$f(x) = arctan(x) \quad \text{avec} \quad c_0 = 1.45 \quad \text{ou} \quad c_0 = 0$$

Accélération de la convergence

On a vu que pour les racines d'ordre supérieur à 1, la méthode de Newton-Raphson passe d'une convergence quadratique à une convergence linéaire.
Afin de revenir à une convergence quadratique, on peut appliquer ce que l'on appelle une méthode d'***accélération de la convergence***.

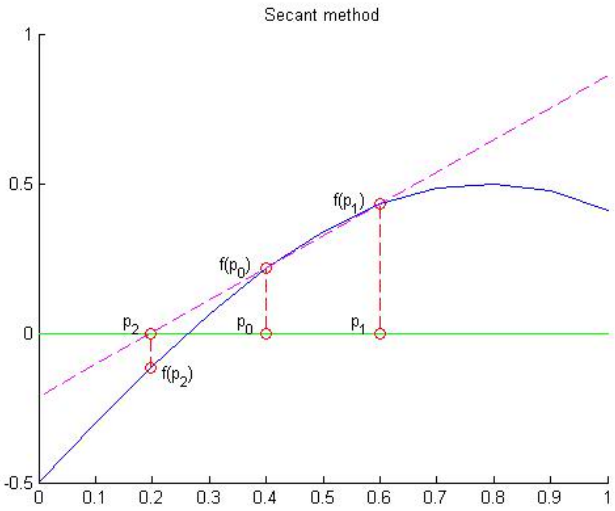


FIGURE 7.3 – Illustration graphique de la méthode de la sécante

Théorème

Soit une suite $\{c_n\}_{n=0}^{\infty}$ utilisée pour trouver la racine d'une équation $f(x) = 0$ par la méthode de Newton-Raphson. Si cette suite converge vers une racine $x = r$ d'ordre $M > 1$, alors la suite $\{q_n\}_{n=0}^{\infty}$ définie par

$$q_k = q_{k-1} - \frac{Mf(q_{k-1})}{f'(q_{k-1})} \tag{7.7}$$

converge quadratiquement vers r.

7.5.2 Méthode de la sécante

La méthode de Newton-Raphson requiert l'évaluation de deux fonctions, f et f' à chaque itération, et en particulier au point c_0 d'initialisation de la suite $\{c_n\}_{n=0}^{\infty}$. Ceci peut s'avérer difficile à calculer selon la forme de la fonction f.
Afin de contourner le problème éventuel du calcul de la dérivée première de la fonction f, on peut la remplacer par son approximation tirée du développement de Taylor à l'ordre 1. C'est ce qui est fait dans la méthode de la sécante.

On utilise donc le même algorithme que pour la méthode de Newton-Rpahson en remplaçant $f'(c_n)$ par

$$f'(c_n) = \frac{f(c_n) - fc_{n-1}}{c_n - c_{n-1}}$$

On remarquera que cela nécessite d'initialiser le problème avec non plus une mais deux valeurs c_0 et c_1 que l'on choisit assez proches de la racine r [1].

A l'itération n on calculera donc

$$c_{n+1} = c_n - \frac{f(c_n)\,(c_n - c_{n-1})}{f(c_n) - f(c_{n-1})} = \frac{c_n f(c_{n-1}) - c_{n-1} f(c_n)}{f(c_n) - f(c_{n-1})} \tag{7.8}$$

Une autre façon de voir la méthode de la sécante est de considérer qu'il s'agit d'une révision de la méthode de la fausse position vue plus haut.

Comme dans la méthode *regula falsi*, on considère l'intersection de la droite reliant deux points $(c_0, f(c_0))$ et $(c_1, f(c_1))$ proches du point $(r, 0)$, et on cherche c_2 le point d'intersection de cette droite avec l'axe des abscisses.
Comme dans la méthode de la fausse position, on peut déterminer la pente de cette droite en utilisant soit les points $(c_0, f(c_0))$ et $(c_1, f(c_1))$, soit les points $(c_1, f(c_1))$ et $(r, 0)$:

$$m = \frac{f(c_1) - f(c_0)}{c_1 - c_0} \quad \text{ou} \quad m = \frac{0 - f(c_1)}{r - c_1}$$

En combinant ces deux expressions on trouve

$$c_2 = c_1 - \frac{f(c_1)\,(c_1 - c_0)}{f(c_1) - f(c_0)}$$

ce qui nous redonne bien l'équation (7.8) quand on généralise pour un indice n quelconque.

On notera que le critère de convergence de cette méthode se fait également sur la racine $|c_n - c_{n-1}| < \varepsilon$.

Convergence

Théorème

Soir f une fonction continue de classe $C^1(I)$, avec I un intervalle dans $\mathbb{R}$.
Si f possède une racine simple en $x = r$ sur l'intervalle I, alors l'ordre de convergence de la suite définissant la méthode de la sécante est égal au nombre d'or :

1. Comme la méthode de Newton-Raphson, la méthode de la sécante converge localement et le choix des points d'initialisation est donc important pour la convergence de la suite.

$$|E_{k+1}| \approx |E_k|^{1.618} \left| \frac{f''(r)}{2f'(r)} \right|^{0.618} \tag{7.9}$$

où

$$R = \frac{1+\sqrt{5}}{2} \approx 1.618$$

(voir équation (7.1)).

7.6 Racines de polynômes

Lorsque on recherche le zéro d'une fonction qui est un polynôme, ou plus généralement lorsqu'on recherche des racines d'une équation $f(x) = 0$ qui possède des racines multiples, on utilisera des méthodes dites de ***déflation***. Ces méthodes consistent à appliquer des factorisations successives pour trouver les racines.

Pour une fonction $f(x)$ avec plusieurs racines, on commence par chercher une des racines r, puis on pose

$$F(x) = \frac{f(x)}{(x-r)}$$

On cherche ensuite une racine de l'équation $F(x) = 0$ et on répète le processus jusqu'à avoir trouvé toutes les racines.

7.6.1 Méthode de Laguerre pour les polynômes de degré n

La méthode de Laguerre consiste à fabriquer une méthode itérative qui converge vers une racine quel que soit le point de départ de la méthode, puis qui utilise la déflation pour factoriser par cette racine et recommencer le processus.

