Table des matières

Chapitre 1

ERREURS ET STABILITÉ

Les méthodes de l'analyse numérique visent à traiter numériquement un certain nombre de problème. Ce traitement se fait maintenant essentielllement en utilisant un ordinateur.

Evidement, tous les nombres ne peuvent pas être représentés en machine. Une annexe à ce chapitre rappelle quelques principes de la représentation des nombres en machine. On notera génériquement $\mathcal{F}$ l'ensemble des nombres réels accessibles par la machine que l'on utilise. L'application qui envoie les réels représentables dans $\mathcal{F}$ s'appelle application d'arrondi et se note :

$$x \longrightarrow \mathrm{fl}(x) = m_x \, b^{e_x} \in \mathcal{F}. \tag{1.1}$$

b désigne la base de la représentation, et en général $b = 2$.

m_x s'appelle la mantisse ; sauf pour les nombres très petits, elle est normalisée de façon que $1 \leq m < b$ et s'écrit avec p chiffres dans la base b.

e_x désigne l'exposant de cette représentation ; il est compris entre les deux valeurs $e_{\min}$ et $e_{\max}$.

1.1 Erreurs

Les erreurs sont présentes un peu partout dans la vraie vie : la baguette de pain que vous achetez chez le boulanger pèse-t-elle exactement 250 grammes, l'image que vous voyez sur votre récepteur de télévision n'est-elle pas un peu floue, le professeur qui a noté votre copie l'a-t-il bien évaluée ...

Certaines erreurs sont admises par le bon sens. On se doute par exemple qu'une quantité exprimée en millions ou en milliards est souvent une valeur approchée. Le philosophe des sciences Michel Serre raconte l'anecdote suivante :

En visite au Museum d'histoire naturelle, alors en travaux, et devant un squelette fossile dont la fiche de présentation a disparu, j'avise un gardien et lui demande si on en connait l'age :

-" Oh je peux vous le dire ; il a 180 millions d'annéees et 11 mois ! "

-" Ah bon ! Et pourquoi ces 11 mois ? "

-" Et bien quand je suis arrivé dans ce service, en septembre dernier, la fiche était encore là, et je me rappelle qu'elle indiquait 180 millions d'années ; c'était il y a 11 mois ! "

L'anecdote rapportée nous fait sourire car on n'imagine pas que l'age de ce fossile ait pu être déterminé avec une précision qui rende significatif cet écart de 11 mois ... mais par ailleurs, pour des enfants qui commencent leur scolarité un écart de 11 mois entre ceux qui sont nés en janvier et ceux qui sont nés en décembre d'une même année est très significatif !

1.1.1 Erreurs relatives et erreurs absolues

Définitions

Si $\tilde{x}$ est la valeur approchée d'une quantité x, l'**erreur absolue** sur cette estimation est $\Delta x = |\tilde{x} - x|$.

L'**erreur relative** est $\delta x = \left|\dfrac{\tilde{x} - x}{x}\right|$.

Ainsi, si l'erreur relative est de 10^{-8}, on peut penser que les 8 premiers chiffres du nombre $\tilde{x}$ seront corrects ; on ne peut rien dire de tel au vu d'une erreur absolue.

Exemple

La formule de Taylor appliquée à e^x s'écrit

$$e^x = \sum_{k=0}^{n} \frac{x^k}{k\,!} + \frac{e^c}{(n+1)\,!}\, x^{n+1} = s_n(x) + r_n(x) \tag{1.2}$$

pour un c compris entre 0 et x. Mathématiquement, on peut montrer que le reste $r_n(x) = \dfrac{e^c}{(n+1)!}\, x^{n+1}$ tend vers 0 lorsque n tend vers l'infini ; on est en droit de penser qu'on évaluera bien e^x en négligeant ce reste si on choisit n suffisament grand.

La figure 3.9 nous montre les erreurs absolues et relatives obtenues pour $1 \leq n \leq 100$ dans les évaluations de e^{20} et e^{-20}. Ces erreurs sont représentées par une courbe qui utilise une échelle logarithmique pour les ordonnées.

On constate qu'à partir de $n = 70$, les erreurs sont à peu près constantes. Les erreurs absolues sont comparables : pour $x = 20$ on a $\Delta = 5.9605\ 10^{-8}$, et pour $x = -20$, $\Delta = 3.5607\ 10^{-9}$.

Mais c'est pourtant l'estimation de e^{20} qui est la meilleure comme le montre la courbe représentant l'évolution des erreurs relatives, où l'on constate que pour $x = -20$, l'erreur relative reste de l'ordre de 1 (en fait $\delta \simeq 1.7275$), alors que pour $x = 20$, elle vaut à peu près $1.2285\ 10^{-16}$, ce qui correspond à la précision de l'arithmétique utilisée. Ainsi, on a calculé $s_{70}(-20) = 5.8390\ 10^{-9}$, alors que $e^{-20} \simeq 2.0612\ 10^{-9}$; dans ce cas l'erreur est plus importante que la quantité que l'on voulait calculer, et aucun chiffre du nombre estimé n'est correct.

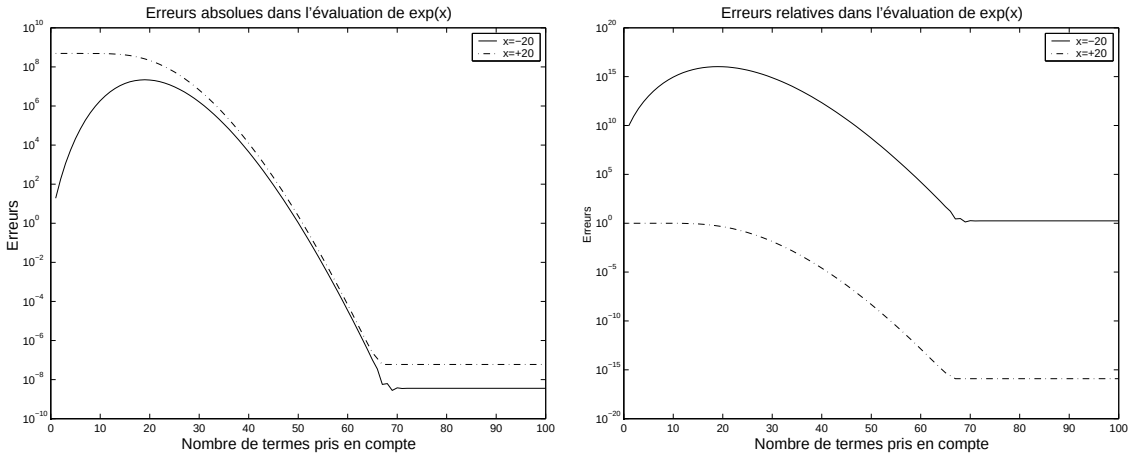


FIGURE 1.1 – Une bonne erreur absolue n'est pas toujours une bonne erreur relative

Pour $x = 20$, on a calculé $s_{70}(20) = 4.851651954097902\ 10^8$ alors que $e^{20} \simeq 4.851651954097903\ 10^8$; on a 15 chiffres corrects dans cette estimation, ce qui est conforme à la valeur de l'erreur relative $\delta = 1.2285\ 10^{-16}$

On retrouvera une bonne approximation de e^{-20} en prennant l'inverse de la valeur calculée pour $x = 20$: $\dfrac{1}{s_{70}(20)}$ fournit cette valeur approchée avec une précision relative de $2\ 10^{-16}$.

1.1.2 Erreurs d'arrondi

Si l'on regarde attentivement les résultats du calcul approché des exponentielles, on constate également que contrairement à ce qu'annoncent les mathématiques, la suite des approximations calculées ne converge pas vers la valeur attendue. A partir d'une certaine valeur de l'entier n, l'erreur ne décroit plus ! Ce phénomène est dû aux erreurs d'arrondi . Les erreurs d'arrondi sont omniprésentes dans les calculs sur ordinateur, avec des effets le plus souvent négligeables, mais quelquefois beaucoup plus importants.

Un exemple anodin

Pour mieux comprendre ce phénomène, essayons de nous rappeler le 1er janvier 2002 en reprenant un exemple de Joceline Erhel ; ce matin là, un client au porte-monnaie plein de pièces de francs français se présente à sa boulangerie pour acheter une baguette :

- *Bonjour, c'est combien la baguette aujourd'hui ?*
- *Oh, je n'ai pas augmenté, c'est toujours* 4.30 *F, mais ajourd'hui, c'est en euros ; voyez avec mon euro-calculette, ça fait* 0.66 € ...

- *Attendez, je vérifie ; j'ai aussi une euro-calculette ! Ah mais,* 0.66 €, *ça fait* 4.33 *F ! Bon, c'est pas grave, mais je n'ai pas d'euros ! je peux vous payer avec une pièce de* 5 *F ?*
- *Oui, bien sûr, mais je vous rend la monnaie en euros... Alors,* 5 *F, ça fait* 0.76 €, *donc je vous rend* 0.10 € *et le compte est bon.*
- *Le compte est bon... c'est vite dit ! Regardez ma calculette :* 0.10 €, *ça fait* 0.66 *F, et* 5 *F moins* 0.66, *ça me fait maintenant la baguette à* 4.34 *F !*
- *Au fait, avec tout cela, j'ai failli oublier qu'il me faut aussi une baguette pour ma voisine, alors j'en prend 2. Je vous dois donc* 8.60 *F, et ça fait* 1.31 € *: Eh, j'y gagne un peu. Je vous paie avec une pièce de* 10 *F.*
- *Comme vous voulez ; voyons...* 10 *F, ça fait* 1.52 €, *et je vous rends* $1.52 - 1.31 = 0.21$ €.
- *Finalement, je limite les dégats : les* 0.21 € *que vous me rendez, ça fait* 1.38 *F, ce qui me met la baguette à* $\frac{10 - 1.38}{2} = 4.31$ *F !*

Cette euro-calculette fonctionnait en arrondissant au plus près avec 2 chiffres après la virgule (le point !).

En prenant la valeur de l'euro avec 5 décimale,soit 1€ = 6.55957 F, on constate que toutes les valeurs calculées sont conforme à la stratégie d'arrondi au plus près :

- 4.30 F = 0.65553 € est correctement arrondi à 0.66 €.
- 0.66 € = 4.32932 F est correctement arrondi à 4.33 F,
- 5F- 4.30F = 0.70 F, mais 0.76225€ - 0.65553 € = 0.106720, correctement arrondi à 0.10 € font 0.65596 F, correctement arrondi à 0.66 F, ce qui donne le second prix de baguette à 4.34 F.
- Par contre, 0.106720€ font 0.70003 F que l'on arrondirait à 0.70 F pour retrouver la somme correcte !

On voit que les erreurs d'arrondi peuvent s'accumuler ou se compenser et qu'elles dépendent de la précision de l'arithmétique utilisée. Ici, les valeurs sont conformes à la stratégie d'arrondi au plus près, et les erreurs restent de l'ordre de la précision de l'arithmétique, c'est à dire de l'ordre du centime d'euro.

Un exemple aux conséquences plus graves

Les erreurs d'arrondi peuvent avoir des conséquences beaucoup plus grave : le 25 février 1991, pendant la guerre du Golfe, un missile "Patriot" de l'armée américaine a manqué le missile SCUD de l'armée irakienne qu'il devait intercepter. Le missile SCUD atteignit un batiment de l'armée américaine, tuant 28 personnes et en blessant une centaine. Tout cela était dû à une erreur d'arrondi.

Le système de guidage des missiles "Patriots" était implanté dans une arithmétique à virgule fixe en base 2 avec 24 bits. L'horloge de la batterie de "Patriot" était échelonnée en dixièmes de secondes, et le nombre de dixièmes de secondes était multiplié par 0.10 pour obtenir le temps en secondes. Dans cette arithmétique le

nombre $\frac{1}{10}$ s'écrit 0.00011001100110011001100 : l'erreur ϵ peut s'évaluer facilement en utilisant une des expressions (1.18) ou (1.19) :

$$\epsilon = \left(\frac{1}{2^4} + \frac{1}{2^5}\right) \sum_{k=5}^{\infty} \frac{1}{16^k} \simeq 9.5367\ 10^{-8}$$

La batterie était en fonction depuis une centaine d'heures. En multipliant ϵ par le nombre de dixièmes de secondes correspondant, on obtient une erreur sur le temps d'environ

$$9.5367\ 10^{-8} \times 100 \times 60 \times 60 \times 10 \simeq 0.3433 \text{ s}$$

Un missile SCUD avance à la vitesse de 1 676 m/s, et parcourt donc environ 575 m pendant ce temps. Cette distance suffit à le faire manquer par le "Patriot" dont la fénêtre de visée de 500 m est calculée en fonction de cette vitesse et de l'heure précise de la dernière détection par un radar.

1.1.3 Opérations arithmétiques élémentaires

Perte des propriétés algébriques

En général, un arrondissage est effectué à chaque affectation d'une valeur à une variable. Par exemple, si on programme `x = 0.1 ;` la valeur de la variable `x` ne sera pas exactement 0.1, puisque son écriture en base 2 se fait au moyen d'une infinité de chiffres. On note fl(0.1) cette valeur.

L'application d'arrondi (1.1) se fait généralement suivant la stratégie d'arrondi au plus près. En raison de la normalisation de la mantisse, elle garantit une **précision relative** constante. Un note u et on appelle unité d'arrondi le maximum de cette erreur relative :

$$u = \max\left\{\left|\frac{x - \text{fl}(x)}{x}\right|,\ x \text{ représentable}\right\} = \frac{1}{2}\, b^{1-p}. \tag{1.3}$$

Mais, $\mathcal{F}$ n'est pas stable pour les opérations usuelles, de sorte qu'un nouvel arrondissage sera effectué après chaque opération. Si on programme `y = x + x ;` la valeur de la variable `y` sera fl(fl(0.1) + fl(0.1)). Les erreurs d'arrondi vont s'accumuler.

Certaines des propriétes usuelles de opérations élémentaires ne sont pas conservées : associativité de l'addition, distributivité de la multiplication par rapport à l'addition

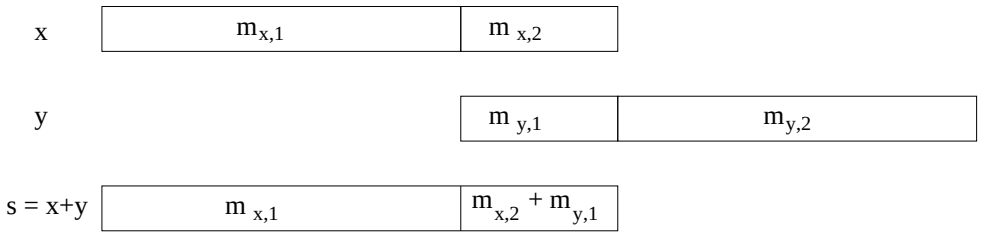
Addition flottante et mécanisme d'absorption

Soient x et y, de même signe, par exemple positif, représentés en machine par $\text{fl}(x) = m_x b^{e_x}$ et $\text{fl}(y) = m_y b^{e_y}$; on supose que $e_x > e_y$ et on note $n = e_x - e_y$.

On a donc $\text{fl}(y) = m_y b^{-n}\, b^n b^{e_y}$ et on peut étudier l'addition en introduisant $\overline{m_y} = b^{-n} m_y$ de sorte que :

$$\mathrm{fl}(x) + \mathrm{fl}(y) = b^{e_x}\,(m_x + \overline{m_y})$$

$m_x + \overline{m_y}$ a $p+n$ chiffres, donc n'est pas une mantisse, et sera arrondie, éventuellement après une renormalisation. Les derniers chiffres de la mantisse m_y seront perdus, et lorsque n est voisin de p, c'est presque tous les chiffres de m_y qui sont ainsi "absorbés" !



Lorsque $n \geq p$, on devine sur la figure ci-dessus que $m_{y,1}$ est vide, de sorte que $\mathrm{fl}(x+y) = x$.

Un exemple :

On sait que la série harmonique, série numérique de terme géneral $\dfrac{1}{n}$, est divergente. Son que son comportement asymptotique est donné par :

$$\left(\sum_{n=1}^{N} \frac{1}{n}\right) - \ln N \sim \gamma \quad (N \to \infty)$$

où γ désigne la constante d'Euler, $\gamma \simeq 0.5772\ 1566\ 490$.

En arithmétique flottante, cette série finit par converger, et la somme calculée dépend de l'ordre des calculs !

Les sommes partielles $S_N = \displaystyle\sum_{n=1}^{N} \frac{1}{n}$ de cette série finissent, pour N assez grand, par absorber totalement les termes suivants ; on peut limiter les dégats en sommant selon les indices décroissants, mais ça n'est pas encore très satisfaisant.

Voici les résultats obtenus pour quelques valeurs de N sur une station DEC-alpha :

N	indices croissants	indices décroissants
10^3	7.4854 7172 546	7.48547840 118
10^4	9.7876 0433 197	9.7876 1291 503
10^5	12.0901 5274 04	12.0908 5083 00
10^6	14.3573 5797 88	14.3926 5155 79
10^7	15.4036 8270 87	16.6860 3134 15
10^8	15.4036 8270 87	18.8079 1854 85
10^9	15.4036 8270 87	18.8079 1854 85

En fait, il existe des procédés de calculs qui permettent de suivre à peu près le comportement théorique.

Soustraction flottante et erreur d'élimination

En reprennant les notations précédentes avec x et y positifs et $x > y$, on obtient :

$$\mathrm{fl}(x) - \mathrm{fl}(y) = b^{e_x} (m_x - \overline{m_y})$$

Cette fois, lorsque $x - y$ est très petit, on rencontre un autre type de problème : c'est l'élimination des chiffres significatifs de m_x et m_y qui risque de se produire. En effet, dans ce cas, $n = 0$ et les k premiers chiffres des mantisse de x et y sont égaux ; la soustraction $m_x - m_y$ fait apparaitre des zéros dans les k premières positions ; la renormalisation va consister à décaler vers la gauche de k positions les chiffres de $m_x - m_y$, et à remplir les k dernières positions avec des 0 arbitraires.

Exemple

On considère la fonction $f(x) = \dfrac{1 - \cos x}{x^2}$, que l'on souhaite évaluer pour $x = 1.2 \times 10^{-5}$. On utilise la valeur approchée

$$\cos x \simeq c = 0.9999\,9999\,99,$$

dont tous les chiffres sont corrects ; les calculs fournissent alors :

$$\begin{aligned} 1 - c &= 10^{-10} \\ \frac{1-c}{x^2} &= \frac{10^{-10}}{1.44 \times 10^{-10}} = 0.6944 \end{aligned}$$

Ce résultat est clairement faux, puisque le résultat doit être au plus 0.5 : dans ce calcul, le résultat de la soustraction $1 - c = 10^{-10}$, qui est exact, est du même ordre

de grandeur que l'erreur de départ $|c - \cos x|$. Cette soustraction donne du poids à l'erreur de départ.

On obtient un résultat plus correct avec $c = 0.9999\ 9999\ 9928$.

Cependant, si on suppose que l'on ne peut utiliser que 10 décimales, on écrira :

$$\cos x = 1 - 2\sin^2\left(\frac{x}{2}\right)$$

$$f(x) = \frac{1}{2}\left(\frac{\sin\left(\frac{x}{2}\right)}{\frac{x}{2}}\right)^2$$

Avec $\sin\left(\frac{x}{2}\right) \simeq s = 0.5999\ 9999\ 99 \times 10^{-6}$, on obtient $f(x) = 0.4999\,9999\,98$, où six chiffres sont corrects.

A chaque étape du calcul, les quantités manipulées sont relativement grandes par rapport à l'erreur initiale.

Elimination globale

Lorsqu'une élimination se produit après une absorption, les chiffres de la mantisse résultante ne représentent plus rien. L'erreur absolue n'est généralement pas très importante, mais l'erreur relative peut l'être.

Si le résultat est amplifié par une multiplication ou une division, le résultat est catastrophique.

Pour illustrer ce phénomène, on considère la fonction $y(x) = \dfrac{(1+x)-1}{x}$

On a calculé les valeurs de cette fonction pour 400 valeurs de x réparties uniformément sur l'intervalle $[10^{-16}, 10^{-14}]$, en respectant le parenthèsage. La figure 3.6 représente la courbe passant par toutes les valeurs calculées.

Les valeurs oscillent de plus en plus autour de la valeur mathématiquement correcte 1, avant de se fixer à 0.

Remarque 1.1 *En choisissant pour valeurs de x les puissances croissantes de $\dfrac{1}{2}$, on n'obtient que des* 1*, puis des* 0 *: on dispose ainsi d'un moyen de compter le nombre de bits réservés à la mantisse.*

C'est aussi un moyen pour évaluer l'unité d'arrondi de l'aritmétique utilisée.

En fait, sachant que la base utilisée est la base 2, on calculera plus simplement u en implantant la boucle :

$u \longleftarrow 1$;
$y \longleftarrow 1 + u$;
Tant que $y \neq 1$,
$\quad u \longleftarrow \dfrac{u}{2}$;
$\quad y \longleftarrow 1 + u$; ;

C'est un travail que vous aurez à réaliser en travaux pratiques.

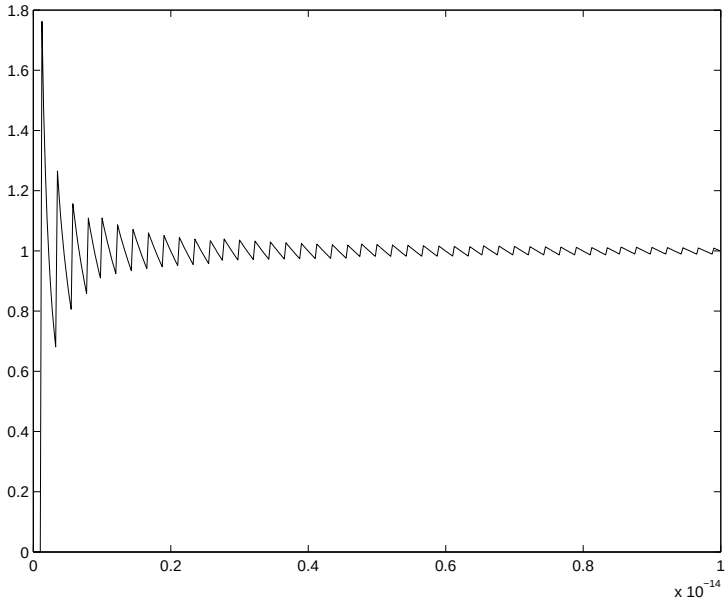


FIGURE 1.2 – Elimination catastrophique

1.1.4 Erreurs de discrétisation

Dans le domaine du calcul scientifique, on ne manipule que des ensembles finis de nombres : on est donc parfois amenés à remplacer certaines opérations **continues** par une approche **discrète**.

Calcul numérique d'une dérivée

Soit $f(x)$ une fonction dont on veut calculer la dérivée. On est souvent obligé d'avoir recours à un procédé numérique, par exemple si f n'est connue que par des mesures physiques, ou encore si elle n'est pas connue explicitement : elle peut être solution d'une équation différentielle compliquée et ne pas s'exprimer à l'aide de fonctions usuelles.