On considère un polynôme $P(x) \in \mathbb{P}_{\ltimes}$ et on définit les fonctions suivantes

$$G(x) = \frac{P'(x)}{P(x)} \qquad \text{et} \qquad H(x) = G(x)^2 - \frac{P''(x)}{P(x)} \tag{7.10}$$

On remarque immédiatement que $G(x)$ est en fait la dérivée du logarithme du polynôme P, soit $\frac{dln(P(x))}{dx}$. $H(x)$ correspond à la dérivée seconde du logarithme du polynôme $P(x)$ multipliée par -1
Soit maintenant x_0 une estimation de la racine du polynôme P que l'on recherche. La valeur de cette racine est obtenue en minimisant la différence $a = x_{k+1} - x_k$ où les termes de la suite $\{x_k\}$ sont définis par la récurrence suivante

$$x_{k+1} = x_k - \frac{n}{G(x) + s\sqrt{(n-1)\left(nH(x) - G(x)^2\right)}} \tag{7.11}$$

où $s = \pm 1$ est choisi à chaque itération de façon à minimiser le dénominateur.

Pour trouver l'équation (7.11), on fait deux hypothèses fortes :

1. la racine recherchée, x_1 se trouve à une distance $a = x_0 - x_1$ de l'estimation utilisée comme point de départ
2. ***toutes les autres racines*** x_i ***(i = 2,...,n) du polynôme*** P sont équidistantes du point de départ x_0 et se trouvent à une distance $b = x_0 - x_i = cte$ $\forall\ i > 1$ de ce point.

En utilisant ces deux hypothèses pour exprimer G et H, on obtient le système suivant

$$\begin{cases} \frac{1}{a} + \frac{n-1}{b} = G \\ \frac{1}{a^2} + \frac{n-1}{b^2} = H \end{cases}$$

qui va donner pour a :

$$a = \frac{n}{G(x) + s\sqrt{(n-1)\,(nH(x) - G(x)^2)}}$$

ce qui donne bien l'équation (7.11).

On acceptera la racine lorsque a sera suffisamment petit.
Cette méthode diverge très rarement et fonctionne aussi pour des $a \in \mathbb{C}$. Pour éviter les très rares cas de non-convergence (présence de cycle limite), on pourra mettre en place des stratégies consistant à changer sporadiquement la taille du pas d'intégration (voir *Numerical Recipes*, chap. 9).

7.6.2 Accélération de la convergence : procédé d'Aitken

Afin d'accélérer le processus de convergence des suites que l'on doit résoudre pour trouver la ou les racines de l'équation $f(x) = 0$, on peut appliquer le ***procédé d'Aitken*** qui est ***valable pour toute suite à convergence linéaire***.

Définition

Pour une suite $\{c_n\}_{n=0}^{\infty}$, on définit la différence finie à deux points vers l'avant Δc_n

$$\Delta c_n = c_{n+1} - c_n$$

Les différences finies de degré supérieur sont définies récursivement par

$$\Delta^k c_n = \Delta^{k-1}(\Delta c_n)$$

Théorème

Soit une suite $\{c_n\}_{n=0}^{\infty}$ qui converge linéairement vers la limite r. On suppose que $r - c_n \neq 0 \; \forall \; n \geq 0$.
S'il existe un réel A, $|A| < 1$ tel que

$$\lim_{n\to\infty} \frac{r - c_{n+1}}{r - c_n} = A$$

alors la suite $\{q_n\}_{n=0}^{\infty}$ définie par

$$q_n = c_n - \frac{(\Delta c_n)^2}{\Delta^2 c_n} = c_n - \frac{(c_{n+1} - c_n)^2}{c_{n+2} - 2c_{n+1} + c_n}$$

converge plus rapidement vers r et on a

$$\lim_{n\to\infty} \left| \frac{r - q_n}{r - c_n} \right| = 0$$

7.7 Résolution de systèmes d'équations non-linéaires

Dans les paragraphes précédents, nous avons décrit un certain nombre de méthodes pour trouver les racines d'équations unidimensionnelles.
En physique pourtant c'est souvent à des équations à plus d'une dimension que l'on a à faire. Dans ce cas, on utilisera préférentiellement la méthode de Newton-Raphson, qui est la même que celle décrite au § 7.5.1, mais où en lieu et place de la dérivée de la fonction, on va devoir manipuler le jacobien du système d'équations.
Nous donnons ici la description détaillée de cette méthode dans le cas à 2 dimensions.

7.7.1 Matrice jacobienne

Soit un système de n équations à n inconnues $f_k(x_1, x_2, ..., x_n), \quad k = 1, ..., n$.
La matrice jacobienne ou ***jacobien*** de ce système est notée $\mathbf{J}(x_1, x_2, ..., x_n)$ et est définie par

$$\mathbf{J} = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \dots & \dots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \dots & \dots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \frac{\partial f_{n-1}}{\partial x_1} & \frac{\partial f_{n-1}}{\partial x_2} & \dots & \dots & \frac{\partial f_{n-1}}{\partial x_n} \\ \frac{\partial f_n}{\partial x_1} & \frac{\partial f_n}{\partial x_2} & \dots & \dots & \frac{\partial f_n}{\partial x_n} \end{pmatrix}$$

Pour un système d'équations

$$\begin{cases} u = f_1(x, y, z) \\ v = f_2(x, y, z) \\ w = f_3(x, y, z) \end{cases} \tag{7.12}$$

où la valeur des fonctions est connue au point (x_0, y_0, z_0), on peut définir le jacobien en ce point par la relation

$$d\mathbf{F} = \mathbf{J}(x_0, y_0, z_0)d\mathbf{X} \tag{7.13}$$

où les quantités notées en gras sont des vecteurs.

7.7.2 Méthode de Newton-Raphson à 2 dimensions

Soit le système suivant

$$\begin{cases} u = f_1(x, y) \\ v = f_2(x, y) \end{cases} \tag{7.14}$$

On s'intéresse au comportement de ce système au voisinage du point (x_0, y_0) ayant pour image (u_0, v_0).

Si f_1 et f_2 sont des fonctions aux dérivées partielles continues, on peut écrire les développements de Taylor à l'ordre 1

$$\begin{aligned} \Delta u = u - u_0 &\approx (x - x_0) \left.\frac{\partial f_1}{\partial x}\right|_{(x_0,y_0)} + (y - y_0) \left.\frac{\partial f_1}{\partial y}\right|_{(x_0,y_0)} \\ \Delta v = v - v_0 &\approx (x - x_0) \left.\frac{\partial f_2}{\partial x}\right|_{(x_0,y_0)} + (y - y_0) \left.\frac{\partial f_2}{\partial y}\right|_{(x_0,y_0)} \end{aligned} \tag{7.15}$$

On cherche les racines du système (7.14) en posant $u = v = 0$, et en notant (p, q) une racine de ce système.

Si on considère des petites variations au voisinage du point de départ (p_0, q_0) ayant pour image (u_0, v_0), et que l'on adopte la notation

$$\begin{cases} \Delta p = x - p_0 \\ \Delta q = y - q_0, \end{cases}$$

en posant (x,y) = (p,q) et en utilisant les expressions de Δu et Δv données en (7.15), on a alors le système suivant

$$\begin{aligned} \Delta u = u - u_0 = f_1(p, q) - f_1(p_0, q_0) = -f_1(p_0, q_0) &\approx \Delta p \left.\frac{\partial f_1}{\partial x}\right|_{(p_0,q_0)} + \Delta q \left.\frac{\partial f_1}{\partial y}\right|_{(p_0,q_0)} \\ \Delta v = v - v_0 = f_2(p, q) - f_2(p_0, q_0) = -f_2(p_0, q_0) &\approx \Delta p \left.\frac{\partial f_2}{\partial x}\right|_{(p_0,q_0)} + \Delta q \left.\frac{\partial f_2}{\partial y}\right|_{(p_0,q_0)} \end{aligned} \tag{7.16}$$

soit encore

$$\underbrace{\begin{pmatrix} \left.\frac{\partial f_1}{\partial x}\right|_{(p_0,q_0)} & \left.\frac{\partial f_1}{\partial y}\right|_{(p_0,q_0)} \\ \left.\frac{\partial f_2}{\partial x}\right|_{(p_0,q_0)} & \left.\frac{\partial f_2}{\partial y}\right|_{(p_0,q_0)} \end{pmatrix}}_{Jacobien} \begin{pmatrix} \Delta p \\ \Delta q \end{pmatrix} = - \begin{pmatrix} f_1(p_0,q_0) \\ f_2(p_0,q_0) \end{pmatrix}$$

Si la matrice jacobienne $\mathbf{J}(p_0, q_0)$ n'est pas une matrice singulière, on peut résoudre ce système en inversant le jacobien

$$\Delta\mathbf{P} \approx -\mathbf{J}^{-1}\mathbf{F}(p_0, q_0)$$

Une nouvelle approximation $\mathbf{P_1} = \begin{pmatrix} p_1 \\ q_1 \end{pmatrix}$ de la solution $\mathbf{P} = \begin{pmatrix} p \\ q \end{pmatrix}$ s'écrit alors :

$$\mathbf{P_1} = \mathbf{P_0} + \Delta\mathbf{P} = \mathbf{P_0} - \mathbf{J}^{-1}\mathbf{F}(p_0, q_0) \tag{7.17}$$

7.7.3 Algorithme pour la méthode de Newton-Raphson

On part d'un point $\mathbf{P_0} = \begin{pmatrix} p_0 \\ q_0 \end{pmatrix}$

A chaque itération k

- on évalue la fonction

$$\mathbf{FP_k} = \begin{pmatrix} f_1(p_k, q_k) \\ f_2(p_k, q_k) \end{pmatrix}$$

- on évalue le jacobien du système

$$\mathbf{J} = \begin{pmatrix} \left.\frac{\partial f_1}{\partial x}\right|_{(p_k,q_k)} & \left.\frac{\partial f_1}{\partial y}\right|_{(p_k,q_k)} \\ \left.\frac{\partial f_2}{\partial x}\right|_{(p_k,q_k)} & \left.\frac{\partial f_2}{\partial y}\right|_{(p_k,q_k)} \end{pmatrix}$$

- on résout le système linéaire

$$\mathbf{J(P_k)}\Delta\mathbf{P} = -\mathbf{F(P_k)} \quad \text{pour} \quad \Delta\mathbf{P}$$

- on calcule le point suivant

$$\mathbf{P_{k+1}} = \mathbf{P_k} + \Delta\mathbf{P}$$

- on s'arrête lorsque $|\mathbf{P_{n+1}} - \mathbf{P_n}| < \varepsilon$

Chapitre 8

Introduction aux méthodes de Monte Carlo

Sommaire

8.1 Introduction

On désigne par le terme générique ***méthode de Monte Carlo*** les méthodes statistiques de résolution d'un problème physique qui utilisent des nombres aléatoires.

Utilisées de façon régulière dans de nombreuses disciplines (économie, sciences du comportement, ...), on les emploie en physique pour :

– simuler des processus complexes intrinsèquement stochastiques
– remplacer un problème déterministe par un problème stochastique
– calculer des intégrales multidimensionnelles.

Les simulations de type Monte-Carlo décrivent directement le système physique qu'elles représentent plutôt que son comportement comme c'est le cas pour les méthodes déterministes. Elles sont utilisables dès lors que le système physique peut être décrit par des lois de probabilité.

8.2 Principe des méthodes de Monte-Carlo

Les méthodes de Monte-Carlo ont toutes une structure similaire que l'on décrit ici.

En supposant qu'un système physique peut être décrit par des densités de probabilité identifiées, une simulation Monte-Carlo de ce système consistera à faire un nombre important de tirages aléatoires sur les configurations possibles du système. La moyenne des réalisations de ces expériences aléatoires donnera une estimation de la solution du problème.

Les composantes principales d'un algorithme de Monte-Carlo sont :

1. les densités de probabilité décrivant le système
2. un générateur de nombres aléatoires
3. une règle de tirage aléatoire
4. une estimation de l'erreur sur la solution du problème
5. une méthode de réduction de la variance

8.3 Processus stochastiques et chaînes de Markov

Définition - processus stochastique

On appelle processus stochastique tout processus dont le résultat est soumis au hasard.

De façon plus formelle, un processus stochastique est une famille $\{Y_t,\ t \in I\}$ de variables aléatoires définies sur un même espace de probabilité.

L'indice t est souvent assimilé au temps. Ainsi on dira qu'un processus stochastique est en **temps continu** si I est continu (c-à-d $I = [0, \infty)$), en en **temps discret** si I est discret (c-à-d, I= {1,2,3,...}).

Lorsque t est continu, on notera $Y_t = Y(t)$. On supposera que Y_t prend ses valeurs dans un espace réel à d dimensions $\mathbb{R}^d$.