Si l'on sait que f est assez régulière, la formule de Taylor fournit, pour un $\eta \in [x,\ x+h]$:

$$f(x+h) = f(x) + h\, f'(x) + \frac{f''(\eta)}{2}\, h^2. \tag{1.4}$$

Les quantités

$$D_h(x) = \frac{f(x+h) - f(x)}{h} = f'(x) + f''(\eta)\, \frac{h}{2} \tag{1.5}$$

fournissent une approximation de $f'(x)$ dont la qualité augmente lorsque h décroît. $D_h(x)$ s'appelle une **différence finie**.

Prenons l'exemple simple de la fonction $f(x) = \sin(2\,x)$, et calculons par cette méthode des estimations de sa dérivée au point $x = 1$, avec un pas h variant de 10^{-1} à 10^{-16}. Cette dérivée est connue ; elle vaut $f'(1) = 2\,\cos(2) \simeq -0.83229\ 3673$; on peut donc estimer l'**erreur de discrétisation** commise.

Le tableau ci-dessous donne les valeurs de cette erreur. On constate que contrairement à ce que peut laisser espérer la formule (1.5), cette erreur ne décroît avec h ! Apparemment, la meilleure valeur est obtenue pour $h = 10^{-9}$.

h	Erreurs absolues
10^{-1}	$1.7571655696663135\ 10^{-1}$
10^{-2}	$1.8129857211159339\ 10^{-2}$
10^{-3}	$1.8180393850947274\ 10^{-3}$
10^{-4}	$1.8185393630021363\ 10^{-4}$
10^{-5}	$1.8185894958389071\ 10^{-5}$
10^{-6}	$1.8185430400441405\ 10^{-6}$
10^{-7}	$1.8296248016635985\ 10^{-7}$
10^{-8}	$1.6429026472586372\ 10^{-7}$
10^{-9}	$1.1634909868885046\ 10^{-7}$
10^{-10}	$5.6043830853891308\ 10^{-7}$
10^{-11}	$3.8804537900727354\ 10^{-6}$
10^{-12}	$4.0528467195044549\ 10^{-5}$
10^{-13}	$7.3662765004256503\ 10^{-4}$
10^{-14}	$3.7359537458259151\ 10^{-4}$
10^{-15}	$5.5884746605840308\ 10^{-2}$
10^{-16}	$8.3229367309428481\ 10^{-1}$

Explication

Si on admet que les valeurs du sinus sont fournies avec une erreur de l'ordre de la précision de l'arithmétique utilisée (ici la double précision IEEE, de sorte que $u \simeq 1.1102\ 10^{-16}$), alors la majoration de l'erreur $|D_h(1) - f'(1)| \leq \dfrac{h}{2}\,|f"(\eta)|$, doit être remplacée par

$$|D_h(1) - f'(1)| \leq \frac{h}{2}\,|f"(\eta)| + \frac{2\,u}{h} \simeq \frac{h}{2}\,|f"(1)| + \frac{2\,u}{h}. \tag{1.6}$$

η est compris entre 1 et $1+h$, et on peut penser que $f"(\eta)$ varie peu. La fonction $e(h) = \dfrac{h}{2}\,|f"(1)| + \dfrac{2\,u}{h}$ a pour dérivée $e'(h) = \dfrac{1}{2}\,|f"(1)| - \dfrac{2\,u}{h^2}$. Cette dérivée est négative si $h^2 < \dfrac{4\,u}{|f"(1)|}$, c'est-à-dire si $h < 2\sqrt{\dfrac{u}{|f"(1)|}}$. C'est donc pour cette valeur de h qu'elle atteindra son minimum. Avec $f"(1) = -4\,\sin(2) \simeq -3.6372$, on

peut avoir une idée de cette valeur, $h \simeq 1.1\,10^{-8}$, qui est conforme à ce qu'on a observé.

La figure 1.3 montre la courbe représentant $e(h)$ ainsi que les erreurs effectivement observées, avec une échelle logarithmique pour les axes. On y constate que notre majoration est assez bonne.

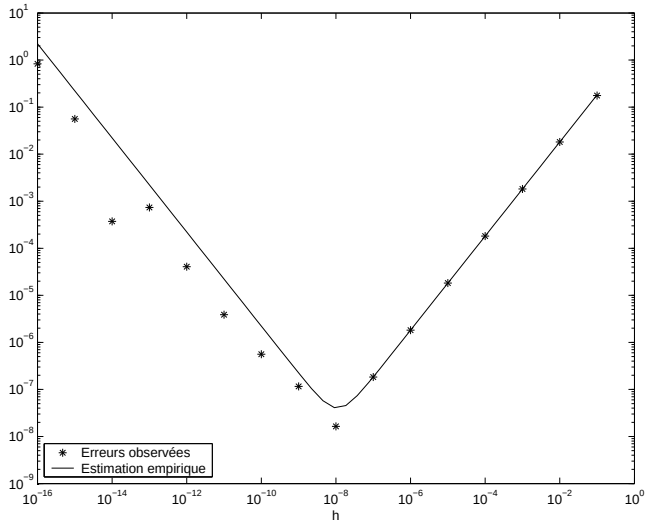


FIGURE 1.3 – Calcul approché de la dérivée, et estimation empirique de l'erreur.

L'échelle logarithmique nous donne la forme approximative d'un V pour la courbe représentant $e(h)$:

- la branche descendante représente essentiellement le terme d'erreur de discrétisation de (1.6), $\frac{h}{2}\,|f"(\eta)|$,
- la branche ascendante représente essentiellement le terme d'erreur d'arrondi, $\frac{2\,u}{h}$.

Augmentation de la précision théorique

L'erreur de discrétisation s'appelle aussi erreur de **consistance**. La notion de consistance est plus générale que celle de discrétisation. Elle s'utilise pour caractériser la distance entre un problème et son approximation : ici, c'est la distance entre $f'(x)$ et $D_h(x)$. Une formule sera donc plus ou moins consistante si l'erreur de consistance est plus ou moins importante en fonction de h. On mesure cette consistance avec la notion d'**ordre** d'une formule. La formule (1.5) est d'ordre 1, car la valeur de la dérivée est fournie avec une erreur qui est $\mathcal{O}(h^1)$.

En utilisant des développements de $f(x+h)$ et $f(x-h)$ à l'ordre 3, on obtient une formule d'ordre 2 :

$$f'(x) = \frac{f(x+h) - f(x-h)}{2\,h} + f'''(x)\,\frac{h^2}{6} + o(h^2). \tag{1.7}$$

En utilisant des développements de $f(x+2\,h)$, $f(x+h)$, $f(x-h)$ et $f(x-2\,h)$ à l'ordre 5, on obtient une formule d'ordre 4 :

$$f'(x) \simeq \frac{-f(x+2h)+8f(x+h)-8f(x-h)+f(x-2h)}{12\,h} + f^{(5)}(x)\,\frac{h^4}{30}. \qquad (1.8)$$

Nous pouvons reprendre le principe de l'étude précédente, et proposer un calcul effectif et une estimation empirique des erreurs commises en utilisant les 3 formules. La figure 1.4 nous montre les résultats de ces trois formules pour le calcul de la dérivée au point $x = 1$ de la fonction $f(x) = 2\sqrt{x}$.

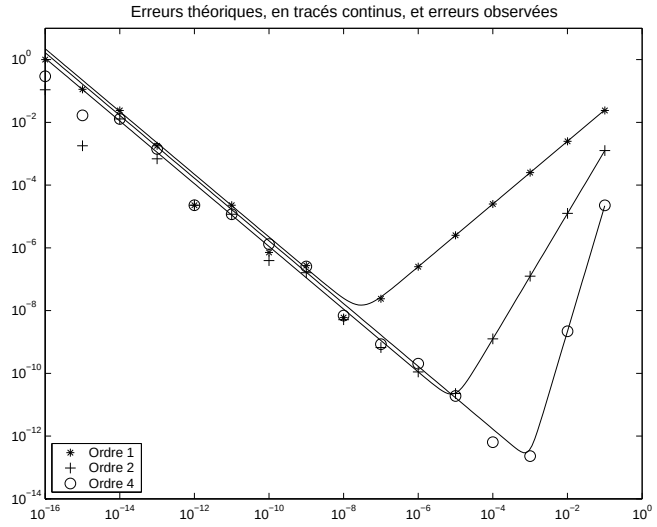


FIGURE 1.4 – Comportement de l'erreur pour les trois formules.

Dans chaque cas, les erreurs observées sont constituées d'un terme de consistance de la formule et d'un terme d'arrondi :

- $e_{c1} = \dfrac{h}{2}\,|f"(\eta)| \simeq \dfrac{h}{4}$, pour la consistance et $e_{a1} = \dfrac{2\,u}{h}$ pour l'arrondi ; le minimum se situe aux alentours de $\sqrt{u} \simeq 10^{-8}$; la branche ascendante de la courbe en echelle logarithmique est approximativement un segment de pente 1.
- $e_{c2} = \dfrac{h^2}{6}\,|f'''(\eta)| \simeq \dfrac{h^2}{8}$, pour la consistance et $e_{a2} = \dfrac{u}{h}$ pour l'arrondi ; le minimum se situe aux alentours de $u^{\frac{1}{3}} \simeq 10^{-5}$; la branche ascendante de la courbe en echelle logarithmique est approximativement un segment de pente 2.
- $e_{c3} = \dfrac{h^4}{30}\,|f^{(5)}(\eta)| \simeq \dfrac{7\,h^4}{32}$, pour la consistance et $e_{a3} = \dfrac{3\,u}{2\,h}$ pour l'arrondi ; le minimum se situe aux alentours de $u^{\frac{1}{5}} \simeq 10^{-3}$; la branche ascendante de la courbe en echelle logarithmique est approximativement un segment de pente 4.

1.1.5 Erreurs sur les données

Pour la valeur $x = 1$, on obtient une précision de l'ordre de 10^{-2} dans le calcul de la dérivée d'une fonction régulière en utilisant la formule d'ordre 1 avec le pas $h = 10^{-2}$.

On se propose maintenant de calculer la dérivée de $f(x) = \sin(2\,x)$ de façon approchée sur l'intervalle $[1.5,\ 3]$ avec un pas $h = 0.01$. On définit les absisses $x_i = 1.5 + 0.01 \times (i-1)$ et on calcule les $y_i = \sin(2\,x_i)$ pour $1 \leq i \leq 152$. On obtient alors les dérivées approchées par

$$D_h(x_i) = 100 * (y_{i+1} - y_i),\ 1 \leq i \leq 151. \tag{1.9}$$

Les figures 1.5 comparent le résultat de ces calculs sur cet intervalle et sur l'intervalle $[2.3,\ 2.4]$ avec la fonction $f'(x) = 2\,\cos(2\,x)$: sur la première, elles sont en effet quasiment indiscernables, et la seconde montre que l'erreur est bien conforme aux prévision.

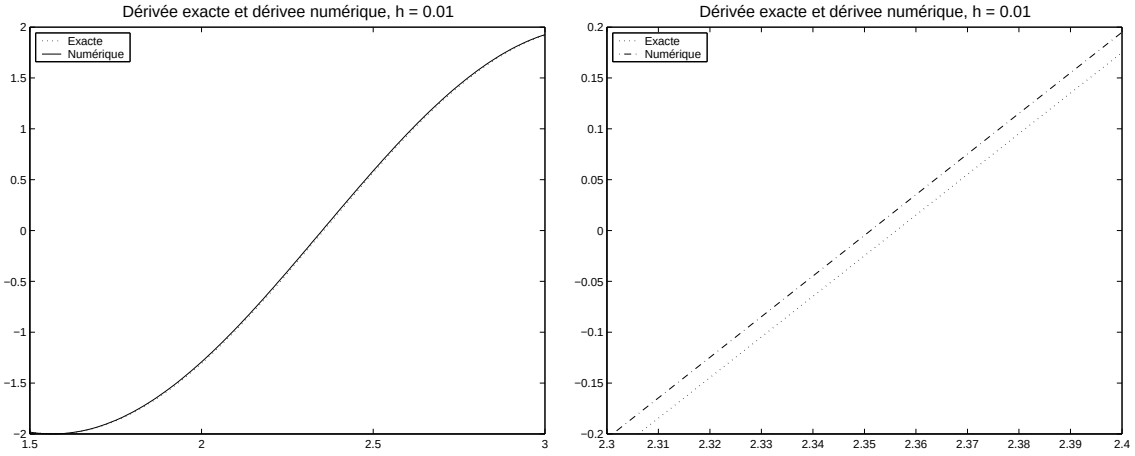


FIGURE 1.5 – La dérivée numérique approche la dérivée de façon consistante.

Supposons maintenant que nos valeurs de la fonction f soient les résultats de mesures d'un signal sinusoïdal de période π. Pour ces mesures, on dispose de 2 appareils, l'un fournissant une précison au centième, et l'autre au millième : par exemple, au lieu de $f(1.6) \simeq -0.0583741$, on obtient pour le premier appareil -0.06, et pour le second -0.058.

Ces mesures sont faites sur l'intervalle $[1.5,\ 3]$ pour chaque centième de l'unité de temps utilisée. On utilise les valeurs obtenues pour calculer la dérivée du signal sur cet intervalle, selon la stratégie utilisée précédement avec la formule d'ordre 1.

Nous pouvons alors constater sur les figures 1.6 que le calcul de la dérivée de ce signal n'est plus du tout correct.

Ces comportements s'expliquent encore à l'aide de la formule (1.6). Cette fois la précision de l'estimation de f n'est plus l'unité d'arrondi u de l'arithmétique

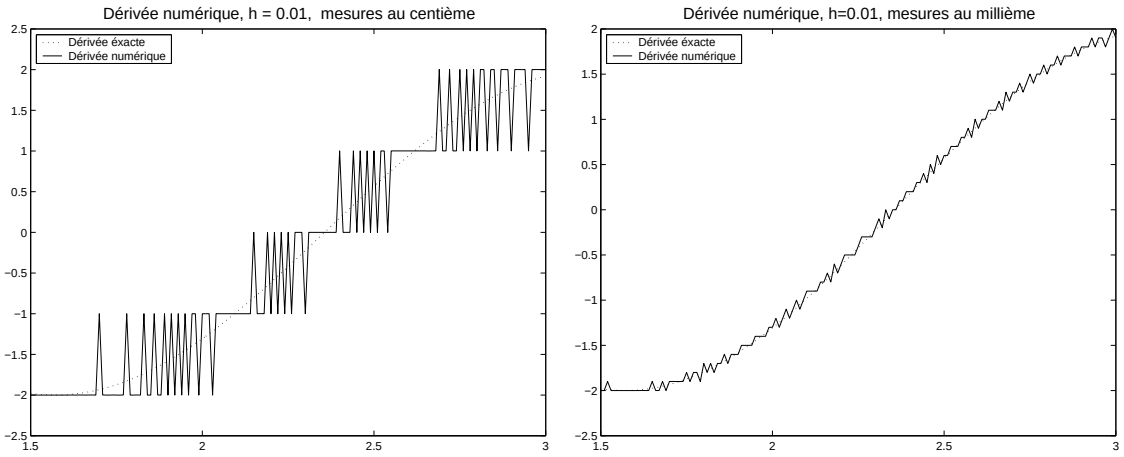


FIGURE 1.6 – Les mesures doivent être plus précises que la discrétisation.

utilisée, mais l'unité d'arrondi u_m de l'appareil de mesure, et pour l'appareil au centième $u_m = 0.005$; dans ce cas, les erreurs peuvent être de l'ordre de 1, car maintenant $\dfrac{2\,u_m}{h} = \dfrac{2 \times 0.005}{0.01} = 1\,!$

Et pour ce même appareil au centième, en choisissant un pas de $h = 0.1$ on obtient la courbe représentée à la figure 1.7 : elle est beaucoup plus correcte avec seulement 15 valeurs calculées, mais des erreurs sont de l'ordre de 0.1 : le terme de majoration empirique de l'erreur d'arrondi est $2 \times \dfrac{0.005}{0.1} = 0.1$. Il est du même ordre que le terme d'erreur de consistance puisque $\dfrac{h}{2}|f''(x)| \leq 2\,h$.

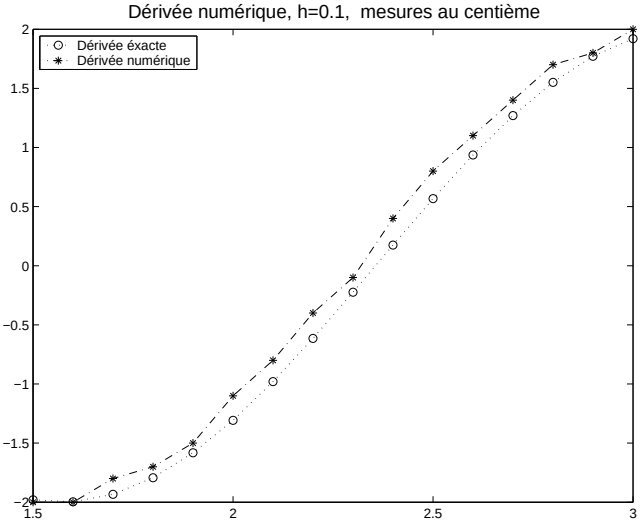


FIGURE 1.7 – Les mesures doivent être plus précises que la discrétisation (suite).

1.2 Stabilité et conditionnement

1.2.1 Une cascade d'approximations

Au départ, il y a un problème physique, ou plus généralement un système, pour lequel a été proposé un **modèle mathématique**. Ce modèle est déja le plus souvent une approximation plus ou moins grossière de la réalité. Ce modèle est écrit sous forme d'un système d'équations, par exemple un système d'équations différentielles. Il n'est pas garanti que l'on saura expliciter la solution de ce système d'équations.

Dans un second temps, on propose une **discrétisation du problème**, en vue de calculer une solution numérique. Cette discrétisation conduit à un système d'équations numériques, ou de relations caractérisant des nombres réels ou complexes qui représenteront correctement la solution.

Une troisième étape consistera à choisir ou à construire un **algorithme** permettant de calculer ces nombres. Cet algorithme sera parfois direct, permettant de calculer ces valeurs de façon exacte dans $\mathbb{R}^N$, mais parfois on ne pourra que construire une suite qui converge vers ces valeurs, qu'on ne pourra donc jamais calculer exactement. Ce sera en particulier le cas pour des équations non linéaires.

Enfin, cet algorithme sera **implanté en machine**, c'est-à-dire plongé dans l'ensemble $\mathcal{F}$ d'un ordinateur, et le programme sera appliqué en utilisant un jeu de données qui seront plus ou moins bruitées : les erreurs d'arrondi et de données viendront s'ajouter.

1.2.2 Notion de stabilité

Il y a donc toute une hiérarchie d'approximations avant d'arriver au résultat calculé par l'ordinateur. Pour espérer des "bons" résultats, il faut, mais il ne suffit pas nécessairement, que chacun de ces problèmes soit **stable**, c'est-à-dire peu sensible à d'éventuelles perturbations : chacune de ces approximations est en effet une perturbation du problème précédent. L'amplitude de cette perturbation est mesurée par une erreur de consistance. On est rassuré quand on peut établir une inégalité de la forme

$$\|e_{sol}\| \leq K \|e_{cons}\| \tag{1.10}$$

où e_{cons} et e_{sol} désignent les erreurs de consistance et sur la solution calculée.

Il faut donc, pour le moins, qu'une petite perturbation de chaque donnée du probléme initial n'engendre qu'une petite perturbation de la solution. Cette notion de stabilité est une sorte de généralisation de la notion de continuité.

1.2.3 Stabilité du problème

Retour à notre exemple d'élimination globale

Le calcul, pour x devenant de plus en plus petit, le calcul des valeurs de l'expression :

$$y(x) = \frac{(1+x)-1}{x}$$

peut s'interpréter comme celui de :

$$\lim_{h\to 0} \frac{(1+h)-1}{h} = i'(1),$$

c'est-à-dire de la valeur de la dérivée au point $x = 1$ de la fonction identité définie par $i(x) = x$. Cette dérivée est bien évidemment égale à 1.

L'opération qui associe à une fonction f sa dérivée f', n'est pas stable. En effet, si l'on considère, pour une fonction f de classe $\mathcal{C}^1$, une perturbation $\delta_\varepsilon(x) = \varepsilon \sin(\omega x)$, on vérifie que

$$(f + \delta_\varepsilon)'(x) = f'(x) + \varepsilon\,\omega\,\cos(\omega x)$$

et $\varepsilon\,\omega\,\cos(\omega x)$ peut prendre des valeurs arbitrairement grandes pour des ε arbitrairement petits : c'est le cas par exemple si $\omega = \dfrac{1}{\varepsilon^2}$.

Ainsi par exemple, le bruit que vous entendez dans les haut-parleurs de votre auto-radio lorsqu'une station n'est pas très bien captée, est un signal de très faible amplitude (ε est très petit) mais de très haute fréquence (ω est grand).

On peut en conclure qu'il est dangereux d'utiliser une formule de différences finies pour évaluer une dérivée. On l'a d'ailleurs constaté en calculant de cette façon la dérivée de la fonction $f(x) = \sin 2x$ avec des données perturbées !

Dans d'autres contextes cependant, de telles formules sont utilisées à bon escient : pour discrétiser certains problèmes différentiels, on peut écrire des schémas de discrétisation en exprimant la fonction inconnue à partir de ses dérivées approchées ; cette utilisation des formules de différences finies peut être stable puisqu'on part de la dérivée pour remonter à la fonction. Contrairement à la dérivation, l'intégration est une opération stable.

De façon générale, ce seront les derniers aspects de la hiérarchie d'approximations décrite plus haut qui retiendront surtout notre attention dans ce cours. On va cependant donner encore un exemple de problème instable.

Papillons et cyclônes

En 1961, le mathématicien Edward Lorenz qui se passionnait pour la météorologie a cherché à étudier plus précisément un des multiples phénomènes qui régissent le comportement de ce système naturel extrèmement complexe que constitue l'atmosphère ; il s'est intéressé à la convection. Ce phénomène repose sur une observation

simple : l'air chaud monte et l'air froid descend. Mais le soleil chauffe les océans, qui réchauffent l'air, qui se met à monter avant de refroidir en altitude et de redescendre. Si l'on veut décrire de façon exacte ce phénomène, en vue de prévisions par exemple, il faudra prendre en compte le paysage, les éléments qui le constituent (lac, forêt, plaine cultivée ou labourée, etc...) mais aussi la différence de température entre pôles et équateur, la rotation de la terre et les vents qu'elle engendre et une multitude d'autres facteurs.

Quand on veut passer aux calculs, il devient vraiment extrèmement compliqué de tenir compte de tous les facteurs. A titre d'exemple, les prévisions météorologiques se font à partir d'observations de stations réparties en un réseau si dense que les données à traiter se chiffrent par millions.