Définition - Processus markovien / Chaîne de Markov

Une chaîne de Markov est une séquence d'événements soumis au hasard dont le résultat n'est pas déterminé par les configurations passées du système auquel ces événements se réfèrent.
Plus formellement, on dira qu'un processus stochastique est **markovien** si, conditionnellement à sa valeur à l'instant présent t, son évolution future est indépendante de son passé.
Pour toute variable aléatoire X fonction de $\{Y_s,\ s > t\}$, la loi de X conditionnelle à $\{Y_s,\ s \leq t\}$ est la même que celle conditionnelle à Y_t.
Y_t contient toujours assez d'information pour "générer" le futur.

On parlera de chaîne de Markov temps discret lorsque $I = \{0, 1, ...\}$.

8.4 Rappels de probabilités

8.4.1 Probabilités discrètes : définitions

Définitions de variable aléatoire, espérance et variance

Voir chapitre 3 sur l'interpolation, Extrapolation, Ajustement (paragraphe 3.6).

Définition - Probabilité conditionnelle - Fonction de répartition

Soit X une variable aléatoire définie sur l'univers de l'expérience Ω muni de la probabilité P. L'ensemble des éléments de Ω dont l'image par X est inférieure ou égale au nombre réel x se note : X $\leq$ x, et cet ensemble est un événement. La probabilité de cet événement se note : $P(X \leq x)$.
La **fonction de répartition de la variable aléatoire X** est la fonction F_X (parfois notée simplement : F) définie sur $\mathbb{R}$ par :

$$F_X(x) = P(X \leq x)$$

Définition - Loi de probabilité d'une variable aléatoire discrète

Soit X une variable aléatoire discrète. Soit $x_i \in \mathbb{R} \mid$ i $\in$ I, avec I $\subset \mathbb{N}$, l'ensemble des images par X des éléments de Ω (Ω est l'univers de l'expérience muni de la probabilité P). La loi de probabilité de X est l'ensemble des couples : $(x_i, P(X = x_i$)) pour i $\in$ I.

8.4.2 Probabilités continues

Définition - Densité de probabilité d'une variable aléatoire continue X

Une fonction f est une ***densité de probabilité*** sur un intervalle I donné si :

$$\begin{cases} f \text{ est positive ou nulle et continue (intégrable) sur I} \\ \int_I f(x)dx = 1 \end{cases}$$

Soit X une variable aléatoire pouvant prendre toutes les valeurs sur un intervalle réel $[a, b]$ et f une densité de probabilité sur ce même intervalle.
On dira que la **loi de probabilité** de X est de **densité de probabilité** f sur $[a, b]$ si

$$\forall\ X \in [a, b], \quad P(X \in [a, b]) = \int_a^b f(x')dx' \tag{8.1}$$

La probabilité pour que X soit égale exactement à une valeur c de $[a, b]$ est nulle dans ce cas.

Exemple de lois de probablilité (voir plus bas)

- Loi uniforme
- Loi normale ou Gaussienne
- Loi de Poisson

Définition - Fonction de répartition d'une variable aléatoire continue X sur $\mathbb{R}$

Soit X une variable aléatoire continue à valeurs sur $\mathbb{R}$, de densité de probabilité f. La fonction de répartition de X, soit F, définie sur $\mathbb{R}$ et à valeurs dans $[0, 1]$ est définie par :

$$\forall A \in \mathbb{R}, \quad F(A) = P(X \leq A) = \int_{-\infty}^{A} f(x')dx' \tag{8.2}$$

Définition - Espérance et variance d'une variable aléatoire continue X sur $\mathbb{R}$

L'espérance d'une variable aléatoire continue X est le premier moment de cette variable et se définit comme suit :

$$E(X) = \mu = \bar{X} = \int_{-\infty}^{+\infty} f(x')x'dx' \tag{8.3}$$

La variance est alors donnée par :

$$V(X) = \sigma^2 = \int_{-\infty}^{+\infty} f(x')(x' - \mu)^2 dx' = \bar{X^2} - \bar{X}^2 \tag{8.4}$$

Définition - Variables aléatoires indépendantes

On dira que n variables aléatoires $X_1, X_2, \cdots, X_n$ sont indépendantes si

$$E\left(\prod_i f_i(X_i)\right) = \prod_i E\left(f_i(X_i)\right)$$

Dans le cas de variables aléatoires indépendantes, on aura aussi

$$V\left(\sum_i X_i\right) = \sum_i V(X_i)$$

On aura par contre $V\left(\sum_i(-X_i)\right) = \sum_i V(X_i)$.

Enfin, la variance de la moyenne de variables aléatoires independantes $\bar{X} = \frac{1}{n}\sum_{i=1}^{n} X_i$ vaut

$$V(\bar{X}) = V\left(\frac{1}{n}\sum_{i=1}^{n} X_i\right) = \frac{V(X)}{n} = \frac{\sigma^2}{n}$$

8.4.3 Expérience historique de Monte-Carlo : problème de Buffon

La plus vieille expérience de type Monte-Carlo documentée est l'expérience du Comte de Buffon en 1777, connue sous le nom de problème des aiguilles de Buffon.

L'expérience de Buffon consiste à jeter une aiguille de longueur l au hasard sur un plancher de bois fait de planches parallèles de largeur d, $(d > l)$, et de calculer la probabilité pour que l'aiguille tombe sur une intersection entre deux planches.

Si on pose r la distance entre le centre de l'aiguille et la jointure la plus proche, et θ l'angle entre l'aiguille et la jointure (voir figure), alors l'aiguille sera sur une intersection si

$$\frac{l}{2} sin\theta > r$$

.

Si on suppose que r et θ suivent des lois uniformes sur $[0, \frac{d}{2}]$ et $[0, \frac{\pi}{2}]$ respectivement, la probabilité de tomber sur une jointure est alors :

$$P = \frac{1}{\frac{\pi}{2}\frac{d}{2}} \int_0^{\frac{\pi}{2}} \frac{l}{2} sin\theta d\theta = \frac{2l}{\pi d} \tag{8.5}$$

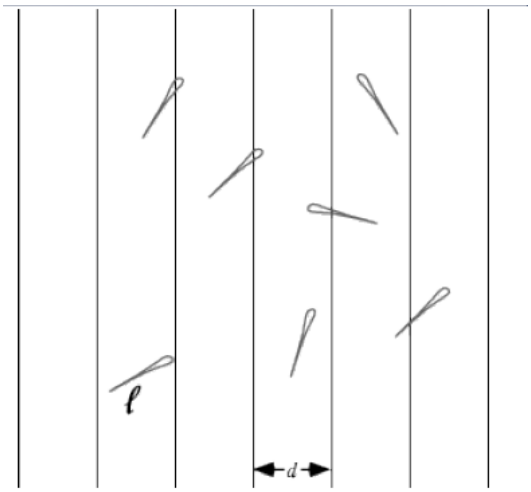


FIGURE 8.1 – Illustration de l'expérience de Buffon

Ainsi, connaissant d et l et en lançant l'aiguille un nombre suffisamment grand de fois (voir loi des grands nombres ci-après), on a avec cette expérience une méthode pour évaluer π.