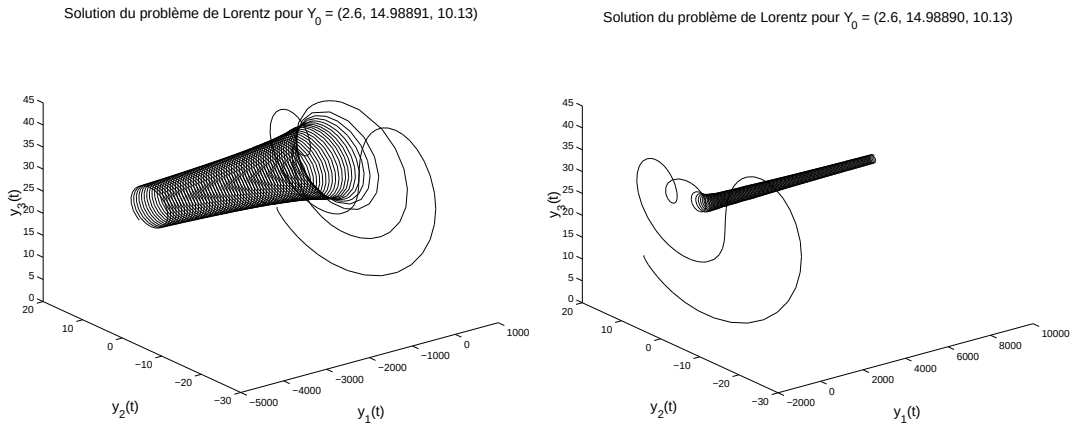


FIGURE 1.8 – Instabilité du modèle de Lorenz.

Lorenz a choisi de chercher à comprendre le phénomène, à partir d'un modèle extrèmement simplifié de trois équations différentielles reliant trois fonctions inconnues :

$$\begin{cases} \dfrac{dy_1}{dt} &= a\,(y_2 - y_1) \\ \dfrac{dy_2}{dt} &= y_1\,(b - y_3) - y_2 \\ \dfrac{dy_3}{dt} &= y_1\,y_2 - c\,y_3 \end{cases}$$

Les variables y_1 et y_2 représentent respectivement l'amplitude du champ de vitesse et du champ de température, et y_3 est reliée à l'évolution verticale de la température. a, b et c sont des paramètres physiques.

Ces équations semblent très simples, mais elles ne sont pas linéaires en raison de la présence des termes $y_1\,y_3$ et $y_1\,y_2$. On ne sait pas calculer leur solution générale.

En utilisant une méthode de discrétisation, Lorenz a décidé de faire des simultaions sur ordinateur, ce qui à l'époque n'était pas si fréquent. C'est par inadvertance

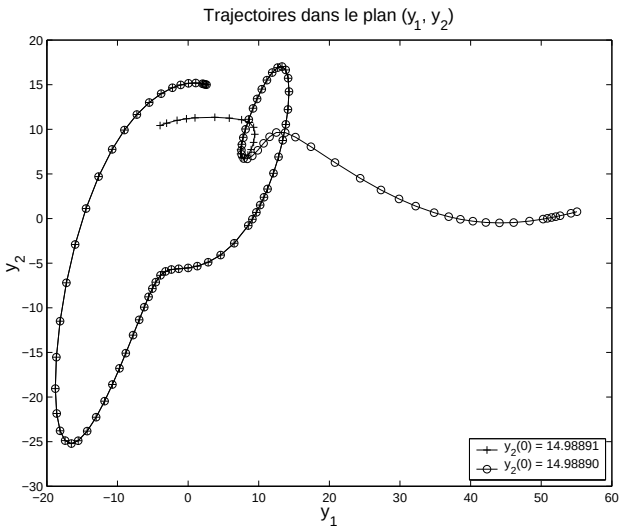
qu'il s'est rendu compte qu'en modifiant légèrement les données initiales, il obtenait des comportement très différents au bout d'un certain temps.

La figure 1.8 montre un exemple de courbes obtenues, en choisissant $a = 10, b = 28$ et $c = 3.3$.

Ces courbes représentent les trajectoires du point $M(t) = (y_1(t),\ y_2(t),\ y_3(t))$ sur un intervalle $[0,\ T]$. Pour calculer la solution approchée d'un tel système différentiel, il faut connaitre la valeur à l'instant 0 de chacune des fonctions inconnues ; on note $y_0 = (y_1(0),\ y_2(0),\ y_3(0))$ cette donnée initiale. Ces deux courbes montrent qu'une très petite perturbation de la donnée $y_2(0)$ suffit à modifier radicalement la solution ! On a en effet choisi,

- pour la courbe de gauche, (2.6, 1.498891, 10.13),
- pour la courbe de droite, (2.6, 1.498890, 10.13).

En fait, quelque soit la perturbation des données, les courbes finiront par adopter un comportement différent. Ici, elles restent presque confondues au début, avant de se séparer : la fonction $y_1(t)$ part dans deux directions opposées.



C'est ce comportement qui a amené Lorenz à proposer cette hypothèse, qu'une perturbation aussi infime pour l'atmosphère que le battement des ailes d'un papillon au Brésil pouvait engendrer bien plus tard une tornade au Texas. Un tel modèle ne permet pas de prévoir le temps à moyen terme, et d'autres canicules peuvent encore nous surprendre.

1.2.4 Stabilité d'un algorithme

Certaines méthodes peuvent se révèler très instables, pour calculer la solution de problèmes qui pourtant sont très stables. Il faut par exemple se méfier de certaines relations de récurrence.

L'exemple suivant a été proposé par Malcolm, Forsythe et Mohler en 1974.

Le problème

On souhaite calculer les valeurs des intégrales :

$$I_k = \int_o^1 x^k e^{-x} dx, \qquad \forall k,\ 1 \leq k \leq 50.$$

Une première analyse du problème permet d'établir :

$$\begin{aligned} I_0 &= 1 - \tfrac{1}{e}, \\ I_{k+1} &= (k+1) I_k - \frac{1}{e}, \quad k \geq 0 \end{aligned}$$

Il s'agit donc d'une formule de récurrence. On propose immédiatement l'algorithme suivant :

$I_0 \leftarrow 1 - \frac{1}{e}$;
Pour k **de** 0 **à** 49
$\quad I_{k+1} \leftarrow (k+1) I_k - \dfrac{1}{e}$;
Fin pour k

On obtient n'importe quoi : sur DEC-alpha, par exemple, $I_{18} = -0.1959\ 248$, et $I_{50} = -1.0275\ 336 \times 10^{48}$. C'est extravagant quand on sait que les valeurs doivent décroître en restant positives !

Analyse de sensibilté

On va perturber la donnée initiale et observer l'évolution de l'agorithme.

$I_0 = 1 - \dfrac{1}{e}$, mais la valeur calculée par la machine est légèrement différente ; supposons donc que l'on initialise notre récurrence par

$$\tilde{I}_0 = I_0 + \varepsilon.$$

Les premiers itérés que l'on va calculer seront :

$$\begin{aligned} \tilde{I}_1 &= \tilde{I}_0 - \frac{1}{e} &&= I_1 + \varepsilon, \\ \tilde{I}_2 &= 2\,\tilde{I}_1 - \frac{1}{e} &&= I_2 + 2\,\varepsilon, \\ \tilde{I}_3 &= 3\,\tilde{I}_2 - \frac{1}{e} &&= I_3 + 3!\,\varepsilon, \end{aligned}$$

et si l'on suppose que $\tilde{I}_k = I_k + k!\,\varepsilon$ on vérifie que $\tilde{I}_{k+1} = I_{k+1} + (k+1)!\,\varepsilon$.

On voit que la perturbation de départ est amplifiée par un facteur qui croît très vite.

Cette analyse de sensibilité permet de mettre en évidence l'instabilité de l'algorithme proposé. En double précision IEEE, on a $u \approx 1.11 \times 10^{-16}$ et comme

$50! \simeq 3.04 \times 10^{64}$, on comprend mieux les résultats précédents. La valeur calculée pour I_{50} vérifie en effet :

$$(I_{50})_c \simeq 0.3 \times u \times 50!$$

On a surtout calculé, et sans doute assez correctement, la propagation de l'erreur.

La bonne méthode

La récurrence $I_{k+1} = (k+1)\, I_k - \dfrac{1}{e}$ se retourne selon :

$$I_k = \frac{I_{k+1} + \frac{1}{e}}{k+1},$$

et cette idée sera exploitée pour implanter une récurrence descendante : lors du passage de I_{k+1} à I_k, les erreurs sur le calcul de I_{k+1} seront divisées par $k+1$.

Supposons que l'on parte, pour un réel a arbitraire, de

$$\tilde{I}_{60} = I_{60} + a.$$

La récurrence descendante va nous fournir

$$\begin{array}{lcll}
\tilde{I}_{59} & = & \dfrac{\tilde{I}_{60} + \frac{1}{e}}{60} & = I_{59} + \dfrac{a}{60}, \\
\tilde{I}_{58} & = & \dfrac{\tilde{I}_{59} + \frac{1}{e}}{59} & = I_{58} + \dfrac{a}{60 \times 59}, \\
\ldots & & \ldots & \ldots \\
\tilde{I}_{50} & = & \dfrac{\tilde{I}_{51} + \frac{1}{e}}{51} & = I_{50} + \dfrac{a}{60 \times 59 \ldots \times 51}.
\end{array}$$

En choisissant arbitrairement $a = 1$, on obtient une valeur de I_{50} à peu près exacte à la double précision IEEE puisque $\dfrac{50!}{60!} \simeq 3.6551\ 10^{-18}$.

On retrouve dans le tableau ci-dessous quelques valeurs calculées par les deux méthodes :

indice	indices croissants	indices décroissants
0	$6.321206e-01$	$6.321206e-01$
10	$3.646133e-02$	$3.646133e-02$
20	$-8.217689e+01$	$1.835047e-02$
30	$-8.961529e+15$	$1.224950e-02$
40	$-2.756558e+31$	$9.191388e-03$
50	$-1.027536e+48$	$7.504683e-03$

1.2.5 Notion de conditionnement

Erreur directe et erreur inverse

Supposons qu'on veuille calculer $y = f(x)$ en arithmétique finie, c'est-à-dire en utilisant un ordinateur. En raison du risque d'accumulations d'erreurs d'arrondi, on calculera en général une valeur $\tilde{y}$ différente de y. On sera satisfait si l'erreur relative $\dfrac{\|\tilde{y} - y\|}{\|y\|}$ est de l'ordre de la précision de l'arithmétique utilisée.

Cette erreur relative n'est en général pas accessible, faute de connaître la solution exacte. On peut alors s'intéresser à la question de savoir si la valeur effectivement calculée $\tilde{y}$ est solution exacte dans $\mathbb{R}$ (ou $\mathbb{R}^N$) d'un problème de même nature, c'est-à-dire s'il existe une perturbation Δx de la donnée x telle que :

$$\tilde{y} = f(x + \Delta x) \tag{1.11}$$

La figure 1.9 illustre ce raisonnement.

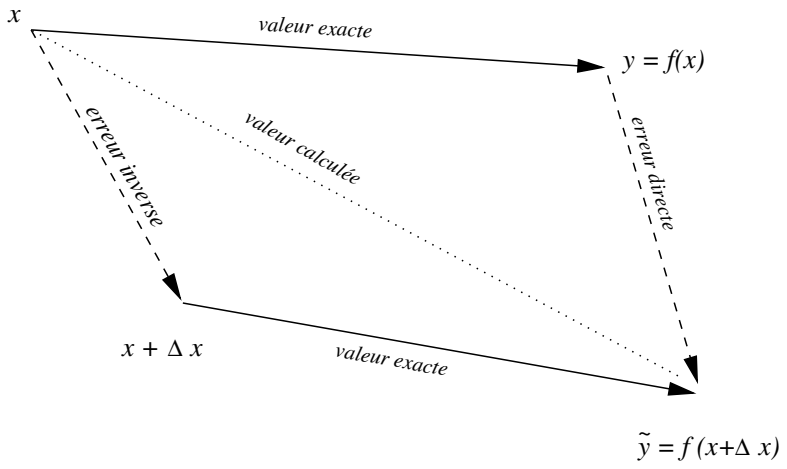


FIGURE 1.9 – Principe de l'analyse régressive.

On peut s'attendre à ce qu'il y ait un grand nombre de choix possibles pour Δx, et même qu'il y en ait une infinité. L'**erreur inverse** $e_{inv}(x)$ sera une mesure de la perturbation minimale qui permet de vérifier (1.11) :

$$e_{inv}(x) = \min \{\|\Delta x\|,\ \tilde{y} = f(x + \Delta x)\} \tag{1.12}$$

Cette expression est éventuellement divisée par $\|x\|$ pour avoir une erreur inverse relative. Dans le cadre de ce type d'analyse, l'erreur sur la solution est appelée **erreur directe**.

La relation entre erreur inverse et erreur directe se fait par le nombre de **conditionnement**. Elle est de la forme :

$$\|y - \tilde{y}\| \leq K_x \times e_{inv}(x) \tag{1.13}$$

En d'autres termes, le nombre de conditionnement donne une majoration de l'erreur sur la solution en fonction de perturbations des données ; il caractérise la sensibilité du problème à ces perturbations. Cette majoration est optimale lorsque l'on a accès à l'erreur inverse.

L'analyse d'erreur inverse, introduite par Wilkinson dans les années 1950, est la recherche de bornes pour l'erreur inverse. Cette analyse se fait en étudiant l'**algorithme de résolution** que l'on utilise. Ce type d'analyse est assez technique, et nous ne le développerons pas.

La recherche du nombre de conditionnement se fait en étudiant le **problème** à résoudre.

Conditionnement

L'étude du conditionnement permet d'avoir une idée du risque d'erreur sur la solution. Il convient tout d'abord de considérer les relations (1.12) et (1.13) en prenant en compte des erreurs relatives.

Une façon de définir le nombre de conditionnement consiste à poser :

$$K_x = \lim_{\Delta \to 0} \frac{\left| \frac{f(x+\Delta x)-f(x)}{f(x)} \right|}{\left| \frac{\Delta x}{x} \right|} \tag{1.14}$$

Le conditionnement global du problème sera la borne supérieure de ces quantités,

$$K = \sup_x \ K_x \tag{1.15}$$

Dans le cas de notre exemple, si la fonction f est suffisament régulière, on pourra écrire, pour un $\theta \in]0,\, 1[$:

$$\tilde{y} - y = f'(x)\,\Delta x + \frac{(\Delta x)^2}{2} f''(x + \theta \Delta x)$$

d'où

$$\frac{\tilde{y} - y}{y} = \frac{x f'(x)}{f(x)} \frac{\Delta x}{x} + \mathcal{O}(\Delta x^2)$$

qui permet de définir le conditionnement du problème :

$$K_x = \left| \frac{x f'(x)}{f(x)} \right|.$$

On obtient alors, pour quelques fonctions usuelles :

$f(x)$	$a\,x$	$a\,\sqrt{x}$	$a\,x^\alpha$	$x - a$	$\dfrac{1}{x-a}$
K_x	1	$\dfrac{1}{2}$	$\lvert\alpha\rvert$	$\dfrac{\lvert x\rvert}{\lvert x-a\rvert}$	$\dfrac{\lvert x\rvert}{\lvert x-a\rvert}$
K	1	$\dfrac{1}{2}$	$\lvert\alpha\rvert$	∞	∞

Dans le cas où le problème est de dimension $d > 1$, avec encore f de classe $\mathcal{C}^1$, on obtient, par un développement limité à l'ordre 1, et en notant $J_f(x)$ la matrice jacobienne de f en x,

$$f(x + \Delta x) - f(x) = J_f(x)\,\Delta x + o\left(\|x\|\right). \tag{1.16}$$

On obtient alors, en notant de façon analogue les normes vectorielle et matricielle induite,

$$\mathcal{K}_x = \frac{\|J_f(x)\|\,\|x\|}{\|f(x)\|}. \tag{1.17}$$

Exemple 1.1 *Le problème du calcul de $f(x_1, x_2) = x_1 - x_2$, avec donc f définie sur $\mathbb{R}^2$ à valeurs dans $\mathbb{R}$ a pour conditionnement relativement à la norme euclidienne*

$$\kappa = \frac{\sqrt{2\,(x_1^2 + x_2^2)}}{|x_1 - x_2|}.$$

On retrouve le fait que le calcul de la différence de 2 nombres est un problème mal conditionné, susceptible de conduire sur ordinateur à des erreurs d'absorption.

1.2.6 Conclusion

Qu'il s'agisse d'erreurs de discrétisation, d'erreurs sur les données ou d'erreurs d'arrondi, on cherche toujours une majoration de l'erreur sur la solution qui tienne compte d'une erreur de consistance, et d'une constante de stabilité qui mesure l'amplification de cette erreur.

Cette majoration respecte un principe fondamental de l'analyse numérique :

CONSISTANCE + STABILITE ⇒ CONVERGENCE

1.3 Annexe : Représentation des nombres en machine

1.3.1 Diverses représentations des nombres réels

Représentation traditonnelle

- Utilise la base 10,
- suite quelconque de chiffres après la virgule,

$$\forall x \in \mathbb{R}, \text{ il existe } N \in \mathbb{Z} \text{ tel que } 10^N \leq x \leq 10^{N+1}, \text{ et } \{a_i\}_{i\in\mathbb{N}},$$

$$\text{tels que } a_i \in \{0, \cdots, 9\}\ \forall i, \text{ et x}= \sum_{i=0}^{\infty} a_i\, 10^{N-i}$$

Représentation en base b

Elle repose sur le même principe que la représentation décimale, mais le nombre 10 est remplacé par un autre nombre.

– b un entier positif,

– $\forall x \in \mathbb{R}$, il existe $N \in \mathbb{Z}$ tel que $b^N \leq x \leq b^{N+1}$, et $\{\alpha_i\}_{i\in\mathbb{N}}$,tels que $\alpha_i \in \{0, \cdots, b-1\}$ $\forall i$, et x=$\sum_{i=0}^{\infty} \alpha_i\, b^{N-i}$.

La base 2

Cette base n'utilise que les deux chiffres 0 et 1. Elle est particulièrement intéressante pour l'informatique, car pour stocker ou transmettre des nombres, il suffit de disposer d'un mécanisme physique à deux états. Les tables d'opérations sont simples, et peuvent être simulées par des circuits électroniques simples.

Par exemple, $2^3 = 8 \leq 10 < 2^4 = 16$, et on vérifie que

$$10 = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0,$$

de sorte que le nombre 10 s'écrit 1010 en base 2.

La représentation du nombre $\dfrac{1}{10}$ est donnée par :

$$\frac{1}{10} \rightsquigarrow 0.00011001100110011\ldots. \tag{1.18}$$

La série de ses chiffres est infinie, mais périodique. On peut retrouver cette écriture de plusieurs façons :

- en effectuant la division dans la base 2, ce qui est assez déroutant :

1	0	0	0	0					**1**	**0**	**1**	**0**						
−	**1**	**0**	**1**	**0**					**0**.	**0**	**0**	**0**	**1**	**1**	**0**	**0**	**1**	…
	0	**1**	**1**	**0**	0													
	−	**1**	**0**	**1**	**0**													
			0	**1**	**0**	0	0	0										
				−	**1**	**0**	**1**	**0**										
					0	**1**	**1**	**0**										
					.	.	.	.										

- en remarquant que $\dfrac{1}{10} = \dfrac{1}{2^4} + \dfrac{1}{2^5} + \dfrac{1}{2^8} + \dfrac{1}{2^9} + \dfrac{1}{2^{12}} + \dfrac{1}{2^{13}} + \ldots + \dfrac{1}{2^{4k}} + \dfrac{1}{2^{4k+1}} + \ldots$

  En effet, comme $2^3 < 10 < 2^4$, il vient $2^{-4} < \dfrac{1}{10} < 2^{-3}$ et on vérifie, en partant de l'égalité $10 = 2^3 + 2^1$, que

$$\begin{aligned} 2^{-4}\,(2^3 + 2^1) &= \frac{1}{2} + \frac{1}{8} \\ (2^{-4} + 2^{-5})\,(2^3 + 2^1) &= \frac{1}{2} + \frac{1}{4} + \frac{1}{8} + \frac{1}{16} \;=\; 1 - \frac{1}{16} \end{aligned}$$

d'où

$$\left(\frac{1}{2^4}+\frac{1}{2^5}\right)\times\frac{1}{1-\frac{1}{16}}=\frac{1}{2^3+2^1}=\frac{1}{10}.$$

Mais $1-\dfrac{1}{16}$ est l'inverse de la somme de la série géométrique de raison $\dfrac{1}{16}=2^{-4}$; il vient donc :

$$\frac{1}{10}=\left(\frac{1}{2^4}+\frac{1}{2^5}\right)\sum_{k=0}^{\infty}\frac{1}{2^{4\,k}}=\sum_{k=1}^{\infty}\frac{1}{2^{4\,k}}+\sum_{k=1}^{\infty}\frac{1}{2^{4\,k+1}}. \tag{1.19}$$

Virgule flottante et notation scientifique

En anglais, on parlera de "floating point". Les représentations précédentes sont dites a virgule fixe, et la virgule sépare la partie entière de la partie "décimale", c'est-à-dire la partie composée des termes formés avec les puissances positives ou nulle de la base, et celle constituèe des termes formés avec les puissances négatives.

En base 10, les nombres 12.345, 0.12345 10^2 et 12345 10^{-3} sont égaux. Leur écriture est de la forme $x=\pm m\,b^e$, et dans cette écriture :

- m s'appelle la **mantisse**, ci-dessus respectivement 12.345, 0.12345 et 12345,
- b désigne la **base**, ci-dessus $b=10$,
- e s'appelle l'**exposant**, ci-dessus respectivement 0, +2 et −3.

De façon analogue, on peut écrire le nombre $x=\dfrac{1}{10}$ dans la base 2 sous la forme

$$x=0.0001100110011...\ 2^0=0.11001100...\ 2^{-3}=1.100110011\ldots\ 2^{-4}.$$

Le nombre de possibilités est infini. Pour mettre un peu d'ordre, on introduit la notion de virgule flottante normalisée : elle consiste à choisir l'exposant de façon que la mantisse soit de la forme

$$\alpha_0.\alpha_1\,\alpha_2\,\ldots\alpha_p\ldots,$$

avec $\alpha_0>0$. Cette convention entraine donc que $1\le m\le b$. On l'appelle notation en virgule flottante normalisée, ou notation scientifique.

Remarque 1.2 *Les nombres réels écrits ci-dessus le sont avec un point au lieu d'une virgule. C'est la notation que nous utiliserons dans ce cours.*

1.3.2 Représentation de nombres en machine

Il est impossible de représenter tous les nombres en machine, et pas seulement tous les nombres réels. On ne peut pas non plus espérer représenter l'ensemble de tous les nombres entiers, qui est infini.

Les nombres entiers

Dans la base 2, les nombres entiers seront stockés dans des éléments de mémoire formé de cellules qui sont elles mêmes constituées d'un nombre fixe d'unités élémentaires appélée bits (**bi**nary **it**ems) pouvant prendre la valeur 0 ou 1. Les cellules sont généralement constituées de 8 bits : on les appelle **octets** (ou bytes en anglais), et l'ordinateur utilise un certain nombre d'octets pour stocker les entiers :

- s'il utilise 2 octets, il représentera les nombres de compris entre $-32768 = -2^{15}$ et $32767 = 2^{15} - 1$,
- s'il utilise 3 octets, il représentera les entiers compris entre -2^{23} et $2^{23} - 1 = 8\ 388\ 607$ ...