Une autre méthode pour calculer π consiste à considérer un cercle inscrit dans un carré. Si on place aléatoirement un grand nombre de points dans le carré, certains seront dans le cercle et d'autres en dehors. Comme le rapport des aires de ces deux figures géométriques vaut π, une estimation de π est ainsi donnée par le rapport du nombre de points qui tombent dans le cercle au nombre de points au total.

8.5 Génération de nombres pseudo-aléatoires

Un algorithme déterministe et donc un ordinateur (qui est une machine déterministe par essence) ne peut pas produire de nombres aléatoires. Par contre, un ordinateur peut produire des séquences de nombres arbitraires dont on espère qu'ils ne sont pas corrélés au problème qu'on souhaite résoudre. On parlera ici de **nombres pseudo-aléatoires**.

Une suite pseudo-aléatoire est une suite créée par un algorithme et ayant les propriétés statistiques d'une vraie série aléatoire

$$I_{n+1} = f(I_n)$$

Les nombres aléatoires I_n générés par cette suite sont dans l'intervalle [0,M-1] où M est un très grand nombre entier.
Le nombre qui initialise la suite pseudo-aléatoire , I_0 est appelé ***graine***.
La même graine donne toujours la même suite de nombres, et c'est pour cette raison

qu'il est important d'utiliser un générateur avec une ré-initialisation fréquente de cette valeur.

Définition - Générateurs congruentiels

Un générateur congruentiel génère des nombres aléatoires suivant une distribution uniforme :

$$I_{n+1} = (aI_n + c)mod(b)$$

La suite se répète ici avec la fréquence b.
C'est ce type de générateur de nombres pseudo-aléatoires qui est implanté dans la majorité des langages de programmation.

En Python, on utilisera la fonction random() pour générer une suite de nombres pseudo-aléatoires uniformément distribués sur l'intervalle [0,1[.
Si la graine (random.seed[x]) n'est pas initialisée, l'heure courante est utilisée par défaut comme graine de départ.

Pour des générations successives de suites de nombres pseudo-aléatoires suivant le même algorithme, la graine devra être modifiée à chaque expérience [1].
Vous trouverez une description complète du générateur de nombres pseudo-aléatoires de Python à cette URL :

http ://docs.python.org/3.1/library/random.html

8.6 Transformation d'une loi de probabilité

Nous avons vu que pour générer des nombres pseudo-aléatoires, on peut utiliser le générateur congruentiel implanté dans Python, qui donne des nombres distribués selon une **loi uniforme**.

Si on considère une densité de probabilité normalisée $P(x)$ et la fonction de répartition associée $A(x)$ définies par

$$\int_{-\infty}^{+\infty} P(x)dx = 1 \qquad A(x) = \int_{-\infty}^{x} P(x')dx'$$

pour une loi uniforme, qui décrit des événements équiprobables, la densité de probabilité sur un intervalle $[a, b]$ est égale à l'inverse de la largeur de l'intervalle :

$$P(x) = \frac{1}{b-a}$$

1. Ceci sera fait si on ne définit pas la graine de la fonction random.

L'utilisation de random nous donnera donc des nombres suivant cette densité de probabilité. Cependant, en physique, on aura souvent besoin de générer des nombres qui suivent des lois non-uniformes dont voici quelques exemples :

– Loi normale ou Loi gaussienne

$$P(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}}$$

– Loi de Poisson

$$P(x) = \frac{\lambda^x e^{-\lambda}}{x!}$$

– Loi de Lorentz

$$P(x) = \frac{a}{\pi(a^2 + x^2)}$$

Il faut donc être en mesure de générer une suite de nombres aléatoires distribués suivant des densités de probabilité de loi non-uniforme.
Pour cela, on peut partir d'une loi simple comme la loi uniforme, et appliquer une transformation de loi de probabilité.

8.6.1 Transformation directe d'une loi de probabilité

Soit un ensemble de nombres aléatoires $\{x_i\}$ suivant une loi uniforme sur l'intervalle $[a, b]$, et $\{y_i\}$ leurs images sur ce même intervalle par une fonction bijective f : $\{y_i\} = \{f(x_i)\}$.

La densité de probabilité de la variable aléatoire X, soit $p(X)$ étant connue, on cherche à connaître la densité de probabilité de la variable aléatoire Y, image de X, $g(Y)$.

La fonction f doit conserver la probabilité (car elle est bijective), c-à-d que $p(x)dx = g(y)dy$.
Dans ce cas la loi de transformation de p vers g est donnée par :

$$g(y) = p(x) \left| \frac{1}{dy/dx} \right| \equiv p(f^{-1}(y)) \cdot \left| \frac{df^{-1}(y)}{dy} \right| \tag{8.6}$$

On peut donc transformer une distribution uniforme en n'importe quelle autre distribution sous réserver de savoir calculer dx/dy et f^{-1}.

En pratique, cela s'avère souvent complexe et on utilisera plutôt la fonction de répartition de la variable aléatoire.

8.6.2 Transformation d'une loi de probabilité par la fonction de répartition

Soit une série de nombres aléatoires $\{x_i\}$ suivant une loi uniforme sur l'intervalle $[0, 1]$.

Pour construire une nouvelle série $\{y_i\}$ suivant une loi non–uniforme de densité de probabilité $p(Y)$ à partir de la série connue, on utilise la fonction de répartition associée à la loi $p(Y)$:

$$A(y) = \int_{-\infty}^{y} p(y')dy'$$

On calcule les $y_i = A^{-1}(x_i)$, et la série ainsi générée a la bonne distribution.

Par exemple, si on souhaite générer des nombres suivant une loi de Lorentz

$$p = \frac{1}{\pi(1 + y^2)}$$

la fonction de répartition associée à cette loi vaut :

$$A(y) = \frac{1}{2} + \frac{arctan(y)}{\pi},$$

qui est une fonction qui s'inverse facilement pour donner

$$A^{-1}(x) = tan\left(\pi x - \frac{\pi}{2}\right)$$

On calcule alors les $\{x_i\}$ uniformément distribués et on calcule ensuite les $y_i = A^{-1}(x_i) = tan\left(\pi x_i - \frac{\pi}{2}\right)$, qui seront alors distribués selon une lorentzienne.