Les réels flottants

- On ne peut représenter qu'un sous-ensemble fini de $\mathbb{R}$, l'ensemble $\mathcal{F}$ des flottants-machine. Un flottant est un mot-machine, c'est-à-dire une juxtaposition de n bits. Cet ensemble n'est pas universel ; il dépend de la machine.
- $\mathcal{F}(b, p, e_{\min}, e_{\max})$,
  - b = base,
  - p = nombre de chiffres de la mantisse,
  - $e_{\min}, e_{\max}$ = exposants minimum et maximum,
- représentation normalisée : $1 \leq m < b$, en fait plus précisément $m \leq b - \dfrac{1}{b^{p-1}}$
- $b = 2$ ou une puissance de 2, (8 ou 16), mais en principe, ne doit être que 2 (norme IEEE),
- double précision par adjonction de 2 mots-machine de n bits,
- les deux principaux formats de représentation des nombres réels préconisés par la norme IEEE-standard 754 sont les suivants :

précision	taille	mantisse	exposant	$e_{\min}$	$e_{\max}$
simple	32 bits	23+1 bits	8 bits	-126	127
double	64 bits	52+1 bits	11 bits	-1022	+1023

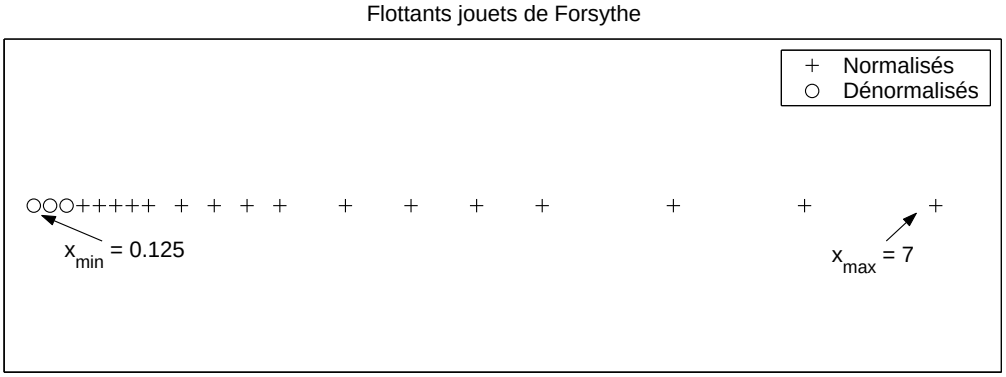
- la base étant 2, le premier bit de la mantisse est implicite (normalisation),
- dénormalisation autour de 0, etc...

Flottants jouets de Forsythe

C'est l'ensemble $\mathcal{F}(2, 3, -1, 2)$ où $p = 3 = 2+1$, avec le premier bit implicite ; les nombres positifs appartenant à cet ensemble sont représentés sur la figure ci-dessous :

On vérifie bien les propriétés annoncées ci-dessus :

- $1.00 \leq m \leq 1.11$, les bornes correspondant aux valeurs 1, $1+\dfrac{1}{2}$ et $1+\dfrac{1}{2}+\dfrac{1}{2^2} = 2 - \dfrac{1}{2^2} = 1.75$ écrites en base 2,
- les valeurs correspondant à la dénormalistion autour de 0 sont $\dfrac{3}{8}$, $\dfrac{1}{4}$ et $\dfrac{1}{8}$.



Représentation des réels en machine

- Il existe un plus grand nombre représentable, $x_{\max} = b^{e_{\max}}\left(b - \frac{1}{b^{p-1}}\right)$, appelé seuil d'**overflow** ; x est représentable en machine si $|x| \leq x_{\max}$; pour les flottants jouets, ce nombre est $7 = 2^2\,(2 - \frac{1}{2^2})$. En double précision IEEE, ce nombre est $x_{max} = (2 - \frac{1}{2^{52}}) \times 2^{1023} \simeq 1.8 \times 10^{308}$.
- il faut arrondir la plupart des nombres représentables ; on définit une application d'arrondi, notée fl :

$$x \in [-x_{\max}, x_{\max}] \longrightarrow \mathrm{fl}(x) \in \mathcal{F}\,(b, p, e_{\min}, e_{\max})$$

application d'arrondi, avec essentiellement deux stratégies :

- par **troncature**, $\mathrm{fl}(x) = \begin{cases} \max\{y \in \mathcal{F},\ y \leq x\} \text{ si } x > 0 \\ \min\{y \in \mathcal{F},\ y \geq x\} \text{ si } x < 0 \end{cases}$
  Dans le cas des flottants jouets, cette stratégie donne $\mathrm{fl}(3.4) = 3$. En effet, le nombre $x = 3.4$ s'écrit $x = 1.1011\ 0011\ 0011\ 0011\ldots\ 2^1$ en virgule flottante normalisée dans la base 2. En tronquant sa mantisse à 3 chiffres, il reste $\mathrm{fl}(x) = 1.10\ 2^1$.
- **au plus près**, $\forall y \in \mathcal{F}$, $|x - \mathrm{fl}(x)| \leq |x - y|$; c'est la stratégie préconnisée par IEEE, avec un dernier chiffre de mantisse pair en cas d'équidistance à deux flottants ("round to even"). Cette stratégie d'arrondi est mise en oeuvre en ajoutant 1 au $p+1$-ième chiffre de la mantisse avant de tronquer. Pour notre exemple, on tronquera le nombre $1.1101\,0011\,0011\,0011\ldots\ 2^1$, ce qui donne $\mathrm{fl}(x) = 1.11\ 2^1$, et donc $\mathrm{fl}(x) = 3.5$.
  Ajouter 1 au quatrième chiffre de la mantisse revient ici à ajouter le nombre d qui s'écrit $d = 0.001\ 2^1$ en base 2, c'est-à-dire $d = 2^{-2} = 0.25$, qui représente exactement la moitié de l'écart entre les 2 nombres de $\mathcal{F}$ qui encadrent 3.4 : 3 et 3.5.

- La **précision relative** est constante : l'intervalle entre deux nombres consécutifs de $\mathcal{F}\,(b, p, e_{\min}, e_{\max})$ n'est pas constant, mais pour tout nombre représentable x, l'erreur d'arrondi relative $\left|\frac{x - \mathrm{fl}(x)}{x}\right|$ sera du même ordre de grandeur. La plus grande valeur de cette erreur d'arrondi relative est l'unité d'arrondi,

$u = \frac{1}{2} b^{1-p}$.

– Pour les flottants jouets, on a $u = 0.125 = 2^{-3}$. On remarque que c'est la moitié de l'écart entre 1 et le plus petit nombre strictement supérieur à 1 représenté dans $\mathcal{F}$.

Chapitre 2

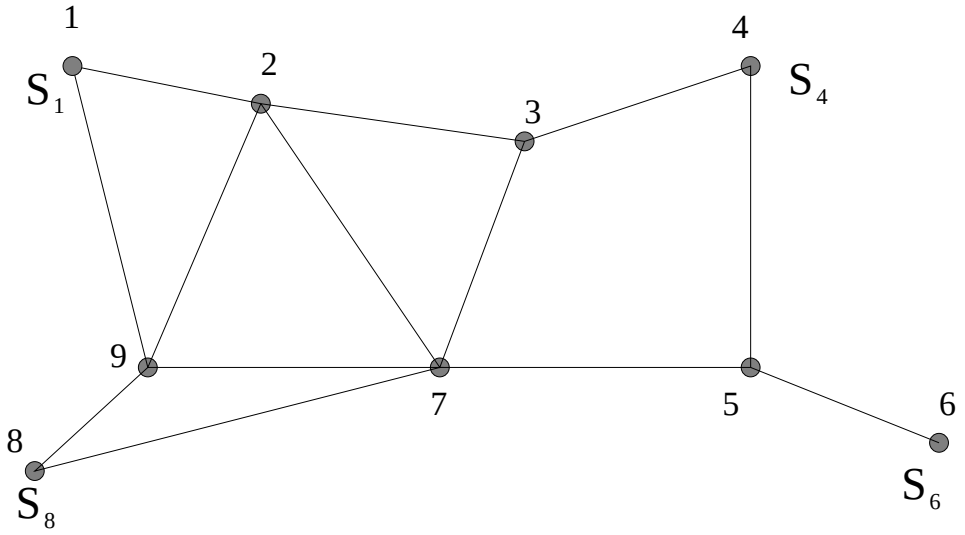
SYSTÈMES LINÉAIRES

L'étude des méthodes de résolution des systèmes linéaires est une étape obligatoire d'un cours de calcul scientifique ; presque tous les calculs passent par la résolution de tels systèmes : nous en avons rencontrés à l'occasion de problèmes d'interpolation et d'approximation, et l'ingénieur se trouve fréquemment confronté à la résolution de tels systèmes.

2.1 Problèmes de réseaux

Un réseau est un ensemble de noeuds P_i et d'arètes $E_{i,j}$ reliant certains de ces noeuds :

- lignes éléctriques,
- canalisations d'eau, égouts,...



Dans chaque arète circule un fluide ; à chaque noeud i est associé un potentiel u_i ; l'intensité (ou le débit) $q_{i,j}$ du fluide circulant entre les noeuds i et j est pro-

portionnelle à la différence de potentiel entre les extrémités i et j de l'arète où il circule ; c'est la loi d'Ohm pour les circuits électriques :

$$q_{i,j} = k_{i,j}(u_i - u_j)$$

Une loi physique de conservation (de Kirchoff dans le cas électrique) impose un équilibre : la somme algébrique des intensités en chaque noeud est égale à la valeur de la source (ou du puit) S_i qu'il figure (0 s'il est isolé).

Au noeud P_i, on a dans le cas du circuit électrique :

$$S_i = \sum_j q_{i,j} = \sum_j k_{i,j}(u_i - u_j)$$

Cette somme est étendue aux noeuds P_j adjacents de P_i ; considérons le réseau représenté par la figure ci-dessus. Les équations d'équilibre s'écrivent :

$$\left\{\begin{array}{lcl} S_1 & = & k_{1,2}(u_1 - u_2) + k_{1,9}(u_1 - u_9) \\ 0 & = & k_{2,1}(u_2 - u_1) + k_{2,9}(u_2 - u_9) + k_{2,7}(u_2 - u_7) + k_{2,3}(u_2 - u_3) \\ \cdots & = & \cdots \\ S_8 & = & k_{8,7}(u_8 - u_7) + k_{8,9}(u_8 - u_9) \\ 0 & = & k_{9,1}(u_9 - u_1) + k_{9,2}(u_9 - u_2) + k_{9,7}(u_9 - u_7) + k_{9,8}(u_9 - u_8) \end{array}\right.$$

de sorte que les potentiels à l'équilibre seront connus en résolvant le système linéaire

$$Au = S$$

avec une matrice A dont les coefficients non nuls sont représentés ci-dessous par une $*$:

$$A = \begin{bmatrix} * & * & & & & & & & * \\ * & * & * & & & * & & * & \\ & * & * & * & & * & & & \\ & & * & * & * & & & & \\ & & & * & * & * & * & & \\ & & & & * & * & & & \\ & * & * & & * & & * & * & * \\ & & & & & & * & * & * \\ * & * & & & & & * & * & * \end{bmatrix}$$

Le second membre est défini par $S^T = (S_1, 0, 0, S_4, 0, S_6, 0, S_8, 0)$.

On sait aujourd'hui résoudre assez correctement des systèmes à plusieurs millions d'inconnues (et d'équations), lorsqu'ils sont "assez creux", c'est-à-dire lorsque la matrice du système possède beaucoup de coefficients nuls ; pour les systèmes "pleins", on commence à avoir des problèmes au-delà de quelques centaines de milliers d'inconnues. Cela dépend bien sur des performances des ordinateurs que l'on utilise.

Pour les très grands systèmes, on utilise des méthodes itératives : on construit une suite de vecteurs qui converge vers la solution. Ces méthodes sont adaptées aux grands systèmes creux car elles ne manipulent pas la matrice, mais seulement une fonction qui réalise le produit de cette matrice avec un vecteur quelconque.

Pour les systèmes pleins, on utilise des méthodes directes, susceptibles de fournir la solution en arithmétique exacte après un nombre fini d'opérations élémentaires. Ce sont ces méthodes que nous étudierons dans ce cours.

On trouve des solveurs de systèmes linéaires qui mettent en oeuvre ce type de méthodes dans le domaine public et dans tous les logiciels de calcul scientifique. La disponibilité de ces solveurs ne dispense pas de la connaissance du principe des algorithmes qu'ils mettent en oeuvre, ne serait-ce que pour les utiliser à bon escient, et pour comprendre les indications qui sont fournies, ou les messages d'erreur qui sont renvoyés.

2.2 Sensibilité des systèmes linéaires

Un système comme celui que l'on vient de décrire sera construit à partir de valeurs mesurées. On peut se poser la question de la sensibilité de la solution à d'éventuelles erreurs de mesures.

Plus généralement, le système étant résolu par ordinateur en effectuant un certain nombre d'opérations élémentaires, la solution sera-t-elle très affectée par l'accumulation d'erreurs d'arrondis qu'implique cette succession d'opérations.

2.2.1 Exemple

On considère le système linéaire :

$$A\,x = \begin{bmatrix} 0.780 & 0.563 \\ 0.913 & 0.659 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0.217 \\ 0.254 \end{bmatrix} = b \tag{2.1}$$

dont la solution est $x = (1,\ -1)^T$.

La solution approchée $\widetilde{x^1} = (0.999,\ -1)^T = x + \Delta_1 x$ peut sembler satisfaisante au vu de sa faible différence avec la solution exacte ; elle fournit le résidu :

$$r^1 = A\,\widetilde{x^1} - b = \begin{bmatrix} -0.00078 \\ -0.00091 \end{bmatrix}$$

Le vecteur $\widetilde{x^2} = (0.341,\ -0.087)^T = x + \Delta_2 x$ ne semble pas être un candidat raisonnable pour la résolution de ce système ; et pourtant, cette fois, le résidu est bien plus satisfaisant :

$$r^2 = A\,\widetilde{x^2} - b = \begin{bmatrix} -0.000001 \\ 0 \end{bmatrix}$$

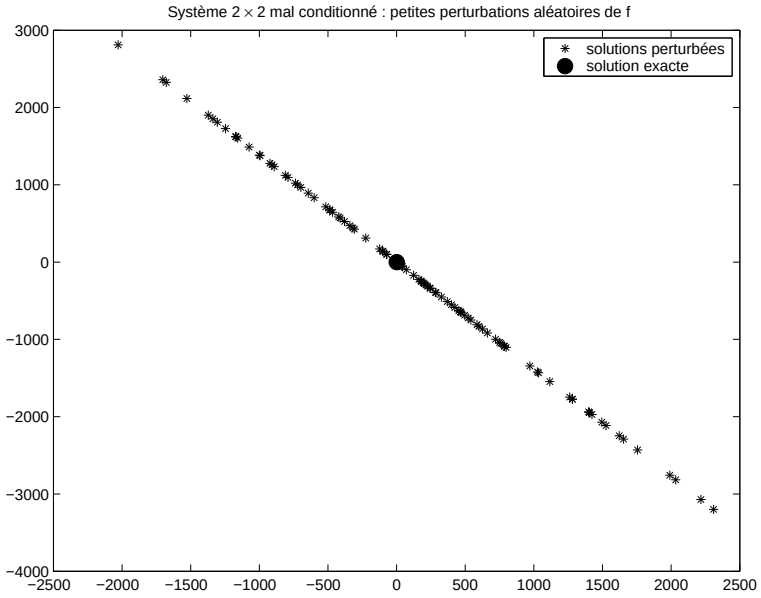
En terme de résidus et d'erreur, on a obtenu, respectivement pour x_1 et x_2 :

$\|\Delta x\|_2$	0.001	1.126
$\|r\|_2$	0.0012	0.000001

On constate ici que $\|\Delta x\|_2$ **et** $\|r\|_2$ **qui sont nuls simultanément ne sont pas petits simultanément !**

Une autre expérience consiste à modifier légèrement le second membre. Considérons le vecteur $\Delta b = \begin{bmatrix} -0.7603 \\ 0.6496 \end{bmatrix}$ et résolvons $A\,\widetilde{x^3} = b + 10^{-3}\,\Delta b$. Cette fois la solution est approximativement $\begin{bmatrix} -865.76 \\ 1199.84 \end{bmatrix}$ soit en notant $\Delta_3 x = x - \widetilde{x^3}$, $\|\Delta_3 x\|_2 \simeq 1481$.

La figure suivante représente la solution exacte du système $A\,x = b$ et les solutions obtenues pour 100 perturbations aléatoires du second membre, toutes inférieures en norme euclidienne à 0.0029.



On peut imaginer que le second membre est le résultat d'une mesure, ou de calculs précédents : accepteriez vous, par exemple, d'être le premier passager d'un avion en sachant que sa conception est passée par la résolution de ce système !

Le problème de la résolution d'un système linéaire n'est pas toujours un problème facile. On voit déja dans cet exemple très simple que l'on peut se poser la question de savoir si l'on cherche le vecteur solution du système, ou un vecteur qui vérifie le système. A précision donnée, les réponses peuvent être très différentes !

2.2.2 Distance à la singularité

On dit que la matrice A est **singulière** lorsque les systèmes linéaires de matrice A ne sont pas inversibles. Une matrice $A \in M_N(\mathbb{R})$ sera donc singulière si l'une ou l'autre des conditions équivalentes suivantes est vérifiée :

- le déterminant de A est nul,
- une des valeurs propres de A est nulle,
- le rang de A est strictement inférieur à N : c'est le cas si les lignes (ou les colonnes) de A ne sont pas linéairement indépendantes.

Une interprétation géométrique de l'exemple

On constate sur la figure précédente que toutes les solutions perturbées semblent se trouver sur une même droite. Un système linéaire 2×2 s'écrit sous la forme :

$$\left\{ \begin{array}{rcrcl} a_{1,1}\, x_1 & + & a_{1,2}\, x_2 & = & b_1 \\ a_{2,1}\, x_1 & + & a_{2,2}\, x_2 & = & b_2 \end{array} \right.$$

Chacune des équations est l'équation d'une droite du plan $(x_1,\, x_2)$. La solution est donnée par les coordonnées du point d'intersection de ces deux droites.

Pour notre exemple, les coefficients directeurs de ces deux droites sont respectivement -1.3854351 et -1.3854324 ! Elles sont donc presque parallèles. Il n'est pas étonnant que la localisation du point d'intersection soit délicate.

Si l'on modifie par exemple b_1, la première droite est remplacée par une droite qui lui est parallèle. Si les deux droites sont presque parallèles, le nouveau point d'intersection est très éloigné du précédent. La figure 2.2.2 illustre ce phénomène pour deux droites dont les coefficients directeurs sont -1 et $-\dfrac{10}{9}$, c'est-à-dire quand même assez différents, et qui se coupent en un point P. La droite en pointillé est parallèle à l'une des deux droites ; on a modifié la première composante du second membre pour l'obtenir. Son intersection avec la droite inchangée est le point Q.

Sensibilité et distance à la singularité

Le système pécédent est sensible aux perturbations. Comme on obtient un système équivalent en multipliant chaque ligne par un scalaire non nul, on peut le remplacer par le système obtenu en remplaçant chaque équation $(Eq : i)$ par $\dfrac{1}{a_{i,1}}\,(Eq : i)$

$$\left[\begin{array}{cc} 1 & 0.7217948 \\ 1 & 0.7217962 \end{array} \right] \left[\begin{array}{c} x_1 \\ x_2 \end{array} \right] = \left[\begin{array}{c} 0.2782051 \\ 0.2782037 \end{array} \right] \tag{2.2}$$

On voit ici que les lignes `A(1,:)` et `A(2,:)` sont presque dépendantes puisque

$$\texttt{A}(1,:) - \texttt{A}(2,:) = [\texttt{0},\ \texttt{1.4}\,\texttt{10}^{-\texttt{6}}].$$

A vrai dire, (2.2) est une autre façon d'écrire l'équation des deux droites mentionnées précédemment. On voit mieux que ces droites sont presque parallèles.

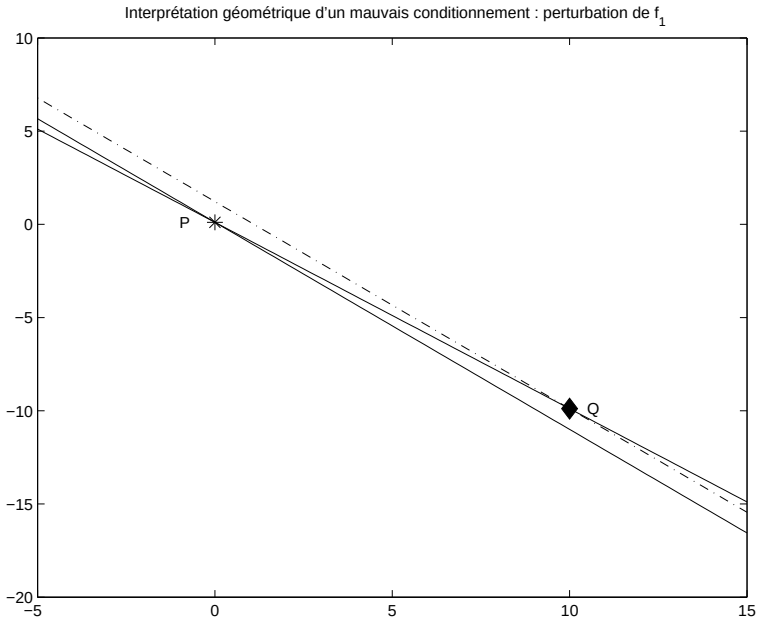


FIGURE 2.1 – Intersection de deux droites presque parallèles

Lorsque les lignes d'une matrice ne sont pas indépendantes, cette matrice est non inversible. On peut donc penser, que d'une certaine façon, c'est parce que la matrice A est proche d'une matrice singulière que le système est très sensible aux perturbations.

C'est effectivement une matrice proche de la singularité que l'on a utilisé. La plus petite valeur propre de A est à peu près 0.000000694. La matrice $A + \Delta A$ ci-dessous est singulière ; la perturbation ΔA qui la rend singulière est très petite.

$$A + \Delta A = \begin{bmatrix} 0.780 & 0.563 \\ 0.913 & 0.659 \end{bmatrix} + \begin{bmatrix} 0 & 0.000001095290252 \\ 0 & 0 \end{bmatrix}$$

Remarque 2.1 *Dans $\mathbb{R}$, le seul élément singulier est 0, de sorte que $|x|$ mesure la distance du réel x à la singularité. Dans $M_n(\mathbb{R})$, les matrices de la forme $\epsilon\, I$, où I désigne la matrice identité, sont d'inverse $\dfrac{1}{\epsilon} I$, et restent facilement inversibles tant que $\epsilon \neq 0$. Par contre des matrices, comme la matrice $A + \delta A$ ci-dessus, qui ne semblent pas particulièrement proches de 0 peuvent être non inversibles.*

2.2.3 Mesurer la distance à la singularité

On a constaté que la matrice de notre exemple était très proche d'une matrice singulière. On peut penser que l'on risque de rencontrer des problèmes en cherchant à résoudre des systèmes linéaires dont la matrice est proche d'une matrice singulière.

Une matrice est singulière lorsqu'elle n'est pas inversible. Il existe plusieurs façons de caractériser cela. $A \in M_n(\mathbb{R})$ est singulière si :

- son determinant est nul ; cela ne peut pas nous être d'une grande utilité, puisque par exemple, la matrice $0.5\,I_{100}$ qui est loin d'être singulière a un determinant à peu près égal à 7.8886×10^{-31},
- une de ses valeurs propres est nulle ; le calcul des valeurs propres n'est pas un problème très simple, et cette caractérisation n'est pas utilisable en pratique
- son rang est strictement inférieur à n ; dans ce cas, son noyau n'est pas réduit à 0, et il existe un vecteur x non nul tel que $A\,x = 0$.

Nous allons chercher à nous appuyer sur cette dernière caractérisation. Mais, il convient de réfléchir un peu si on veut le faire. Pour un vecteur x d'une grandeur donnée, mesurée par sa norme $\|x\|$, le vecteur $A\,x$ pourra être d'une grandeur très variable. La remarque 2.1 nous laisse penser que lorsque $\dfrac{\|A\,x\|}{\|x\|}$ reste constant, l'inversion de la matrice n'est pas toujours un problème numériquement difficile.

Nous pouvons alors associer à une matrice A donnée les deux nombres

$$\begin{array}{rcccl} \mathrm{a}(A) & = & \max\limits_{\|x\|=1} \|A\,x\| & = & \max\limits_{x\neq 0} \dfrac{\|A\,x\|}{\|x\|} \\ \mathrm{r}(A) & = & \min\limits_{\|x\|=1} \|A\,x\| & = & \min\limits_{x\neq 0} \dfrac{\|A\,x\|}{\|x\|} \end{array} \tag{2.3}$$

Ces deux nombres s'appelleront coefficients d'agrandissement et de réduction de la matrice A relativement à la norme considérée. Leurs deux expressions sont égales par définition des normes.

L'ensemble des $x \in \mathbb{R}^2$ tels que $\|x\|_2 = 1$ est le cercle unité. Si $A \in M_2(\mathbb{R})$, l'image du cercle unité par A est une ellipse. La figure 2.2 montre ces images dans le cas de la matrice $M = \begin{bmatrix} 0.5 & 1 \\ 0.75 & 0.5 \end{bmatrix}$ et dans le cas de la matrice A de (2.1).

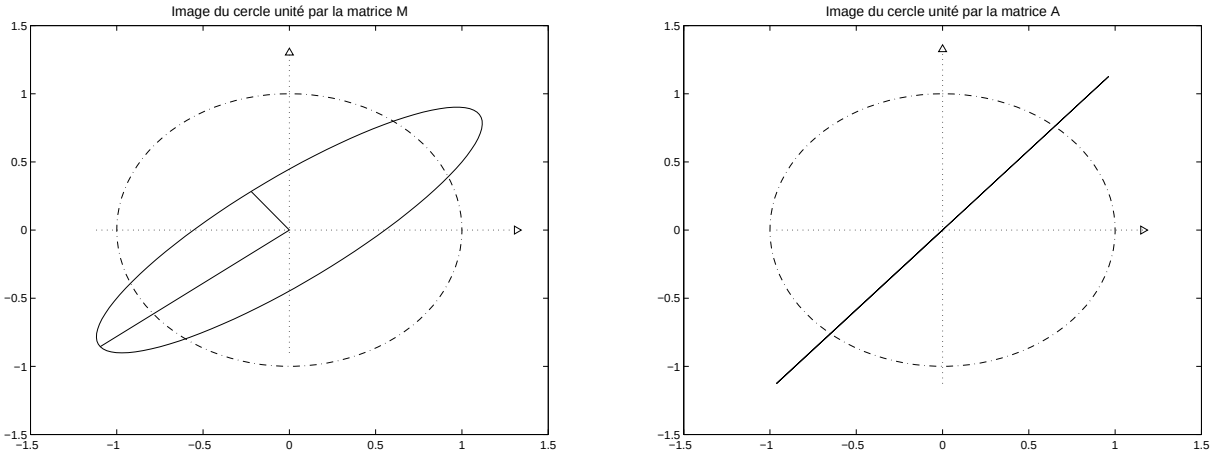


FIGURE 2.2 – Images du cercle unité par les matrices M et A

La partie gauche montre 2 vecteurs qui réalisent l'agrandissement et la réduction de la matrice M pour la norme euclidienne. Cette matrice n'est pas particulièrement proche d'une matrice singulière, et l'ellipse n'a rien de vraiment particulier.

Par contre, on constate sur la partie droite de cette figure que l'ellispe associée à la matrice A est très aplatie. On a vu que cette matrice est assez proche d'une matrice singulière.

Si une matrice n'est pas inversible, on aura $A\,u = 0$ pour au moins 2 vecteurs u et $-u$ du cercle et l'ellipse sera complètement aplatie. Mais, plus que l'agrandissement ou la réduction, c'est le rapport de ces deux nombres qui caractérise l'aplatissement de l'ellipse, et on peut penser à utiliser ce rapport $\dfrac{\mathrm{r}(M)}{\mathrm{a}(M)}$ pour évaluer la distance d'une matrice à la singularité. Ce rapport indique un changement de nature géométrique qui illustre la perte de rang caractéristique de la singularité ; on passe d'une surface à un segment. Lorsque ϵ devient très petit, l'image du cercle unité par la matrice $\epsilon\, I$ reste un cercle, et il suffit de changer d'échelle pour retrouver la surface initiale.

Remarque 2.2 *L'utilisation de la norme euclidienne n'est pas indispensable. On peut aussi bien utiliser une autre norme. La figure 2.3 montre les images de la boule unité pour la norme $\|.\|_\infty$, qui est un carré, par les matrices M et A.*

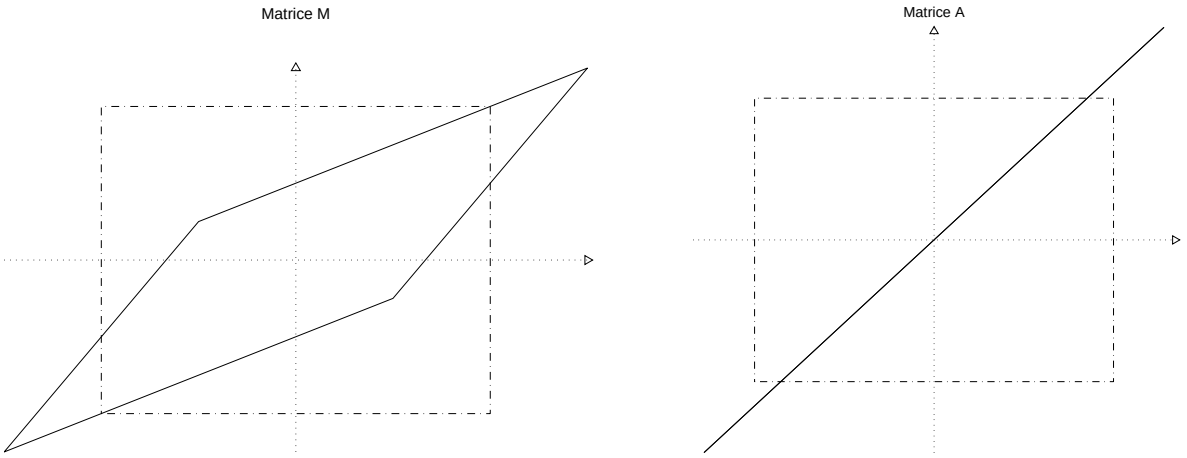


FIGURE 2.3 – Images du carré unité par les matrices M et A

2.2.4 Conditionnement

Conditionnement d'une matrice

C'est plus précisément le conditionnement d'une matrice pour la résolution d'un système linéaire que l'on veut étudier. Soit à résoudre le système linéaire $Ax = b$. On suppose que le calcul de la solution est fait par un algorithme quelconque implanté sur un ordinateur. Notons $\tilde{x} = x + \Delta x$ la solution obtenue.

On va supposer que $x+\Delta x$ est la solution exacte dans $\mathbb{R}^n$ d'un problème perturbé $A\,(x + \Delta x) = b + \Delta b$. On a ainsi choisi l'ensemble des perturbations admissibles, qui ne portent que sur le second membre ; d'autres choix sont possibles.

Comme $A\,x = b$ et $A\,\Delta x = \Delta b$, les définitions (2.3) nous fournissent :

$$\begin{array}{rcl} \|b\| & \leq & \mathrm{a}(A)\,\|x\| \\ \|\Delta b\| & \geq & \mathrm{r}(A)\,\|\Delta x\| \end{array}$$

d'où l'on tire, si A est inversible

$$\frac{\|\Delta x\|}{\|x\|} \leq \frac{\mathrm{a}(A)}{\mathrm{r}(A)}\,\frac{\|\Delta b\|}{\|b\|} \tag{2.4}$$

On peut donc définir $\mathcal{K}(A) = \dfrac{\mathrm{a}(A)}{\mathrm{r}(A)}$. C'est l'inverse de ce que l'on a appelé "la distance à la singularité de la matrice A". $\dfrac{\|\Delta b\|}{\|b\|}$ mesure ici l'erreur inverse relative, c'est-à-dire une perturbation sur la donnée b qui fournit l'erreur relative $\dfrac{\|\Delta x\|}{\|x\|}$ sur la solution .

Conditionnement et normes matricielles

La notion de norme définie pour les vecteurs de $\mathbb{R}^n$ s'étend aux matrices, en posant,

$$\|A\| = \max_{\|x\|=1} \{\|Ax\|\} \tag{2.5}$$

ce qui permet de vérifier, pour tout $x \in \mathbb{R}^n$:

$$\|Ax\| \leq \|A\| \times \|x\| \tag{2.6}$$

L'agrandissement $\mathrm{a}(A)$ n'est autre que $\|A\|$. Par ailleurs, si la matrice A est inversible, elle définit une bijection sur $\mathbb{R}^n$, de sorte que

$$\mathrm{r}(A) = \min_{x\neq 0} \frac{\|A\,x\|}{\|x\|} = \min_{y\neq 0} \frac{\|y\|}{\|A^{-1}\,y\|} = \frac{1}{\max\limits_{y\neq 0} \dfrac{\|A^{-1}\,y\|}{\|y\|}} = \frac{1}{\|A^{-1}\|}.$$

On obtient alors

Définition 2.1 *Soit $A \in M_n(\mathbb{R})$. On appelle conditionnement de A pour la résolution des systèmes linéaires,*

$$\mathcal{K}(A) = \begin{cases} \|A\|\ \|A^{-1}\| & \text{si } A \text{ est inversible,} \\ \infty & \text{sinon.} \end{cases} \tag{2.7}$$

Remarque 2.3 *Les matrices A et A^{-1} jouent un rôle analogue dans (2.7). Le nombre $\mathcal{K}\,(A)$ est en fait aussi le nombre de conditionnement de A pour les produits matrice-vecteur !*

Perturbation de la matrice et du second membre

On a considéré que la matrice A était fixée, et que seul le second membre b pouvait être perturbé. On va maintenant envisager une preturbation de la marice A et du second membre b.

Pour cela, on introduit l'application S définie sur $M_n(\mathbb{R}) \times \mathbb{R}^n$ par $S(A,\, b) = A^{-1}\, b$. Nous allons essayer d'expliciter les valeurs de la dérivée $D_{S(A,\, b)}(H,\, h)$ pour $H \in M_n(\mathbb{R})$ et $h \in \mathbb{R}^n$ suffisamment petits.

$$\begin{aligned} S(A+H,\, b+h) &= (A+H)^{-1}\,(b+h), \\ &= \left(A\,(I + A^{-1}\,H)\right)^{-1}\,(b+h), \\ &= (I + A^{-1}\,H)^{-1}\;A^{-1}\,(b+h), \end{aligned}$$

en fait $\|A^{-1}\,H\| < 1$, Pour H suffisamment petite, $\left(I + A^{-1}\,H\right)^{-1} = \sum_{k\geq 0}(-A^{-1}\,H)^k$. C'est vrai dès que $\|A^{-1}\,H\| < 1$, et on retrouve la somme d'une série géométrique dans $M_n(\mathbb{R})$. On obtient finalement

$$S(A+H,\, b+h) = A^{-1}\,b + A^{-1}\,h - A^{-1}\,H\,A^{-1}\,b - A^{-1}\,H\,A^{-1}\,h + \sum_{k\geq 2}(-A^{-1}\,H)^k\,A^{-1}\,(b+h).$$

On reconnait dans cette expression,
- la partie constante en $(H,\, h)$, $S(A,\, b) = A^{-1}\,b$,
- la partie linéaire $(H,\, h)$, $D_{S(A,\, b)}(H,\, h) = A^{-1}\,h - A^{-1}\,H\,A^{-1}\,b$
- et tous les autres termes, non linéaires en $(H,\, h)$.

On note $x + \Delta x$ la solution de $(A+\Delta A)\,(x+\Delta x) = b + \Delta b)$; si on suppose $\|\Delta A\| \leq \varepsilon$ et $\|\Delta b\| \leq \varepsilon$, il vient suivant le calcul précédent

$$\Delta x = A^{-1}\,\Delta b - A^{-1}\,\Delta A\,x + \mathcal{O}(\varepsilon^2).$$

En utilisant $A\,x = b \;\Rightarrow\; \|b\| \leq \|A\|\,\|x\| \;\Rightarrow\; \dfrac{1}{\|x\|} \leq \dfrac{\|b\|}{\|A\|}$,

$$\begin{aligned} \frac{\|\Delta x\|}{\|x\|} &\leq \frac{\|A^{-1}\|\,\|\Delta b\|}{\|x\|} + \frac{\|A^{-1}\|\,\|\Delta A\|\,\|x\|}{\|x\|} + \mathcal{O}(\varepsilon^2), \\ &\leq \|A^{-1}\|\,\|A\|\left(\frac{\|\Delta b\|}{\|b\|} + \frac{\|\Delta A\|}{\|A\|}\right) + \mathcal{O}(\varepsilon^2). \end{aligned}$$

Finalement, on obtient bien, à l'ordre 1,

$$\frac{\|\Delta x\|}{\|x\|} \leq \mathcal{K}(A)\,\left(\frac{\|\Delta A\|}{\|A\|} + \frac{\|\Delta b\|}{\|b\|}\right) \tag{2.8}$$

Exemple 2.1 *Essayons de vérifier la pertinence de ce nombre de conditionnement sur notre exemple. Nous pouvons utiliser la norme du maximum.*

Pour $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$, *on aura* $A\,x = \begin{bmatrix} 0.780\,x_1 + 0.563\,x_2 \\ 0.913\,x_2 + 0.659\,x_2 \end{bmatrix}$, *de sorte que*

$$\|A\,x\|_\infty = \max\left\{|0.780\,x_1 + 0.563\,x_2|,\; |0.913\,x_2 + 0.659\,x_2|\right\}. \tag{2.9}$$

Si on impose $\max\{|x_1|,\ |x_2|\} = 1$, *le maximum de* $\|A\,x\|_\infty$ *sera atteint pour le vecteur* x *tel que* $x_1 = x_2 = 1$, *et il vaut* $\|A\|_\infty = 1.572$.

Par contre la matrice A^{-1} *est donnée par* $A^{-1} = 10^5 \times \begin{bmatrix} 6.59 & -5.63 \\ -9.13 & 7.80 \end{bmatrix}$, *et par un raisonnement analogue,* $\|A^{-1}\|_\infty = 16.93 \times 10^5$. *On a donc un conditionnement de la matrice donné par* $K_\infty = 2.661396 \times 10^6$.

Appelons u *la solution du système* $A\,u = \begin{bmatrix} 1.343 \\ 1.572 \end{bmatrix}$, *et observons*

$$A\,(u + \Delta u) = \begin{bmatrix} 2.565 \\ -0.121 \end{bmatrix}.$$

Ce second membre est tel que $\dfrac{\|\Delta b\|}{\|b\|} = \dfrac{1.6930}{1.572} \simeq 1.077$.

La solution du premier système est $u \simeq \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, *mais* $u + \Delta u \simeq 10^6 \times \begin{bmatrix} 1.7585 \\ -2.4362 \end{bmatrix}$.

L'inégalité (2.8) est bien vérifiée :

$$10^6 \times 2.4362 \leq 2.661396 \times 10^6 \times 1.077.$$

Remarque 2.4 *La relation (2.9) traduit la définition de la notion de norme matricielle dans un cas particulier. Pour la matrice* $A = (a_{i,j})_{1 \leq j,i \leq n} \in M_n(\mathbb{R})$, *on peut montrer que*

$$\|A\|_\infty = \max_{1 \leq i \leq n} \left\{ \sum_{j=1}^{n} |a_{i,j}| \right\} \tag{2.10}$$

D'autre part, on préfère souvent travailler avec la norme euclidienne. On montrera en TD que la norme matricielle associée est définie par

$$\|A\|_2 = \sqrt{\lambda_{\max}(A^T\,A)} \tag{2.11}$$

où $\lambda_{\max}(A^T\,A)$ *désigne la plus grande valeur propre de la matrice* $A^T\,A$.

Lorsque la matrice A *est symétrique, on aura donc*

$$\|A\|_2 = \lambda_{\max}(A). \tag{2.12}$$

2.3 Factorisation LU

2.3.1 Comment résoudre les systèmes linéaires ?

Les systèmes faciles à résoudre

Il existe des systèmes linéaires qui sont de résolution facile. Commençons par en examiner la résolution.

Systèmes diagonaux Si A est une matrice diagonale, $A = (a_{i,j})_{1 \leq i,j \leq n}$ avec $a_{i,j} = 0$ si $i \neq j$, le système $Ax = b$ est immédiatement résolu en posant pour tout i, $1 \leq i \leq n$, $x_i = \dfrac{b_i}{a_{i,i}}$.

Le coût de cette résolution est $c(n) = n$ opérations élémentaires.

Systèmes triangulaires Les systèmes de matrice triangulaire sont également de résolution facile. Examinons le cas de systèmes triangulaires supérieurs $T\,x = b$. La matrice T est telle que $t_{i,j} = 0$ si $i > j$.

Prenons l'exemple du système

$$\begin{bmatrix} 4 & 5 & 1 \\ 0 & 3 & 4 \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 4 \\ 1 \\ 2 \end{bmatrix}. \tag{2.13}$$

Ce genre de système se résoud par une méthode dite de **substitution rétrograde** ou de **remontée**. On commence par calculer la dernière composante, on utlise sa valeur pour calculer l'avant dernière, et on recommence selon le même principe jusqu'à la première.

On calcule successivement

$$\begin{aligned} x_3 &= \frac{2}{2} &&= 1 \\ x_2 &= \frac{1 - 4 \times 1}{3} &&= -1 \\ x_1 &= \frac{4 - 5 \times (-1) - 1 \times 1}{4} &&= 2 \end{aligned}$$

Ceci est récapitulé par l'algorithme suivant :

$x_n \longleftarrow b_n \,/\, T_{n,n}$
Pour k de $n-1$ à 1 par pas de -1,
$\quad x_k \longleftarrow \left(b_k - \sum_{j=k+1}^{n} T_{k,j} * x_j \right) \Big/ T_{k,k}$

Et qui donne, par exemple en MATLAB :

```
x(n) = b(n)/T(n,n);
for k = n-1:-1:1
    x(k) = (b(k) - T(k,k+1:n)*x(k+1:n))/T(k,k);
end
```

On peut évaluer le nombre d'opérations nécessaires à la résolution d'un système triangulaire en fonction de la taille n du système :

- on effectue 1 division pour $x_n \longleftarrow b_n \,/\, T_{n,n}$

- pour chaque valeur de k comprise entre 1 et N-1, on effectue
  - $n-k$ multiplications et $n-k-1$ additions pour $\alpha = \sum_{j=k+1}^{n} T_{k,j} * x_j$, produit scalaire de deux vecteurs de $n-k$ composantes,
  - 1 soustraction et 1 division pour $(b_k - \alpha)/T_{k,k}$.

Cela fait donc $2\,(n-k)+1$ opérations, et comme k varie de 1 à $n-1$, $n-k$ varie également de 1 à $n-1$: on obtient $2\,\dfrac{(n-1)\,n}{2} + n - 1 = n^2 - 1$ opérations auxquelles on ajoute la première division pour obtenir le coût $c(n) = n^2$ opérations élémentaires.

2.3.2 L'idée des méthodes directes

Transformer le système en un système diagonal, c'est diagonaliser A. Quand c'est possible, c'est le problème de recherche des éléments propres de la matrice A, et c'est beaucoup plus difficile que la résolution du système linéaire ! On ne peut pas espérer s'en sortir de cette façon.

Par contre on peut espèrer transformer notre système en un système triangulaire. En fait, on va remplacer la résolution de notre système par la résolution de deux systèmes triangulaires, à l'aide d'une factorisation de la matrice A de la forme :

$$A = L\,U$$

où L est une matrice triangulaire inférieure (**L**ower) et U une matrice triangulaire supérieure (**U**pper).

2.3.3 De l'élimination de Gauss à la factorisation LU

Elimination de Gauss

Soit à résoudre le système $A\,x = b$, où la matrice A appartient à $M_n(\mathbb{R})$. C'est un système de n équations, notées $(Eq.1), \ldots, (Eq.n)$, à n inconnues notées $x_1, \ldots, x_n$, qui sont les composantes du vecteur x.

Le principe de la méthode est d'éliminer les inconnues de proche en proche, en faisant disparaitre :

- l'inconnue x_1 des équations $(Eq.2)$ à $(Eq.n)$,
- l'inconnue x_2 des équations $(Eq.3)$ à $(Eq.n)$,
- ... et pour k variant de 1 à n, l'inconnue x_k des équations $(Eq.k+1)$ à $(Eq.n)$.

Pour ce faire, on transforme à chaque étape le système en un système équivalent, c'est-à-dire admettant exactement le même ensemble de solutions, au moyen d'une **transformation élémentaire** :

- ajout d'un multiple d'une équation à une équation donnée : cela consiste par exemple à remplacer l'équation $(Eq.i)$ par l'équation $(Eq.i) + \alpha\,(Eq.k)$
- permutation de deux équations,

A l'étape k, le coefficient $a_{k,k}^{(k)}$ de l'inconnue x_k s'appelle le **pivot**. On le note p_k. Il permet l'élimination de l'inconnue x_k des équations $(Eq.k+1)$ à $(Eq.n)$, en remplaçant l'équation $(Eq.i)$ par $(Eq.i) - \dfrac{a_{i,k}^{(k)}}{p_k}(Eq.k)$ pour $k+1 \leq i \leq n$.

On va d'abord s'intéresser à la méthode de Gauss sans permutation des équations. Le pivot intervient en dénominateur d'une fraction : il doit être non nul.