8.6.3 Méthode de rejet

Lorsqu'on ne peut pas inverser la fonction de répartition, ce qui est par exemple le cas pour une variable aléatoire de loi normale, on pourra utiliser **la méthode de rejet**.

Principe

On veut simuler des réalisations d'une variable aléatoire X de densité de probabilité associée f, et on suppose qu'il existe une loi de densité g facilement simulable telle que :

$$\forall\ x \in \mathbb{R}^d, \quad f(x) \leq\ k \cdot g(x) \quad \text{avec} \quad k = cte \in \mathbb{R}$$

On pose ensuite

$$r(x) = \frac{f(x)}{kg(x)}$$

Pour (U_1, X_1) un couple de variables aléatoires indépendantes telles que U_1 suit une loi uniforme sur [0,1] et X_1 est de loi g.

- Si $U_1 \leq r(X_1)$, alors on pose $X = X_1$
- Sinon on rejette X_1 et on recommence en générant une suite $(U_n, X_n)_{n\geq 2}$ de variables indépendantes de même loi que (U_1, X_1) jusqu'à atteindre un indice p pour lequel on a nouveau $U_p \leq r(X_p)$. Dans ce cas on pose $X = X_p$.

La variable aléatoire X ainsi formée suit une loi de probabilité de densité f.

8.6.4 Transformation pour la loi normale - Algorithme de Box-Müller

Dans le cas de la loi normale centrée réduite, c-à-d de moyenne nulle et de variance unité :

$$P(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$$

la fonction de répartition est la fonction erf qui ne peut pas être inversée...

$$A(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{t^2}{2}} dt = erf$$

On doit donc trouver un autre moyen pour construire la loi normale à partir d'une loi uniforme, et c'est ce que réalise l'algorithme de Box-Müller :
Soit (X, Y) une paire de variables aléatoires indépendantes suivant une loi normale standard. La densité de probabilité associée à cette paire est égale au produit des densités de probabilités individuelles et on a

$$f(x, y) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \cdot \frac{1}{\sqrt{2\pi}} e^{-y^2/2} = \frac{1}{\sqrt{2\pi}} e^{-(x^2+y^2)/2}$$

La densité de probabilité ainsi obtenue présente une symétrie radiale, ce qui amène naturellement à considérer les variables aléatoires en coordonnées polaires (R, θ) définies par $0 \leq \theta \leq 2\pi$, et $X = R\, cos\theta$ et $Y = R\, sin\theta$.

θ suit une loi uniforme sur l'intervalle $[0, 2\pi]$, et peut donc être échantillonné par une variable aléatoire de loi uniforme U_1 sur [0,1] telle que

$$\theta = 2\pi U_1$$

Alors qu'en coordonnées cartésiennes, la fonction de répartition ne peut pas être inversée, en coordonnées, polaires, la fonction de répartition de R est plus simple à manipuler :

$$G(R) = P(R \leq r) = \int_{r'=0}^{r} \int_{\theta=0}^{2\pi} \frac{1}{2\pi} e^{-r'^2/2} r dr d\theta = \int_{r'=0}^{r} e^{-r'^2/2} r dr$$

En appliquant le changement de variable $r'^2/2 = s$ et $r'dr' = ds$ de sorte que $r' = r$ quand $s = r^2/2$, on obtient

$$G(r) = \int_{s=0}^{r^2/2} e^{-s} dx = 1 - e^{-r^2/2}$$

On peut donc échantillonner R en résolvant l'équation

$$G(R) = 1 - e^{-R^2/2} = 1 - U_2.$$

Ici, $1 - U_2$ suit une loi uniforme standard si U_2 est de loi uniforme standard.

La solution de cette équation est

$$R = \sqrt{-2\ ln(U_2)}$$

Au final, l'algorithme de Box-Müller prend une paire de variables aléatoires indépendantes de loi uniforme standard (définies sur l'intervalle [0,1]) U_1 et U_2, et produit une paire de variables aléatoires indépendantes de loi normale centrée réduite X et Y en utilisant les formules suivantes :

$$\theta = 2\pi U_1, \quad R = \sqrt{-2\ ln(U_2)}, \quad X = R\ cos\theta, \quad Y = R\ sin\theta$$

Ce qui donne encore

$$X = \sqrt{-2\ ln(U_1)}\ cos(2\pi U_2) \tag{8.7}$$
$$Y = \sqrt{-2\ ln(U_1)}\ sin(2\pi U_2) \tag{8.8}$$

8.7 Calcul d'intégrales par la méthode de Monte-Carlo

Outre l'étude des processus stochastiques, les méthodes de Monte-Carlo sont beaucoup utilisées en physique pour calculer des intégrales (multidimensionnelles surtout).

En pratique, on fait un nombre répété d'expériences dont les réalisations suivent une loi connue de probabilité.
On calcule alors la moyenne empirique des résultats de ces expériences ce qui nous donne une estimation de l'espérance de la variable aléatoire représentant les réalisations :

$$\bar{X} = \frac{1}{N} (X_1 + X_2 + \cdots + X_N) \approx E(X)$$

Le calcul d'intégrales étant un calcul d'aire sous une courbe (au sens large), dans l'approche Monte-Carlo, on va faire un tirage aléatoire de couples de points dans un

rectangle défini par les bornes d'intégration (on parle ici du cas unidimensionnel ; plus généralement on fera un tirage aléatoire de n-uplets dans un espace à n dimensions défini par les bornes d'intégrations). La fraction des points tels que $y_i \leq f(x_i)$ donne une estimation du rapport entre l'aire sous la courbe et l'air totale du rectangle, ce qui permet d'obtenir une estimation de l'intégrale.

8.7.1 Méthodologie - cas unidimensionnel

On se propose d'évaluer l'intégrale $I = \int_a^b g(x)dx$, avec g une fonction définie sur $\mathbb{R}$.

Sur l'intervalle $[a, b]$, on définit la densité de probabilité uniforme f par :

$$f(x) = \begin{cases} \frac{1}{b-a} & \text{si } \ a \leq x \leq b \\ 0 & \text{sinon} \end{cases}$$

On peut donc réécrire I en y injectant f :

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

Il apparaît clairement ici que l'intégrale I est donnée par l'espérance de la variable aléatoire $g(x)$ sur l'intervalle $[a, b]$:

$$I = (b-a) \int_a^b g(x)f(x)dx = (b-a)E(g(x)) = (b-a)\bar{g} \tag{8.9}$$

On introduit à présent des tirages de nombres aléatoires, ce qui est la spécificité des méthodes de Monte-Carlo.
Pour cela on se donne N réalisations discrètes $\{x_n\}$ de la variable aléatoire X de loi uniforme sur [a,b] et on évalue $g(x_n)$ pour chacune de ces réalisations.

On peut alors calculer la moyenne des $g(x_n)$:

$$G = \frac{1}{N} \sum_{i=1}^{N} g(x_i) \tag{8.10}$$

en utilisant la propriété des sommes de variables aléatoires selon laquelle l'espérance de la moyenne de N réalisations d'une variable aléatoire $g(x)$ de loi uniforme standard est l'espérance de la variable aléatoire $g(x)$, on a :

$$\bar{G} = \frac{1}{N} \sum_{i=1}^{N} \bar{g} = \bar{g}$$

En approximant G par sa valeur moyenne, on obtient ainsi une estimation de l'intégrale I

$$I = (b-a)\bar{g} = (b-a)\bar{G} \approx (b-a)\frac{1}{N}\sum_{i=1}^{N} g(x_i) \tag{8.11}$$

L'intégrale est évaluée par la valeur moyenne de l'intégrant sur l'intervalle d'intégration.

8.7.2 Méthodologie - cas multidimensionnel

Dans le cas des intégrales à 2, 3 ou plus de dimensions (comme il n'est pas rare d'en rencontrer en physique), le calcul numérique par des méthodes déterministes peut être complexe ou impossible, et lorsqu'il est réalisable, de convergence lente. On utilisera alors préférentiellement une méthode de Monte Carlo pour faire le calcul.

La procédure à suivre est la même que dans le cas unidimensionnel.

Pour intégrer une fonction à m dimensions

$$\int_{a_1}^{b_1}\int_{a_2}^{b_2}...\int_{a_\mathsf{m}}^{b_\mathsf{m}} f(x'_1, x'_2, ..., x'_\mathsf{m})dx'_1dx'_2....dx'_\mathsf{m},$$

on suit les étapes suivantes :

- Prendre n vecteurs $\vec{x_1}, \vec{x_2}, ..., \vec{x_n}$ dans l'hypervolume V $= [a_1, b_1] \times [a_2, b_2] \times ... \times [a_\mathsf{m}, b_\mathsf{m}]$
- Calculer la valeur moyenne de la fonction dans cet hypervolume

$$\bar{f} = \frac{1}{n}\sum_{i=1}^{n} f(\vec{x_i})$$

- Estimer la valeur de l'intégrale en multipliant cette moyenne par l'hypervolume

$$V = \prod_{i=1}^{\mathsf{m}}(b_i - a_i)$$

- Estimer l'erreur commise par la relation :

$$\varepsilon = \prod_{i=1}^{\mathsf{m}}(b_i - a_i)\sqrt{\frac{\bar{f}^2 - (\bar{f})^2}{n}}$$

8.7.3 Convergence

Pour que l'approximation soit acceptable, il faut faire un nombre important de tirages aléatoires. Ceci est une conséquence de la ***loi forte des grands nombres.***

Théorème

Soit $(X_i, i \geq 1)$ une suite de variables aléatoires discrètes indépendantes suivant la même loi qu'une variable aléatoire X, où on suppose que l'espérance de X est bornée, c-à-d, $E(x) < +\infty$, alors *pour presque tout u dans l'intervalle de départ de* X on a :

$$E(X) = \lim_{n \to +\infty} \frac{1}{n} \left(X_1(u) + X_2(u) + \cdots + X_n(u) \right) \tag{8.12}$$

Si $x_1, x_2, \cdots, x_n$ sont des nombres tirés au hasard uniformément dans]a,b[, alors

$$\frac{1}{n} \sum_{i=1}^{n} g(x_i) \text{ est une approximation de } \frac{I}{b-a}.$$

8.7.4 Vitesse de convergence

La vitesse à laquelle la loi des grands nombres converge est déterminée par le ***théorème de la limite centrale*** :

Théorème

Soit $(X_i, i \geq 1)$, une suite de variables aléatoires discrètes indépendantes suivant la même loi qu'une variable aléatoire X.
On suppose que $E(X^2) < +\infty$, et on note σ^2 la variance de X

$$\sigma^2 = E(X^2) - E(X)^2 = E((X - E(X))^2) \tag{8.13}$$

Si on définit l'erreur

$$\varepsilon_n = E(X) - \frac{1}{n} \sum_{i=1}^{n} X_i$$

alors

$$\frac{\sqrt{n}}{\sigma} \varepsilon_n = \frac{\sqrt{n}}{\sigma} \left(E(X) - \frac{1}{n} \sum_{i=1}^{n} X_i \right) = \frac{1}{\sigma \sqrt{n}} \sum_{i=1}^{n} (X_i - E(X_i)) \tag{8.14}$$

qui converge en loi vers une gaussienne centrée réduite :

$$\forall a < b, \quad \lim_{n \to +\infty} P\left(\frac{\sigma}{\sqrt{n}} a \leq \varepsilon_b \leq \frac{\sigma}{\sqrt{n}} b \right) = \frac{1}{\sqrt{2\pi}} \int_a^b e^{-\frac{x^2}{2}} dx \tag{8.15}$$

L'écart-type de l'erreur vaut $\frac{\sigma}{\sqrt{n}}$ et la méthode converge en $\frac{1}{\sqrt{n}}$.

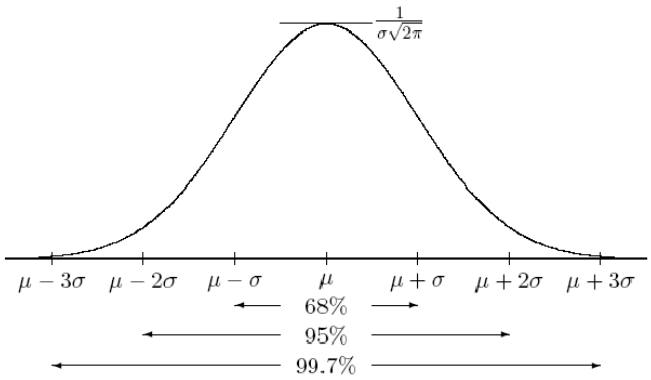


Fig. 1.4. – La densité de la loi gaussienne centrée réduite avec quelques indices de concentration.Noter ce qui se passe lorsque σ diminue.