Exemple

On considère le système

$$\left\{\begin{array}{rcrcrcl} 2x_1 & & +x_2 & & +x_3 & = & 1 \\ 6x_1 & & +2x_2 & & +x_3 & = & -1 \\ -2x_1 & & +2x_2 & & +x_3 & = & 7 \end{array}\right. \iff \begin{bmatrix} 2 & 1 & 1 \\ 6 & 2 & 1 \\ -2 & 2 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ 7 \end{bmatrix} \tag{2.14}$$

On note $A^{(1)} = A$ et $b^{(1)} = b$ ces données initiales : ce système s'écrit donc $A^{(1)}\,x = b^{(1)}$. On peut choisir comme pivot le coefficient $p_1 = 2$ de x_1 dans la première équation. On élimine x_1 des équations $(Eq.2)$ et $(Eq.3)$ par la transformation élémentaire :

- $(Eq.2)$ est remplacée par $(Eq.2) - \dfrac{6}{p_1}(Eq.1) = (Eq.2) - 3\,(Eq.1)$,
- $(Eq.3)$ est remplacée par $(Eq.3) - \dfrac{-2}{p_1}(Eq.1) = (Eq.3) + (Eq.1)$.

On obtient le système équivalent :

$$\left\{\begin{array}{rcrcrcl} 2x_1 & + & x_2 & + & x_3 & = & 1 \\ & - & x_2 & - & 2x_3 & = & -4 \\ & + & 3x_2 & + & 2x_3 & = & 8 \end{array}\right. \iff \begin{bmatrix} 2 & 1 & 1 \\ 0 & -1 & -2 \\ 0 & 3 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ -4 \\ 8 \end{bmatrix}$$

On note $A^{(2)}\,x = b^{(2)}$ ce deuxième système, équivalent au premier. On peut choisir comme pivot le coefficient $p_2 = -1$ de x_2 dans la deuxième équation. On élimine x_2 de l'équation $(Eq.3)$:

- $(Eq.3)$ est remplacée par $(Eq.3) - \dfrac{3}{p_2}(Eq.2) = (Eq.2) + 3\,(Eq.2)$.

On obtient le système équivalent :

$$\left\{\begin{array}{rcrcrcl} 2x_1 & + & x_2 & + & x_3 & = & 1 \\ & - & x_2 & - & 2x_3 & = & -4 \\ & & & - & 4x_3 & = & -4 \end{array}\right. \iff \begin{bmatrix} 2 & 1 & 1 \\ 0 & -1 & -2 \\ 0 & 0 & -4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ -4 \\ -4 \end{bmatrix}$$

Ce dernier système, $A^{(3)}\,x = b^{(3)}$ est triangulaire supérieur. Il est inversible puisque les éléments diagonaux de $A^{(3)}$, p_1, p_2 et $p_3 = -4$ sont tous non nuls. On le résoud par substitution rétrograde pour obtenir $x = \begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix}$.

Ecriture matricielle des transformations élémentaires

A la première étape de l'exemple (2.14), la matrice $A^{(1)} = \begin{bmatrix} 2 & 1 & 1 \\ 6 & 2 & 1 \\ -2 & 2 & 1 \end{bmatrix}$ est remplacée par la matrice $A^{(2)} = \begin{bmatrix} 2 & 1 & 1 \\ 0 & -1 & -2 \\ 0 & 3 & 2 \end{bmatrix}$. En termes de matrice, le fait de modifier une équation correspond à la modification d'une ligne. On constate en effet, en utilisant la notation MATLAB, que :

$$\begin{array}{rcl} \mathtt{A}^{(2)}(1,:) & = & \mathtt{A}^{(1)}(1,:), \\ \mathtt{A}^{(2)}(2,:) & = & \mathtt{A}^{(1)}(2,:) - 3\,\mathtt{A}^{(1)}(1,:), \\ \mathtt{A}^{(2)}(3,:) & = & \mathtt{A}^{(1)}(3,:) - (-1)\,\mathtt{A}^{(1)}(1,:). \end{array}$$

Regroupons ces lignes pour écrire les matrices correspondantes :

$$\begin{bmatrix} \mathtt{A}^{(2)}(1,:) \\ \mathtt{A}^{(2)}(2,:) \\ \mathtt{A}^{(2)}(3,:) \end{bmatrix} = \begin{bmatrix} \mathtt{A}^{(1)}(1,:) \\ \mathtt{A}^{(1)}(2,:) \\ \mathtt{A}^{(1)}(3,:) \end{bmatrix} - \begin{bmatrix} 0 \\ 3\,\mathtt{A}^{(1)}(1,:) \\ -\,\mathtt{A}^{(1)}(1,:) \end{bmatrix} \tag{2.15}$$

Le second terme du membre de droite de (2.15) s'écrit :

$$\begin{bmatrix} 0 \\ 3\,\mathtt{A}^{(1)}(1,:) \\ -\,\mathtt{A}^{(1)}(1,:) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 3a_{1,1}^{(1)} & 3a_{1,2}^{(1)} & 3a_{1,3}^{(1)} \\ -a_{1,1}^{(1)} & -a_{1,2}^{(1)} & -a_{1,3}^{(1)} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 3 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} A^{(1)}. \tag{2.16}$$

de sorte que finalement, en notant I la matrice identité :

$$A^{(2)} = \left(I - \begin{bmatrix} 0 & 0 & 0 \\ 3 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix} \right) A^{(1)} \tag{2.17}$$

Cette écriture se simplifie en introduisant les vecteurs $l^1 = [0,\ 3,\ -1]^T$ et $e_1 = [1,\ 0,\ 0]^T$, premier vecteur de la base canonique :

$$A^{(2)} = (I - l^1\, e_1^T)\, A^{(1)} = E^{(1)}\, A^{(1)}. \tag{2.18}$$

De façon analogue, le passage de $A^{(2)}$ à $A^{(3)} = \begin{bmatrix} 2 & 1 & 1 \\ 0 & -1 & -2 \\ 0 & 0 & -4 \end{bmatrix}$ se fait selon

$$A^{(3)} = \left(I - \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & -3 & 0 \end{bmatrix} \right) A^{(2)} = (I - l^2 e_2^T)\, A^{(2)} = E^{(2)}\, A^{(2)},$$

où $l^2 = [0,\ 0,\ -3]^T$ et e_2 désigne le second vecteur de la base canonique.

Notons $U = A^{(3)}$ la matrice triangulaire supérieure obtenue à l'issue de cette étape d'élimination. Elle vérifie :

$$U = E^{(2)}\, E^{(1)}\, A. \tag{2.19}$$

Les matrices $E^{(1)}$ et $E^{(2)}$ sont inversibles, selon

- $\left(E^{(1)}\right)^{-1} = L^{(1)} = (I + l^1\, e_1^T) = \begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix}$; on le vérifie en écrivant $(I - l^1\, e_1^T)\,(I + l^1\, e_1^T) = I - l^1\, \left(e_1^T\, l^1\right)\, e_1^T = I$, puisque

$$e_1^T\, l^1 = 1 \times 0 + 0 \times 3 + 0 \times (-1) = 0\ ;$$

- $\left(E^{(2)}\right)^{-1} = L^{(2)} = (I + l^2\, e_2^T)$; on le vérifie de la même façon en écrivant $(I - l^2\, e_2^T)\,(I + l^2\, e_2^T) = I - l^2\,(e_2^T\, l^2)\, e_2^T = I$, puisque

$$e_2^T\, l^2 = 0 \times 0 + 1 \times 0 + 0 \times (-3) = 0.$$

En multipliant à gauche (2.19) par $L^{(1)}\, L^{(2)}$, on obtient :

$$L^{(1)}\, L^{(2)}\, U = L\, U = A. \tag{2.20}$$

La matrice L "s'assemble" sans calcul, selon

$$L = \begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ -1 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -3 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ -1 & -3 & 1 \end{bmatrix} \tag{2.21}$$

Pour résoudre le système $A\, x = b$, on procède ensuite en deux étapes :

– l'étape de descente consiste à résoudre $L\, y = b$, c'est-à-dire

$$\begin{bmatrix} 1 & 0 & 0 \\ 3 & 1 & 0 \\ -1 & -3 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} = \begin{bmatrix} 1 \\ -1 \\ 7 \end{bmatrix},$$

dont la solution se calcule par substitutions progressives : $y = [1, -4, -4]^T$;

– l'étape de remontée consiste à résoudre $U\, x = y$; elle est identique à celle que l'on a décrite dans le cadre de l'élimination puisque $y = b^{(3)}$.

Cas général

Dans l'élimination de Gauss pour le système $A\, x = b$, avec $A = A^{(1)} \in M_n(\mathbb{R})$, la première étape s'écrit :

$$A^{(1)}\, x = b^{(1)} \Leftrightarrow A^{(2)}\, x = b^{(2)}$$

de telle façon que la matrice $A^{(2)}$ n'ait que des 0 sous l'élément diagonal $a_{1,1}^{(2)}$ de sa première colonne.

En supposant que $a_{1,1} = a^{(1)}_{1,1} \neq 0$, la matrice $A^{(2)}$ se déduit de $A^{(1)}$ par les relations suivantes :

$$\begin{array}{lcll} a^{(2)}_{1,j} & = & a^{(1)}_{1,j}, & \forall j,\ 1 \leq j \leq n, \\ a^{(2)}_{i,1} & = & 0, & \forall i,\ 2 \leq i \leq n, \\ a^{(2)}_{i,j} & = & a^{(1)}_{i,j} - l_{i,1}\, a^{(1)}_{1,j}, & \forall i, \forall j,\ 2 \leq i, j \leq n. \end{array}$$

où les coefficients $l_{i,1}$ sont définis par $l_{i,1} = \dfrac{a^{(1)}_{i,1}}{a^{(1)}_{1,1}}$. La matrice $A^{(2)}$ s'interprète donc ici aussi comme le produit $A^{(2)} = E^{(1)}\, A^{(1)}$, avec $E^{(1)}$ définie de la façon suivante :

$$E^{(1)} = \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ -l_{2,1} & 1 & \ddots & & \vdots \\ \vdots & 0 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & 1 & 0 \\ -l_{n,1} & 0 & \cdots & 0 & 1 \end{bmatrix}.$$

On vérifie immédiatement que $E^{(1)} = (I - l^1\, e_1^T)$, e_1 désignant le premier vecteur de la base canonique de $\mathbb{R}^n$ et l^1 le vecteur $[0,\, l_{2,1}, \ldots, l_{n,1}]^T$.

La matrice s'écrit :

$$\mathtt{A}^{(2)} = \begin{bmatrix} \mathtt{a}^{(1)}_{1,1} & , & \mathtt{A}^{(1)}(\mathtt{1},\, \mathtt{2:n}) \\ \\ \mathtt{0} & , & \mathtt{A}^{(2)}(\mathtt{2:n},\, \mathtt{2:n}) \end{bmatrix}$$

Si le pivot $p_2 = a^{(2)}_{2,2} \neq 0$, on peut renouveler l'opération pour la matrice $\mathtt{A}^{(2)}(\mathtt{2:n},\, \mathtt{2:n})$ qui appartient à $M_{n-1}(\mathbb{R})$ au moyen d'une matrice $\mathtt{E}^{(2)}(\mathtt{2:n},\, \mathtt{2:n})$ construite de façon analogue à $E^{(1)}$. On complète cette matrice $E^{(2)}$ pour en faire une matrice de $M_n(\mathbb{R})$ telle que le produit à gauche par $E^{(2)}$ laisse invariantes la première ligne et et la première colonne :

$$\mathtt{E}^{(2)} = \begin{bmatrix} \mathtt{1} & , & \mathtt{0} \\ \mathtt{0} & , & \mathtt{E}^{(2)}(\mathtt{2:n},\, \mathtt{2:n}) \end{bmatrix} = \left(I - l^2\, e_2^T\right),$$

avec e_2 le second vecteur de la base canonique de $\mathbb{R}^n$ et $l^2 = [0,\, 0,\, l_{3,2},\, \ldots,\, l_{n,2}]^T$ tel que pour $3 \leq i \leq n$, $l_{i,2} = \dfrac{a^{(2)}_{i,2}}{a^{(2)}_{2,2}}$. On obtient :

$$E^{(2)}\, A^{(2)} = E^{(2)}\ E^{(1)}\ A^{(1)} = \begin{bmatrix} \mathtt{a}^{(1)}_{1,1} & , & \mathtt{a}^{(1)}_{1,2} & , & \mathtt{A}^{(1)}(\mathtt{1},\, \mathtt{3:n}) \\ \mathtt{0} & , & \mathtt{a}^{(2)}_{2,2} & , & \mathtt{A}^{(2)}(\mathtt{2},\, \mathtt{3:n}) \\ \mathtt{0} & , & \mathtt{0} & , & \mathtt{A}^{(3)}(\mathtt{3:n},\, \mathtt{3:n}) \end{bmatrix}$$

avec $\mathtt{A}^{(3)}(\mathtt{3:n},\, \mathtt{3:n}) \in M_{n-2}(\mathbb{R})$: Si au-delà, les $a^{(k)}_{k,k}$ sont tous non nuls, on pourra réitérer l'opération pour obtenir finalement :

$$E^{(n-1)} \cdots E^{(2)}\ E^{(1)}\ A^{(1)} = U.$$

La factorisation $A = L\, U$ cherchée se déduit alors du lemme suivant :

Lemme 2.1 *Soit $E^{(k)}$ la matrice définie pour $1 \leq k \leq n-1$ par $E^{(k)} = I - l^k\, e_k^T$, avec $l^k = [l_{1,k}, \ldots, l_{n,k}]^T$ telle que $l_{i,k} = 0$ si $i \leq k$, et e_k le k-ième vecteur de la base canonique de $\mathbb{R}^n$. Alors,*

(i) la matrice $E^{(k)}$ est inversible d'inverse $L^{(k)} = I + l^k\, e_k^T$,

(ii) le produit $L = L^{(1)} \ldots L^{(n-1)}$ "s'assemble" sans calcul de telle façon que

$$L(i,j) = \begin{cases} 0 & si\ i < j, \\ 1 & si\ i = j, \\ l_{i,j} & si\ i > j \end{cases}$$

En effet, $(I - l^k\, e_k^T)\ (I + l^k\, e_k^T) = I - l^k\ (e_k^T\, l^k)\ e_k^T$ et $e_k^T\, l^k = \displaystyle\sum_{i=k+1}^{n} \delta_{k,i}\, l_{i,k} = 0$.

D'autre part, $(I + l^1\, e_1^T) \ldots (I + l^{n-1}\, e_{n-1}^T) = I + l^1\, e_1^T + \ldots + l^{n-1}\, e_{n-1}^T$ puisque les produits $e_i^T\, l^j$ sont nuls dès que $j \geq i$.

Partant de $E^{(n-1)} \ldots E^{(1)}\, A = U$, on obtient donc :

$$A = L^{(1)} \ldots L^{(n-1)}\, U = L\, U.$$

C'est la factorisation cherchée, avec L triangulaire inférieure d'éléments diagonaux égaux à 1, et dont les élements sous-diagonaux sont les $l_{i,j} = \dfrac{a_{j,j}^{(j)}}{a_{i,j}^{(j)}}$, $1 \leq j < i \leq n$, et U triangulaire supérieure inversible puisque sa diagonale est constituée des pivots qui sont tous différents de 0.

Une condition nécessaire et suffisante :

On admettera le théorème suivant :

Théorème 2.1 *Soit $A = (a_{i,j})_{1 \leq i,j \leq n} \in M_n(\mathbb{R})$; si pour tout k, $1 \leq k \leq n$, la sous-matrice principale $A_k = (a_{i,j})_{1 \leq i,j \leq k}$ de A est telle que $det(A_k) \neq 0$, il existe une matrice triangulaire inférieure L, dont les éléments diagonaux sont égaux à 1, et une matrice triangulaire supérieure inversible U telles que $A = L\, U$. Cette factorisation est unique.*

Cette condition nécessaire et suffisante n'est pas très satisfaisante : on ne peut pas calculer les déterminants mineurs principaux de la matrice d'un système linéaire avant de le résoudre !

Implantation (version élémentaire) :

Nous allons proposer un algorithme du corps d'une fonction qui implante la méthode : on peut "écraser" la matrice A qui lui est passée en argument, car elle alors considérée comme une **variable locale** de la fonction.

A la première étape de la factorisation, la matrice $A = A^{(1)}$ est transformée en $A^{(2)} = (I - l^1\, e_1^T)\, A^{(1)} = A^{(1)} - l^1\, (e_1^T\, A^{(1)})$. Le produit $e_1^T\, A^{(1)}$ est une matrice à une ligne (vecteur ligne) qui est égale à la première ligne de $A^{(1)}$: on a donc

$$A^{(2)} = A^{(1)} - l^1\, A^{(1)}(1, :) \tag{2.22}$$

Le vecteur $l^1 = [0,\, l_{2,1},\, \ldots,\, l_{n,1}]^T$ est tel que :
- la première ligne de $A^{(2)}$ sera égale à la première ligne de $A^{(1)}$,
- les coefficients des lignes 2 à N de la première colonne de $A^{(2)}$ seront nuls. On a donc besoin de ne modifier que la sous-matrice $A^{(1)}(2 : n,\ 2 : n)$; on peut disposer des positions 2 à n de la première colonne de la matrice $A^{(1)}$ pour stocker les composantes non nulles de l^1, c'est-à-dire les $l_{j,1}$, pour $2 \leq j \leq n$.
- dès que les composantes du vecteur l^1 sont calculées, on n'a donc plus à calculer que

$$A^{(2)}(2 : n,\ 2 : n) = A^{(1)}(2 : n,\ 2 : n) - l^1(2 : n)\ A^{(1)}(1,\ 2 : n) \tag{2.23}$$

- si l'on stocke les $l_{j,1}$ non nuls en utilisant les positions $A^{(1)}(2 : n,\ 1)$ inutiles pour la suite, (2.23) devient

$$A^{(2)}(2 : n,\ 2 : n) = A^{(1)}(2 : n,\ 2 : n) - A^{(1)}(2 : n,\ 1)\, A^{(1)}(1,\ 2 : n) \tag{2.24}$$

Le procédé que l'on a décrit va se répéter pour le passage de $A^{(2)}$ à $A^{(3)}$, et cette fois ce sont les positions $A^{(2)}(3 : n,\ 2)$ qui seront utilisées pour stocker les composantes non nulles de l^2 ; seule la sous-matrice $A^{(2)}(3 : n,\ 3 : n)$ sera modifiée par soustraction d'une **matrice de rang 1**, c'est-à-dire une matrice de la forme $M = u\, v^T$ où u et v sont des vecteurs. Et ce même principe sera mis en oeuvre jusqu'à obtention de la factorisation.

A l'issue de cette factorisation, la partie triangulaire supérieure de la matrice A sera occupée par la matrice U, et sa partie triangulaire strictement inférieure par celle de L.

L'algorithme ci-dessous realise cette factorisation de la matrice A, d'ordre n.

```
Pour k de 1 à n − 1
    Si a_{k,k} = 0
        message d'erreur : matrice non inversible
    Pour i de k + 1 à n
        a_{i,k} ⟵ a_{i,k}/a_{k,k}          | colonne k de la matrice L
        Pour j de k + 1 à n                | matrice A^(k)
            a_{i,j} ⟵ a_{i,j} − a_{i,k} ∗ a_{k,j}
```

Cet algorithme est traduit par La fonction MATLAB ci-dessous.

Les commentaires qui suivent immédiatement la ligne de déclaration de la fonction en indiquent l'objet, le format d'appel et la façon de l'utiliser.

```
 function [L, U] = Gauss(A);
%       Gauss calcule la factorisation LU de la matrice A
%       par la methode d'elimination de Gauss.
%       L'appel se fait selon :
%               >> [L, U] = Gauss(A);
%       La solution du systeme Ax = b est donnee par :
%               >> x = U\(L\b);

[n, n] = size(A);
for k = 1:n-1
    if A(k,k) == 0
      error('Matrice non factorisable ...')
    end
    Indices = k+1:n;

    % Colonne k de la matrice L :
    A(Indices,k) = A(Indices,k)/A(k,k) ;

    % Mise a jour de la matrice A^(k) :
    A(Indices,Indices) = A(Indices,Indices)-A(Indices,k)*A(k,Indices) ;
end
L = tril(A,-1) + eye(n);
U = triu(A);
```

Coût de la factorisation

Pour chaque valeur de k, la variable `Indices` contient $n-k$ valeurs ; on doit donc effectuer :

- $n-k$ divisions pour le calcul de la colonne k de L,
- $(n-k)^2$ multiplications pour le calcul de la matrice `A(Indices,k)*A(k,Indices)`, et $(n-k)^2$ soustractions pour la mise à jour de `A(Indices,Indices)`.

Le coût $C_F(n)$ de la factorisation s'obtient en sommant pour k variant de 1 à $n-1$:

$$C_F(n) = \frac{1}{2}n(n-1) + \frac{2}{3}n(n-\frac{1}{2})(n-1) = n(n-1)\frac{4n-1}{6}.$$

On a donc $C_F(n) \sim \dfrac{2\,n^3}{3}$ opérations élémentaires pour l'étape de factorisation.

Pour la résolution d'un système linéaire, on fait suivre cette étape de factorisation par deux étapes de résolution, respectivement d'un système triangulaire inférieur de matrice $L = \texttt{tril(A)} + \texttt{eye(n)}$ et supérieur de matrice $U = \texttt{triu(A)}$:

$$\left\{ \begin{array}{rcl} L\,y & = & b, \\ U\,x & = & y. \end{array} \right.$$

Le coût de la résolution du système reste donc $\mathcal{O}(\frac{2\,n^3}{3})$ opérations.

Retour à l'exemple

On peut observer sur la matrice $A = \begin{bmatrix} 2 & 1 & 1 \\ 6 & 2 & 1 \\ -2 & 2 & 1 \end{bmatrix}$ les différentes étapes de la fonction ci-desssus ; par souci de lisibilité, nous imprimons en caractères gras les valeurs calculées de la matrice L et leur positionnement dans la matrice A :

- pour $k = 1$, la matrice A est remplacée par $A = \begin{bmatrix} 2 & 1 & 1 \\ \mathbf{3} & -1 & -2 \\ \mathbf{-1} & -3 & 2 \end{bmatrix}$,
- pour $k = 2$, la matrice A est remplacée par $A = \begin{bmatrix} 2 & 1 & 1 \\ \mathbf{3} & -1 & -2 \\ \mathbf{-1} & \mathbf{-3} & -4 \end{bmatrix}$.

Les instructions MATLAB

```
L = tril(A,-1) + eye(n);
U = triu(A);
```

permettent bien de retrouver L et U.

2.3.4 Factorisation $P\,A = L\,U$

Principe

Sous réserve d'une permutation de ses lignes, toute matrice inversible est factorisable. Formellement, cette permutation des lignes peut s'écrire comme le produit à gauche de cette matrice par une matrice de permutation.

Théorème 2.2 *Si* $\det(A) \neq 0$*, il existe une matrice* P *de permutation telle que* $P\,A$ *soit factorisable selon* $P\,A = L\,U$*.*

Ce théorème exprime le fait que si A est inversible, on trouvera à chaque étape k de la factorisation, $1 \leq k \leq n-1$, au moins un coefficient $a_{i,k}^{(k)},\ k \leq i \leq n$, non nul.

Un exemple

Pour résoudre le système $A\,x = b$ suivant :

$$\left\{ \begin{array}{rrrrcl} x_1 & +x_2 & +x_3 & +x_4 & = & 1 \\ x_1 & +x_2 & +3x_3 & +3x_4 & = & 3 \\ x_1 & +x_2 & +2x_3 & +3x_4 & = & 3 \\ x_1 & +3x_2 & +3x_3 & +3x_4 & = & 4 \end{array} \right. \iff \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 3 & 3 \\ 1 & 1 & 2 & 3 \\ 1 & 3 & 3 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 3 \\ 4 \end{bmatrix}$$

La première étape de l'élimination conduit au système équivalent :

$$\left\{\begin{array}{rcrcrcrcl} x_1 & + & x_2 & + & x_3 & + & x_4 & = & 1 \\ & & & & 2x_3 & + & 2x_4 & = & 2 \\ & & & & x_3 & + & 2x_4 & = & 2 \\ & & 2x_2 & + & 2x_3 & + & 2x_4 & = & 3 \end{array}\right.$$

On constate que dans ce système, l'inconnue x_2 est déjà éliminée des équations $(Eq.2)$ et $(Eq.3)$, c'est-à-dire que $a^{(2)}_{2,2} = a^{(2)}_{3,2} = 0$. La poursuite du processus d'élimination nécessite une permutation des équations $(Eq.2)$ et $(Eq.4)$:

$$A^{(2)}\,x = b^{(2)} \Longleftrightarrow P_2\,A^{(2)}\,x = P_2\,b^{(2)} \Longleftrightarrow \left\{\begin{array}{rcrcrcrcl} x_1 & + & x_2 & + & x_3 & + & x_4 & = & 1 \\ & & 2x_2 & + & 2x_3 & + & 2x_4 & = & 3 \\ & & & & x_3 & + & 2x_4 & = & 2 \\ & & & & 2x_3 & + & 2x_4 & = & 2 \end{array}\right.$$

La matrice P_2 qui multiplie à gauche chaque membre du système d'équations est la matrice d'une permutation élémentaire. On peut l'écrire

$$P_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$$

et on vérifie immédiatement que

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 1 & 2 \\ 0 & 2 & 2 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 2 & 2 & 2 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 2 & 2 \end{bmatrix}$$

On poursuit l'élimination sans problème pour obtenir

$$\left\{\begin{array}{rcrcrcrcr} x_1 & + & x_2 & + & x_3 & + & x_4 & = & 1 \\ & & 2x_2 & + & 2x_3 & + & 2x_4 & = & 3 \\ & & & & x_3 & + & 2x_4 & = & 2 \\ & & & & & - & 2x_4 & = & -2 \end{array}\right.$$

On a obtenu un système triangulaire inversible.

Ecriture matricielle de la factorisation avec permutation

Le système triangulaire obtenu ci-dessus s'écrit $U\,x = b^{(4)}$, avec $b^{(4)} = [1,\ 3,\ 2,\ -2]^T$ et $U = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 2 & 2 & 2 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & -2 \end{bmatrix}$.

La matrice U de ce système est définie par :

$$E^{(3)}\,E^{(2)}\,P_2\,E^{(1)}\,A = U, \tag{2.25}$$

avec, pour chaque k, $1 \leq k \leq 3$, $E^{(k)} = I - l^k\,e_k^T$, et
- $l^1 = [0,\,-1,\,-1,\,-1]^T$,
- $l^2 = [0,\,0,\,0,\,0]^T$, puisque les coefficients $a_{3,2}^{(2)}$ et $a_{4,2}^{(2)}$ sont tous deux nuls,
- $l^3 = [0,\,0,\,0,\,-2]^T$.

La matrice P_2 s'écrit également sous une forme $P_2 = I - u\,u^T$, avec $u = e_2 - e_4$ (c'est la différence des deux vecteurs de la base canonique correspondant aux indices des lignes à permuter) ! Elle est telle que $P_2^2 = I$ (vérification immédiate). On peut alors glisser P_2^2 à droite de $E^{(1)}$ dans les deux membres de (2.25) :

$$E^{(3)}\,E^{(2)}\,P_2\,E^{(1)}\,P_2\,P_2\,A = U.$$

Le produit matriciel est associatif : on peut isoler les facteurs

$$P_2\,E^{(1)}\,P_2 = P_2\,(I - l^1\,e_1^T)\,P_2 = P_2^2 - P_2\,l^1\,e_1^T\,P_2 = I - (P_2\,l^1)\,(e_1^T\,P_2),$$

et calculer :

- $e_1^T\,P_2 = (1,\,0,\,0,\,0)\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} = e_1^T$,
- $P_2\,l^1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} 0 \\ l_{2,1} \\ l_{3,1} \\ l_{4,1} \end{bmatrix} = \begin{bmatrix} 0 \\ l_{4,1} \\ l_{3,1} \\ l_{2,1} \end{bmatrix} = \widetilde{l^1}$

de sorte que finalement

$$E^{(3)}\,E^{(2)}\,\widetilde{E^{(1)}}\,P_2\,A = U,$$

d'où l'on tire, de façon analogue à ce qu'on a vu précédemment, en notant P la matrice P_2 :

$$P\,A = \widetilde{L^{(1)}}\,L^{(2)}\,L^{(3)}\,U = L\,U \tag{2.26}$$

avec $\mathtt{L} = [\widetilde{\mathtt{l^1}},\,\mathtt{l^2},\,\mathtt{l^3},\,\mathtt{l^4}]$.

Pour représenter cette factorisation, il est inutile de construire la matrice P. Il suffit d'adjoindre à la matrice A un compteur de permutation sous forme d'une colonne supplémentaire, p, initialisée à l'ordre naturel :

$$[A\,|\,p] \quad = \quad \left[\begin{array}{cccc|c} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 3 & 3 & 2 \\ 1 & 1 & 2 & 3 & 3 \\ 1 & 3 & 3 & 3 & 4 \end{array}\right] \quad \Longrightarrow \quad \left[\begin{array}{cccc|c} 1 & 1 & 1 & 1 & 1 \\ \mathbf{1} & 0 & 2 & 2 & 2 \\ \mathbf{1} & 0 & 1 & 2 & 3 \\ \mathbf{1} & 2 & 2 & 2 & 4 \end{array}\right]$$

$$\Longrightarrow \quad \left[\begin{array}{cccc|c} 1 & 1 & 1 & 1 & 1 \\ \mathbf{1} & 2 & 2 & 2 & 4 \\ \mathbf{1} & \mathbf{0} & 1 & 2 & 3 \\ \mathbf{1} & \mathbf{0} & 2 & 2 & 2 \end{array}\right] \quad \Longrightarrow \quad \left[\begin{array}{cccc|c} 1 & 1 & 1 & 1 & 1 \\ \mathbf{1} & 2 & 2 & 2 & 4 \\ \mathbf{1} & \mathbf{0} & 1 & 2 & 3 \\ \mathbf{1} & \mathbf{0} & \mathbf{2} & -2 & 2 \end{array}\right]$$

On vérifie en effet que

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 0 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & 2 & 2 & 2 \\ 0 & 0 & 1 & 2 \\ 0 & 0 & 0 & -2 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 3 & 3 & 3 \\ 1 & 1 & 2 & 3 \\ 1 & 1 & 3 & 3 \end{bmatrix} = P\,A.$$

Le compteur de permutation $p = [1,\, 4,\, 3,\, 2]^T$ permet de retrouver la permutation des lignes que l'on doit effectuer pour résoudre le système $P\,A = L\,U = P\,b$.

Généralisation

Considérons à présent une matrice inversible $A \in M_n(\mathbb{R})$. De même que ci-dessus à la seconde étape de la factorisation, si on rencontre un pivot nul à l'étape k, la permutation de la ligne k avec une ligne $i > k$ telle que $a_{i,k}^{(k)} \neq 0$ s'écrira $P_k = I - u\,u^T$, avec $u = e_k - e_i$.

Pour tout $j < k$, on aura

$$P_k\,E^{(j)}\,P_k = I - P_k\,l^j\,e_j^T\,P_k = I - \widetilde{l^j}\,e_j^T = \widetilde{E^{(j)}},$$

puis

$$\begin{array}{rcl} P_k\,E^{(k-1)}\,\ldots\,E^{(1)} & = & (P_k\,E^{(k-1)}\,P_k)\,(P_k\,\ldots\,Pk)\,(P_k\,E^{(1)}\,P_k)\,P_k \\ & = & \widetilde{E^{(k-1)}}\,\ldots\,\widetilde{E^{(1)}}\,P_k \end{array}.$$

En introduisant des matrices P_k à chaque étape, avec éventuellement $P_k = I$ lorsqu'aucune permutation n'est nécessaire, on obtiendra

$$\left(E^{(n-1)}\,\widetilde{E^{(n-2)}}\,\ldots\,\widetilde{E^{(1)}}\right)\,(P_{n-1}\,\ldots\,P_1)\;A = U,$$

puis

$$P\,A = L\,U.$$

Stratégie de pivot partiel

La factorisation $P\,A = L\,U$ est nécessaire lorsqu'on rencontre un pivot nul. Il se peut que l'on rencontre un pivot qui sans être nul est très petit. Dans ce cas, l'effet des erreurs d'arrondi peut-être catastrophique.

Les instructions MATLAB ci-dessous utilisent la fonction `Gauss` listée précédement pour calculer la factorisation $A = L\,U$ d'une matrice pour laquelle $a_{1,1}$ est très petit, sans être nul :

```
n = 5 ;
A = rand(n) + eye(n) ;
A(1,1) = 2*eps ;
[L, U] = Gauss(A) ;
B = L*U ;
erreur = norm(A-B)
```

La fonction `eps` de MATLAB fournit le double de l'unité d'arrondi de l'arithmétique avec laquelle on travaille. Ici, `eps = 2.2204e-16`. Pour une réalisation j'ai obtenu :

```
>> erreur = 0.1940
```

En divisant par un pivot très petit, on augmente le risque d'erreurs. En effet, si à l'étape k de la factorisation, le pivot $a_{k,k}^{(k)}$ est très petit, les coefficients $l_{i,k}$, $k \leq i \leq n$ seront en général très grand.

Les lignes de la matrice $A^{(k+1)}$ sont définies par l'expression MATLAB

$$\mathtt{A(i,k+1:n)} - \mathtt{l(i,k)} * \mathtt{A(k,k+1:n)};$$

Le premier terme sera négligeable, et ces lignes seront presque dépendantes, puisqu'à peu près proportionnelles à `A(k,k+1:n)` ! En terme de méthode d'élimination, le système $A^{(k+1)}\, x = b^{(k+1)}$ sera très sensible aux perturbations, puisque proche d'un système non inversible.

Pour limiter ce risque, on utilise en général une stratégie de recherche du pivot maximal dans chaque colonne. A l'étape k, plutôt que de rechercher le premier coefficient non nul parmi les $a_{i,k}^{(k)}$, $i \geq k$, on recherche l'indice i_m tel que

$$a_{i_m,k}^{(k)} = \max_{k \leq i \leq n} \left\{ \left| a_{i,k}^{(k)} \right| \right\}.$$

On permute ensuite les lignes k et i_m.

On remarquera qu'il est indispensable de conserver la mémoire des permutations effectuées, de façon à pouvoir les effectuer ensuite sur le vecteur second membre B.

On le fait au moyen d'un tableau monodimensionnel p initialisé par $p_i = i$, pour $1 \leq i \leq n$. L'algorithme de cette décomposition avec permutation s'obtient facilement en modifiant l'algorithme de décomposition sans permutation, pour obtenir l'algorithme suivant :

```
Pour k de 1 à n − 1
    Recherche de i_m tel que a_{i_m,k} = max{|a_{i,k}|, i = k : n}
    Si a_{i_m,k} = 0
        message d'erreur : matrice non inversible. FIN
    A(k,:)  ⟷  A(i_m,:)
    p_k  ⟷  p_{i_m}
    Pour i de k + 1 à n
        a_{i,k} ⟵ a_{i,k}/a_{k,k}                    | colonne k de L
        Pour j de k + 1 à n                          |
            a_{i,j} ⟵ a_{i,j} − a_{i,k} ∗ a_{k,j}    | matrice A^(k)
```

La fonction MATLAB ci-dessous met en oeuvre cette stratégie :

```
function [L, U, permute] = Gausspiv(A);
%       Gauss calcule la factorisation LU de la matrice
%       A avec strategie de pivot partiel.
%       L'appel se fait selon :
%               >> [L, U, permute] = Gauss(A);
%       Elle est telle que A(permute, :) = L*U.
%       La solution du systeme  Ax = b est donnee par :
%              >> x = U \(L \(b(permute, :)) ;

[n, n] = size(A);
permute = 1:n;
for k = 1:n-1

  % Recherche du pivot, permutation des lignes correspondantes
    [maxi, ligne] = max(abs(A(k:n,k)));
    ligne = ligne+k-1;
    A([k, ligne],:) = A([ligne, k],:);
    permute([k, ligne]) = permute([ligne, k]);

    if A(k,k) == 0
      error('Matrice non inversible ...')
    end

    % Colonne k de la matrice L) :
    A(k+1:n,k) = A(k+1:n,k)/A(k,k) ;

    % Mise a jour de la matrice A^(k) :
    A(k+1:n,k+1:n) = A(k+1:n,k+1:n)-A(k+1:n,k)*A(k,k+1:n) ;
end
L = tril(A,-1) + eye(n);
U = triu(A);
```

Cette stratégie est considérée comme suffisament stable par la plupart des bibliothèques scientifiques ; c'est celle qui est généralement implantée, en particulier dans MATLAB par la fonction `lu`.

Un exemple

On peut suivre les étapes de l'exécution de la fonction précédente sur la matrice

$$[\,\mathtt{A}\,|\,\mathtt{p}\,] = \left[\begin{array}{cccc|c} 1 & 2 & 4 & 17 & 1 \\ 3 & 6 & -12 & 3 & 2 \\ 2 & 3 & -3 & 2 & 3 \\ 0 & 2 & -2 & 6 & 4 \end{array}\right]$$

$$\texttt{k = 1 :}\quad \left[\begin{array}{cccc|c} 3 & 6 & -12 & 3 & 2 \\ 1 & 2 & 4 & 17 & 1 \\ 2 & 3 & -3 & 2 & 3 \\ 0 & 2 & -2 & 6 & 4 \end{array}\right] \Longrightarrow \left[\begin{array}{cccc|c} 3 & 6 & -12 & 3 & 2 \\ \mathbf{1/3} & 0 & 8 & 16 & 1 \\ \mathbf{2/3} & -1 & 5 & 0 & 3 \\ \mathbf{0} & 2 & -2 & 6 & 4 \end{array}\right]$$

$$\texttt{k = 2 :}\quad \left[\begin{array}{cccc|c} 3 & 6 & -12 & 3 & 2 \\ \mathbf{0} & 2 & -2 & 6 & 4 \\ \mathbf{2/3} & -1 & 5 & 0 & 3 \\ \mathbf{1/3} & 0 & 8 & 16 & 1 \end{array}\right] \Longrightarrow \left[\begin{array}{cccc|c} 3 & 6 & -12 & 3 & 2 \\ \mathbf{0} & 2 & -2 & 6 & 4 \\ \mathbf{2/3} & \mathbf{-1/2} & 4 & 3 & 3 \\ \mathbf{1/3} & \mathbf{0} & 8 & 16 & 1 \end{array}\right]$$

$$\texttt{k = 3 :}\quad \left[\begin{array}{cccc|c} 3 & 6 & -12 & 3 & 2 \\ \mathbf{0} & 2 & -2 & 6 & 4 \\ \mathbf{1/3} & \mathbf{0} & 8 & 16 & 1 \\ \mathbf{2/3} & \mathbf{-1/2} & 4 & 3 & 3 \end{array}\right] \Longrightarrow \left[\begin{array}{cccc|c} 3 & 6 & -12 & 3 & 2 \\ \mathbf{0} & 2 & -2 & 6 & 4 \\ \mathbf{1/3} & \mathbf{0} & 8 & 16 & 1 \\ \mathbf{2/3} & \mathbf{-1/2} & \mathbf{1/2} & -5 & 3 \end{array}\right]$$

On vérifie que

$$\left[\begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1/3 & 0 & 1 & 0 \\ 2/3 & -1/2 & 1/2 & 1 \end{array}\right] \left[\begin{array}{cccc} 3 & 6 & -12 & 3 \\ 0 & 2 & -2 & 6 \\ 0 & 0 & 8 & 16 \\ 0 & 0 & 0 & -5 \end{array}\right] = \left[\begin{array}{cccc} 3 & 6 & -12 & 3 \\ 0 & 2 & -2 & 6 \\ 1 & 2 & 4 & 17 \\ 2 & 3 & -3 & 2 \end{array}\right] = P\,A.$$

La matrice $P\,A$ ne se calcule pas par un produit matriciel. A l'issue de l'éxécution de cet algorithme, on a $p = [2,\ 4,\ 1,\ 3]^T$ et $P\,A = A(p,\ :)$.

Stratégie de pivot total

Les systèmes qui la rendent indispensable sont très rares , et elle n'est que rarement implantée. La factorisation peut alors s'écrire :

$$P\,A\,Q = L\,U$$

La matrice Q est ici également une matrice de permutation ; son effet est de permuter les colonnes de A. Il faut alors conserver la mémoire de cette permutation pour replacer les inconnues du système dans le bon ordre.

Unicité de la factorisation

On suppose que la matrice A est "factorisable" selon $P\,A = L\,U$, les éléments diagonaux de L étant tous égaux à 1. Supposant que l'on puisse former deux factorisations différentes, on écrit :

$$P\,A = L_1\,U_1 = L_2\,U_2.$$

On peut alors former :

$$M = L_2^{-1}\,L_1 = \ U_2 U_1^{-1}$$

$L_2^{-1}\,L_1$ est triangulaire inférieure de diagonale unité, et $U_2 U_1^{-1}$ est triangulaire supérieure : toutes deux sont égales à l'identité, donc $L_1 = L_2$ et $U_1 = U_2$. Cette décomposition est unique.

Factorisation $L\,D\,R$

Il suffit d'écrire :

$$\begin{bmatrix} u_{1,1} & & & \\ & \ddots & u_{i,j} & \\ & & \ddots & \\ & 0 & & u_{n,n} \end{bmatrix} = \begin{bmatrix} u_{1,1} & & & \\ & \ddots & 0 & \\ & & \ddots & \\ & 0 & & u_{n,n} \end{bmatrix} \begin{bmatrix} 1 & & & \\ & \ddots & \frac{u_{i,j}}{u_{i,i}} & \\ & & \ddots & \\ & 0 & & 1 \end{bmatrix}$$

pour obtenir $U = D\,R$

2.4 Matrices symétriques définies positives

2.4.1 Matrices symétriques définies positives

Parmi les problèmes qui conduisent à un système linéaire où la matrice est symétrique, les plus intéressants sont ceux pour lesquels on n'a pas besoin de stratégie de pivot ; c'est le cas lorsque la matrice est symétrique définie positive.

2.4.2 Factorisation de Choleski

Théorème 2.3 *Une condition nécessaire et suffisante pour que A soit symétrique définie positive est qu'il existe B, triangulaire inférieure inversible, unique si ses éléments diagonaux sont choisis positifs, telle que $A = B\,B^T$.*

On vérifie sans problème la condition suffisante : si $A = B\,B^T$, alors $x^T\,A\,x = x^T\,B\,B^T\,x = \left\|B^T\,x\right\|^2 > 0$ si $x \neq 0$.

Pour la condition nécessaire, on raisonne par récurrence en vérifiant que le théorème est vrai pour $n = 1$, puisque $a \in \mathbb{R}^n$ n'est strictement positive que si et seulement si il existe $b > 0$ tel que $b^2 = a$, et en l'occurence, $b = \sqrt{a}$.

On suppose alors la condition vérifiée jusqu'à l'ordre $n-1$, et on écrit par blocs la matrice A selon :

$$A = \begin{bmatrix} A_{n-1} & a \\ a^T & m^2 \end{bmatrix}$$

où $A_{n-1} \in M_{n-1}(\mathbb{R})$ (sous-matrice principale de A) est également symétrique définie positive, $a \in \mathbb{R}^{n-1}$ et $a_{n,n} = m^2$ est strictement positif.

On cherche B de la forme :

$$B = \begin{bmatrix} B_{n-1} & 0 \\ b^T & b_{n,n} \end{bmatrix}$$

telle que :

$$B\,B^T = \begin{bmatrix} B_{n-1} & 0 \\ b^T & b_{n,n} \end{bmatrix} \begin{bmatrix} B_{n-1}^T & b \\ 0 & b_{n,n} \end{bmatrix} = A$$

c'est-à-dire :

$$\begin{bmatrix} B_{n-1}\,B_{n-1}^T & B_{n-1}b \\ b^T B_{n-1}^T & b^T\,b + b_{n,n}^2 \end{bmatrix} = \begin{bmatrix} A_{n-1} & a \\ a^T & m^2 \end{bmatrix}$$

Par hypothèse de récurrence, on peut trouver B_{n-1} triangulaire inférieure inversible telle que $A_{n-1} = B_{n-1}\,B_{n-1}^T$, et le vecteur $b \in \mathbb{R}^{n-1}$ est solution du système inversible $B_{n-1}b = a$.

Le seul problème est donc le calcul de $b_{n,n}$, qui n'est possible que si $m^2 - b^T\,b > 0$.

$$m^2 - b^T\,b = m^2 - \left(B_{n-1}^{-1}\,a\right)^T\,\left(B_{n-1}^{-1}\,a\right) = m^2 - a^T\,A_{n-1}^{-1}\,a > 0$$

En effet, A est symétrique définie positive : en choisissant le vecteur X qui s'écrit par blocs $X = \begin{bmatrix} A_{n-1}^{-1}\,a \\ -1 \end{bmatrix}$, avec $A_{n-1}^{-1} \in M_{n-1}(\mathbb{R})$ on vérifie que $X^T\,A\,X = m^2 - a^T\,A_{n-1}^{-1}\,a$.

2.4.3 Algorithme

Nous pouvons déduire cet algorithme de la démonstration du théorème 2.3 : le raisonnement par récurrence, dans cette démonstration, permet d'affirmer que si $A_k = (a_{i,j})_{1\leq i,j\leq k}$ désigne la sous-matrice principale de A, et B_k celle de B, alors $A_k = B_k\,B_k^T$.

Le passage de B_{k-1} à B_k se fait en deux étapes :

- calcul de $b^T = \texttt{B(k, 1 : k - 1)}$ en résolvant le système triangulaire inférieur $B_{k-1}\,b = a$, avec $a = \texttt{A(1 : k - 1, k)}$;
- calcul de $b_{k,\,k} = \sqrt{a_{k,k} - b^T\,b}$.