8.8 Estimation de l'erreur

Nous avons vu que l'estimation de l'erreur est une étape fondamentale d'une méthode de Monte-Carlo.
En partant des deux théorèmes énoncés ci-dessus on peut estimer l'erreur dans le cas d'un nombre important de tirages aléatoires.

Pour une série de variables aléatoires x_i de même loi que la variable aléatoire X, on a vu que

$$I_n = \bar{X} = \frac{1}{N}\sum_{i=1}^{N} x_i$$

.

Le théorème de la limite centrale implique alors que

$$\lim_{n\to+\infty} P\left(\|\bar{X} - E(X)\| \leq \alpha\frac{\sigma}{\sqrt{n}}\right) = \frac{1}{\sqrt{2\pi}}\int_{-\alpha}^{+\alpha} e^{-\frac{x^2}{2}}dx \tag{8.16}$$

On choisit α de façon à avoir :

$$\frac{1}{\sqrt{2\pi}}\int_{-\alpha}^{+\alpha} e^{-\frac{x^2}{2}}dx = p \text{ proche de 1}$$

Les valeurs classiques de p sont p = 0.95 (correspondant à $\alpha = 1.96$) et $p = 0.99$ (correspondant à $\alpha = 2.6$).

Dans la pratique on se place dans le cas où un grand nombre de tirages aléatoires nous met à la limite, et on affirme que $P(|\varepsilon_n| \leq 1.96\frac{\sigma}{\sqrt{n}})$ est de l'ordre de 0.95. On

a donc avec une probabilité proche de 0.95 que :

$$E(X) \in \left[\frac{1}{n}(x_1 + x_2 + ... + x_n) - 1.96\frac{\sigma}{\sqrt{n}}, \frac{1}{n}(x_1 + x_2 + ... + x_n) + 1.96\frac{\sigma}{\sqrt{n}}\right]. \tag{8.17}$$

Cela signifie que la probabilité pour que la variable aléatoire $\frac{\varepsilon_n\sqrt{n}}{\sigma}$ soit inférieure à 1.96 est voisine de 95%, donc que l'erreur ε_n est inférieure à $1.96\frac{\sigma}{\sqrt{n}}$ avec une quasi-certitude de 95%.

Ainsi, si on tire aléatoirement n nombres suivant la loi de X, on peut estimer la moyenne (espérance) E par la moyenne empirique $\bar{x}$ par

$$\bar{x} = \frac{1}{n}(x_1 + x_2 + \cdots + x_n).$$

Le théorème de la limite centrale nous dit alors que :

$$P\left(E \in \left[\bar{x} - 1.96\frac{\sigma}{\sqrt{n}}, \bar{x} + 1.96\frac{\sigma}{\sqrt{n}}\right]\right) \approx 0.95 \tag{8.18}$$

L'écart-type σ n'est pas connu explicitement lorsqu'on utilise une méthode de Monte-Carlo, et on doit donc trouver une méthode pour l'estimer. En général on posera :

$$\bar{\sigma_n}^2 = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \bar{x})^2 = \frac{1}{n-1}\sum_{i=1}^{n}x_i^2 - \frac{n}{n-1}\bar{x}^2 \tag{8.19}$$

$\bar{\sigma_n}^2$ est ce que l'on appelle un estimateur sans biais convergent de σ^2, ce qui signifie que $E(\bar{\sigma_n}^2) = \sigma^2$ et que $lim_{n\to+\infty}\bar{\sigma_n}^2 = \sigma^2$.

Au cours de la simulation Monte-Carlo, en plus de stocker la somme des x_i, on stockera donc la somme $\sum_i x_i^2$ de leurs carrés pour pouvoir calculer $\bar{\sigma_n}^2$ à l'aide de l'équation ci-dessus. En remplaçant ensuite σ par $\bar{\sigma_n}$ dans les extrémités de l'intervalle dans (8.18), on pourra affirmer que :

$$P\left(E \in \left[\bar{x} - 1.96\frac{\bar{\sigma_n}}{\sqrt{n}}, \bar{x} + 1.96\frac{\bar{\sigma_n}}{\sqrt{n}}\right]\right) \approx 0.95$$

8.9 Réduction de la variance - accélération de la convergence

L'erreur obtenue sur le résultat d'une simulation de type Monte-Carlo est liée à la variance de la variable aléatoire considérée comme on vient de le montrer. Il en va de même pour la vitesse de convergence de la méthode qui est en $\frac{\sigma}{\sqrt{n}}$.

Pour améliorer la méthode, on cherchera donc à réduire l'erreur, ce qui revient à réduire la variance de la variable aléatoire.
Il existe pour cela de nombreuses techniques parmi lesquelles on trouve :
- l'échantillonnage préférentiel
- l'utilisation d'une variable de contrôle

8.9.1 Échantillonage préférentiel

On suppose qu'on cherche à calculer l'espérance de la variable aléatoire $g(X)$, pour une variable aléatoire X de loi $f(x)dx$ sur $\mathbb{R}$.

$$E(g(X)) = \int_{\mathbb{R}} g(x)f(x)dx$$

que l'on peut toujours réécrire :

$$E(g(X)) = \int_{\mathbb{R}} \frac{g(x)f(x)}{h(x)}h(x)dx$$

avec h une autre densité de probabilité sur $\mathbb{R}$.

Si Y est une variable aléatoire de loi $h(y)dy$ dans $\mathbb{R}$, alors on a une nouvelle expression pour l'espérance de $g(x)$:

$$E(g(X)) = E\left(\frac{g(Y)f(Y)}{h(Y)}\right) = E(Z)$$

On aura alors amélioré l'algorithme si $V(Z) < V(g(X))$

$$V(Z) = E(Z^2) - E(Z)^2 = \int_{\mathbb{R}} \frac{g^2(x)f^2(x)}{h^2(x)}dx - E(g(X))^2 \tag{8.20}$$

8.9.2 Variable de contrôle

Dans le cas de l'utilisation d'une variable de contrôle, on réécrira $E(f(X))$ sous la forme

$$E(f(X)) = E\left(f(X) - h(X)\right) + E(h(X))$$

où on a utilisé les propriétés de l'espérance, et où $E(h(X))$ peut se calculer analytiquement et $V\left(f(X) - h(X)\right)$ est sensiblement plus petit que $V(f(X))$.

On utilise ensuite une méthode de Monte-Carlo pour évaluer $E\left(f(X) - h(X)\right)$, et le calcul direct pour $E(h(X))$.