L'algorithme de factorisation, peut donc s'écrire :

$b_{1,1} = \sqrt{a_{1,1}}$
Pour k **de** 2 **à** n
  Pour ℓ **de** 1 **à** $k-1$ | *Résolution du système triangulaire inférieur*
    $b_{k,\ell} \longleftarrow \left(a_{k,\ell} - \sum_{j=1}^{\ell-1} b_{\ell,j}\,b_{k,j}\right) \Big/ b_{\ell,\ell}$
  $b_{k,k} \longleftarrow \sqrt{a_{k,k} - \sum_{j=1}^{k-1} b_{k,j}^2}$ | *Calcul de l'élément diagonal*

Ce qui donne, en langage MATLAB :

```
N = size(A,1);
B = zeros(size(A));
B(1,1) = sqrt(A(1,1));
for k = 2:N
   % Résolution du système triangulaire inférieur :
   for l = 1:k-1
       B(k,l) = (A(k,l)-B(l,1:l-1)*B(k,1:l-1)')/B(l,l);
   end;
   % Calcul de l'élément diagonal
   B(k,k) = sqrt(A(k,k)-B(k,1:k-1)*B(k,1:k-1)');
end;
```

Remarque 2.5 *On peut tout aussi bien énoncer la factorisation de Choleski de la matrice symétrique définie positive A selon $A = H^T H$ avec H triangulaire supérieure. C'est généralement le cas dans la littérature anglo-saxonne, et c'est en particulier l'implantation de la fonction* `chol` *de* MATLAB.

Complexité

On peut s'aider du programme MATLAB ci-dessus pour calculer le coût de la factorisation. Pour chaque valeur de k comprise entre 2 et n, on effectue, sans compter les extractions de racines carrées :

- une résolution de système triangulaire de taille $k-1$, soit $(k-1)^2$ opérations élémentaires,
- $2\,(k-1)$ opérations élémentaires pour le calcul de l'élément diagonal.

Le coût de la factorisation est donc

$$\begin{aligned} C(n) &= \sum_{l=1}^{n-1} l^2 + 2\sum_{l=1}^{n-1} l, \\ &= \frac{n\,(n-1)\,(2n-1)}{6} + n\,(n-1), \\ &= \frac{n\,(n-1)\,(2n+5)}{6}, \end{aligned}$$

auquel s'ajoutent n calculs de racines. On retiendra que $C(n) = \mathcal{O}(\frac{n^3}{3}$.

2.5 Systèmes surdéterminés

2.5.1 Un problème d'identification de paramètres

Le défenseur d'une région a pour mission d'intercepter d'éventuels missiles. Il dispose de radars relevant les positions de missiles agresseurs, et doit prévoir leur trajectoire pour les intercepter.

Ses radars relèvent les positions suivantes d'un missile :

x_i	0	250	500	750	1000
y_i	0	8	15	19	20

Les x_i désignent la distance au sol sur une trajectoire directe, et les y_i l'altitude du missile. Ces valeurs sont exprimées en kilomètres.

On suppose aussi que ce défenseur possède un très bon service de renseignements qui aura pu lui affirmer que la trajectoire des missiles est parabolique. Elle satisfait donc une équation de la forme

$$y(x) = a_0 + a_1\, x + a_2\, x^2. \tag{2.27}$$

On obtient alors un système d'équations dont les inconnues sont $a_0,\ a_1$ et a_2 ; les équations qui le constituent s'écrivent

$$a_0 + \ x_i a_1 + x_i^2\, a_2 = y_i,$$

pour chacun des couples $(x_i,\ y_i)$ du tableau ci-dessus.

En choisissant comme unité la centaine de kilomètres, on obtient :

$$\left\{\begin{array}{lcccccl} a_0 & & & & & = & 0, \\ a_0 & + & 2.5 a_1 & + & 6.25\, a_2 & = & 0.08, \\ a_0 & + & 5\, a_1 & + & 25\, a_2 & = & 0.15, \\ a_0 & + & 7.5\, a_1 & + & 56.25\, a_2 & = & 0.19, \\ a_0 & + & 10\, a_1 & + & 100\, a_2 & = & 0.20. \end{array}\right. \tag{2.28}$$

Mais ce système a 5 équations pour seulement 3 inconnues. On peut l'écrire sous forme matricielle,

$$A\, a = y,$$

où la matrice A est de taille 5×3, le vecteur $a = (a_0,\ a_1,\ a_2)^T$ est dans $\mathbb{R}^3$, et le vecteur y des altitudes relevées est dans $\mathbb{R}^5$

Ce système n'est pas inversible, et on ne peut donc raisonnablement pas espérer en calculer la solution. Au moins peut on espérer calculer un vecteur a tel que les vecteurs $A\, a \in \mathbb{R}^5$ et y soient assez proches. On cherchera en fait à rendre $\|A\, a - y\|$ aussi petite que possible.

2.5.2 Approximation au sens des moindres carrés

Définition

Plaçons nous dans le cas général : on se donne $A \in M_{n,p}(\mathbb{R})$ avec $n > p$, $b \in \mathbb{R}^n$ et on cherche $x^* \in \mathbb{R}^p$ tel que

$$\forall x \in \mathbb{R}^p,\ \|A\, x^* - b\| \leq \|A\, x - b\|. \tag{2.29}$$

Toutes les normes de $\mathbb{R}^n$ sont équivallentes, mais aucune n'est différentiable. Par contre, toute norme étant positive,

$$\|A\,x^* - b\| \leq \|A\,x - b\| \Leftrightarrow \|A\,x^* - b\|^2 \leq \|A\,x - b\|^2.$$

Le carré de la norme euclidienne étant un polynôme est de classe $\mathcal{C}^\infty$: on peut rechercher un point de minimum en recherchant un zéro de sa dérivée.

Une autre particularité de la norme euclidienne est d'être associée à un produit scalaire : le théorème de Pythagore fournit une autre caractérisation d'un point de minimum.

Le **problème de moindres carrés** consiste à chercher $x^* \in \mathbb{R}^n$ qui minimise $\|A\,x - b\|_2^2$: en effet, si $e = A\,x^* - b$ avec $e = [e_1,\ e_2, \ldots\ , e_m]^T$, la solution x^* réalise le minimum de $\|e\|_2^2 = e_1^2 + e_2^2 + \ldots + e_m^2$.

Approche par calcul différentiel

L'intérêt du choix de la norme euclidienne tient au lemme suivant qui v a permettre de caractériser simplement la solution.

Lemme 2.2 *Soit $J : \mathbb{R}^p \to \mathbb{R}$ l'application définie par $J(x) = \|A\,x - b\|_2^2$. J est deux fois différentiable, et*

1. *le gradient de J est donné par $\nabla_J(x) = 2\left(A^T\,A\,x - A^T\,b\right)$,*
2. *la matrice Hessiene de J est $H_J(x) = 2\,A^T\,A$.*

On note $\langle x,\, y\rangle = x^T\,y$ le produit scalaire euclidien dans $\mathbb{R}^n$.

On peut démontrer le lemme en calculant directement un développement à l'ordre 2 de J au voisinage d'un point x. Soient x et h deux vecteurs de $\mathbb{R}^p$,

$$\begin{aligned}
J(x+h) &= \langle A\,(x+h) - b,\ A\,(x+h) - b\rangle\,, \\
&= \langle (A\,x - b) + A\,h,\ (A\,x - b) + A\,h\rangle\,, \\
&= J(x) + \langle A\,h,\ A\,x - b\rangle + \langle A\,x - b,\ A\,h\rangle + \langle A\,h,\ A\,h\rangle, \\
&= J(x) + 2\,\langle A^T\,(A\,x - b),\, h\rangle + \langle A^T\,Ah,\, h\rangle.
\end{aligned}$$

On reconnait le développement à l'ordre 2 d'une fonction de classe $\mathcal{C}^2$,

$$f(x+h) = f(x) + \langle \nabla_f(x),\, h\rangle + \frac{1}{2}\,\langle H_f(x)\,h,\, h\rangle + \|h\|_2^2\,\epsilon(h).$$

La condition $\nabla_J(x^*) = 0$ entraine que x^* est un extrémum local. Le développement obtenu est exact, la matrice $A^T\,A$ est symétrique positive, donc cet extremum est un minimum global. Il sera unique si $A^T\,A$ est **définie** positive, ce qui est assuré lorsque $\mathrm{rang}(A) = p$.

Théorème 2.4 *Soient $A \in M_{n,p}(\mathbb{R})$ avec $n > p$ et $b \in \mathbb{R}^n$. La solution x^* du problème des moindres carrés défini par A et b est donnée par le système des* **équations normales**

$$A^T A\, x^* = A^T b. \tag{2.30}$$

Elle est unique si et seulement si $\mathrm{rang}(A) = p$.

Approche par le théorème de Pythagore

Suivant ce théorème, $\|r\|_2$ sera minimale si et seulement si $r \perp \mathrm{Im}(A)$. Ce raisonnement est illustré par la figure 2.4 dans le cas où $n = 2$ et $p = 1$.

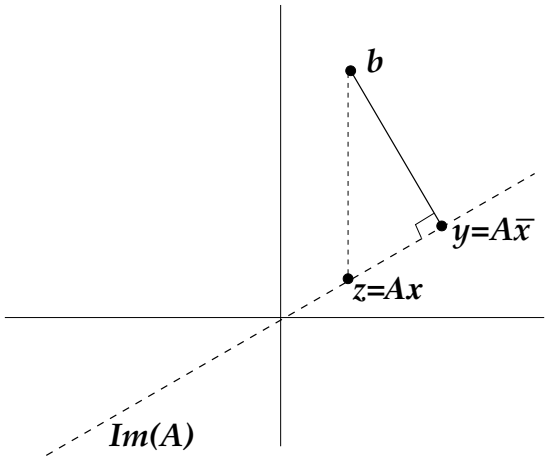


FIGURE 2.4 – Illustration du théorème de Pythagore.

En effet, $\|b - A\, x^*\|_2$ est minimale si et seulement si $b - A\, x^* \perp \mathrm{Im}(A)$: pour tout $z = A\, x$ tel que $b - z$ n'est pas perpenduculaire à $\mathrm{Im}(A)$, on aura alors

$$\|b - A\, x\|_2^2 = \|b - A\, x^*\|_2^2 + \|A\, x^* - z\|_2^2 \geq \|b - A\, x^*\|_2^2 = \|r^*\|_2^2.$$

Mais par ailleurs,

$$\begin{array}{rcl} r \in \mathrm{Im}(A)^\perp & \Leftrightarrow & \forall y \in \mathrm{Im}(A),\ r^T y = 0, \\ & \Leftrightarrow & \forall z \in \mathbb{R}^p,\ r^T A\, z = 0, \\ & \Leftrightarrow & \forall z \in \mathbb{R}^p,\ z^T A^T r = 0, \\ & \Leftrightarrow & A^T r = 0, \\ & \Leftrightarrow & r \in \mathrm{Ker}(A^T). \end{array}$$

On obtient donc bien encore le système des équations normales,

$$A^T \overline{r} = A^T\, (A\, x^* - b)\, = 0.$$

2.5.3 La solution de l'exemple

Le système des équations normales associé à notre exemple s'écrit

$$\begin{bmatrix} 5 & 25 & 187.5 \\ 25.00 & 187.5 & 1562.5 \\ 187.50 & 1562.5 & 13828.125 \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \alpha_2 \end{bmatrix} = \begin{bmatrix} 0.62 \\ 4.375 \\ 34.9375 \end{bmatrix}.$$

C'est un système de 3 équations à 3 inconnues que l'on peut résoudre par la méthode d'élimination de Gauss. Sa solution est donnée approximativement par

$$\alpha \simeq \begin{bmatrix} -0.0023 \\ 0.0398 \\ -0.0019 \end{bmatrix}.$$

On peut maintenant tracer la trajectoire du missile, et on peut constater sur la figure 3.5 que la parabole obtenue interprète bien les mesures faites par le radar.

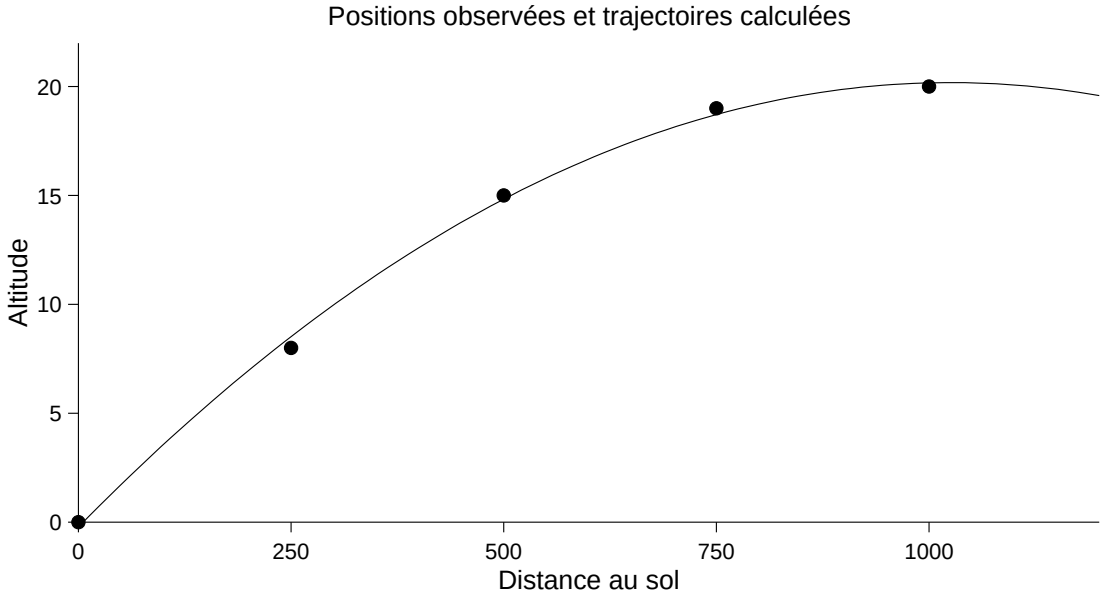


FIGURE 2.5 – Parabole des moindres carrés

2.5.4 Gauss, la planète perdue et les échaffaudages

La trajectoire de ce faux missile partant droit au ciel aurait peut-être croisé celle de la fameuse planète perdue. L'anecdote qui suit est racontée par Carl D. Meyer dans son excellent livre : *Matrix Analysis and Applied Linear Algebra*, SIAM , 2000.

En observant la constellation du taureau, le 1er janvier 1801, l'astronome italien Guiseppe Piazzi remarqua une petite "étoile" inconnue jusqu'alors. Il continua ainsi

que quelques collègues à observer cette “étoile”, et ils remarquèrent qu'elle n'était pas fixe. On conclut donc qu'il s'agissait d'une planète.

Mais cette nouvelle planète disparut complètement à l'automne 1801. Les plus grands astronomes de l'époque unirent en vain leurs efforts pour tenter de la retrouver.

En septembre 1801, le mathématicien allemand Carl Friedrich Gauss alors agé de 24 ans releva le défi de localiser cette planète perdue. Il commença par faire l'hypothèse d'une orbite elliptique, au lieu de la supposer circulaire selon le modèle utilisé à l'époque.

C'est à cette occasion qu'il développa la méthode des moindres carrés dont il avait élaboré les concepts fondamentaux en 1795.

En décembre, il avait gagné son pari. Non seulement il avait réussi à localiser la planète perdue, mais il put prévoir sa position future. C'est ainsi que les astronomes Olbers et De Zach la retrouvèrent. C'était en fait un astéroïde, qu'on appela Céres.

Cette extraordinaire réussite de Gauss, de pouvoir prévoir la trajectoire d'un objet aussi éloigné avec si peu de données, étonna la communauté scientifique et contribua à établir sa réputation de mathématicien génial. Il refusa cependant de révéler les détails de sa méthode, et certains l'accusèrent de sorcellerie.

Gauss ne publia qu'en 1809 un exposé exhaustif et très concis de la méthode des moindres carrés, sans doute poussé par le fait que le mathématicien français Adrien-Marie Legendre l'avait obtenue indépendament en 1805. Cela reflètait bien sa personnalité. Il ne publiait en effet que des théories abouties, sans jamais révéler les cheminements qui l'avaient mené jusqu'à elles.

Quand on lui en faisait le reproche, il aimait répondre que les architectes qui avaient bâti les magnifiques cathédrales européennes avaient toujours enlévé les échaffaudages qui avaient permis leur construction, afin que l'on pût mieux en apprécier la beauté.

Chapitre 3

ÉQUATIONS NON LINÉAIRES

Dans le cas général, on ne peut pas espérer résoudre une équation ou un système d'équations non linéaires par une formule directe : ça n'est possible que pour quelques équations algébriques, comme par exemple les équations du second degré.

La formule qui donnne les deux racines des équations du second degré est connue depuis très longtemps ; elle a été retrouvée sur des tablettes babyloniennes datées des environs de 1500 avant notre ère. Pour les équations de degré 3 la solution a été établie par l'italien Tartaglia en 1535, mais publiée en 1545 par Jérôme Cardan, qui par ailleurs a laissé son nom à un système de transmission cher aux automobilistes. Pour le degré 4, c'est encore un italien, dont le nom sonne de façon plaisante aux oreilles des amateurs d'automobiles qui a établi la méthode (Ludovico Ferrari en 1565). Ces formules sont exactes si l'on considère que l'extraction d'une racine est une opération exacte. Et on sait depuis Niels Abel et Evariste Galois que l'on ne peut pas étendre ce type de formules au calcul des racines d'un polynôme quelconque de degré supérieur.

Mais si les calculettes nous fournissent immédiatement la valeur $\alpha = \sqrt{a}$ pour un réel a positif, cette opération est en fait la résolution de l'équation non linèaire $x^2 - a = 0$, et s'obtient en calculant de proche en proche les itérés d'une suite qui tend vers α.

De façon plus générale, on peut être amené à rechercher une racine réelle d'une équation $F(x) = 0$, où F est une fonction quelconque. On devra construire une suite $(x_k)_{k\in\mathbb{N}}$ qui converge vers une solution du problème. Il est important de remarquer que l'on parle ici d'une solution : on n'a généralement pas unicité de solution, et il arrive que l'on en cherche plusieurs, ou qu'on les cherche toutes.

Il faudra alors :

1. localiser la ou les racines que l'on veut calculer,
2. s'assurer que la suite (x_k) converge vers une limite r telle que $F(r) = 0$,
3. prévoir un test d'arrêt des calculs, et estimer la précision qu'il garantit,
4. éventuellement, savoir comparer différentes méthodes.

3.1 Heron d'Alexandrie et les racines carrées

Egalement appelé Heron l'ancien, il vécut au premier siècle de notre ère et fut un des grands mécaniciens de l'antiquité ; il inventa diverses machines et instruments de mesure comme l'odomètre permettant de mesurer les distances parcourues en comptant le nombres de tours d'une roue, et dont le principe est encore utilisé par les compteurs kilomètriques des bicyclettes.

L'algorithme de calcul des racines carrées qu'on lui attribue était en fait déja connu des Babyloniens.

3.1.1 La méthode de Héron :

Approche pragmatique :

Soit a un nombre positif dont on veut calculer la racine carrée $\alpha = \sqrt{a}$; α est la longueur du côté d'un carré dont l'aire est a. On va chercher à construire un tel carré en partant d'un rectangle, qu'on va peu à peu transformer pour le faire ressembler à un carré. Si l'on considère une valeur x_0 donnée comme une estimation grossière de α, on construit d'abord un rectangle $\mathcal{R}$ d'aire a ayant un côté de longueur $L = x_0$ et un second côté égal à $l = \dfrac{a}{L} = \dfrac{a}{x_0}$.

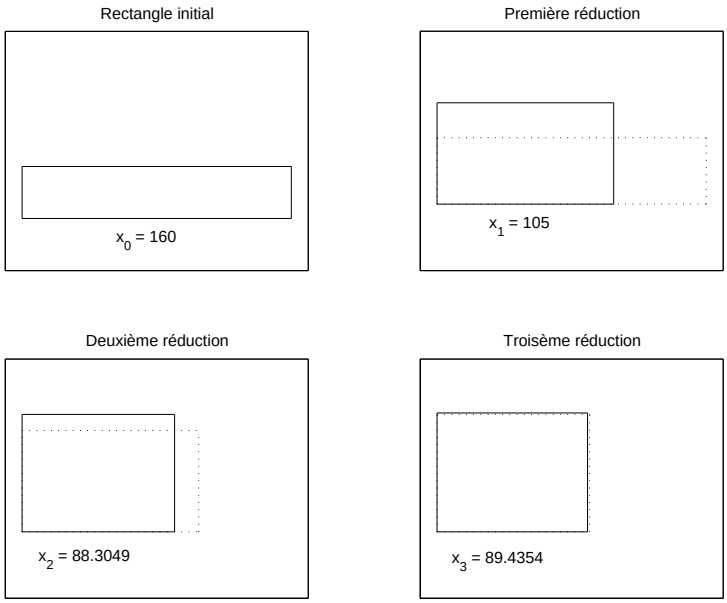


FIGURE 3.1 – Premiers itérés de la méthode de Héron.

Pour construire un nouveau rectangle qui ressemble plus à un carré, il est raisonnable de choisir pour longueur d'un côté la moyenne des longueurs des côtés du rectangle précédent. Cette longueur sera $L = x_1 = \dfrac{1}{2}\,(x_0 + \dfrac{a}{x_0})$. Le second côté aura

pour longueur $l = \dfrac{a}{L} = \dfrac{a}{x_1}$. Et l'on va continuer jusqu'à ce que notre appareil de mesure ne nous permette plus de distinguer les longueurs des deux côtés. La figure 3.1 nous montre trois étapes de ce procédé.

Sur cet exemple, nous cherchons à calculer $\alpha = \sqrt{8000}$. La valeur initiale est $x_0 = 160$. Le rectangle initial a pour côtés 160 et 50.

Pour construire un rectangle qui ressemble plus à un carré, on le remplace par le rectangle dont un côté est la moyenne des deux précédents, c'est-à-dire $x_1 = \dfrac{1}{2}\,(x_0 + \dfrac{a}{x_0}) = 105$ de sorte que l'autre côté a pour longueur $\dfrac{8000}{105} = 76.1905$.

On va ensuite réitérer cette démarche en appliquant la formule

$$x_{k+1} = \frac{1}{2}\,(x_k + \frac{a}{x_k}) \tag{3.1}$$

x_k	$\dfrac{a}{x_k}$
160.00000000000000	50.00000000000000
105.00000000000000	76.19047619047619
90.59523809523810	88.30486202365309
89.45005005944560	89.43538874135297
89.44271940039928	89.44271879958389
89.44271909999159	89.44271909999158

On a obtenu la racine cherchée "à la précision machine". Les chiffres corrects sont soulignés, et l'on constate que dès que l'on a obtenu 3 chiffres corrects (pour x_3), la convergence devient très rapide : 8 chiffres pour x_4 et 15 pour x_5 !

Le principe des approximations successives

On peut regarder cet algorithme d'un point de vue légèrement différent. Partant de l'équation $F(x) = x^2 - a = 0$, qu'on écrit de façon équivalente $x^2 = a$ ou $x = \dfrac{a}{x}$, on obtient successivement :

$$\begin{array}{lcl} x^2 & = & a \\ x^2 + a & = & 2\,a \\ \dfrac{1}{2}\,(x^2 + a) & = & a \\ \dfrac{1}{2}\,(x + \dfrac{a}{x}) & = & \dfrac{a}{x} \quad = \quad x \end{array}$$

On a remplacé l'équation $F(x) = 0$ par une équation équivallente $f(x) = x$, avec ici $f(x) = \dfrac{1}{2}\left(x + \dfrac{a}{x}\right)$, qui permet de retrouver la définition (3.1) de la méthode de

Heron :

$$\begin{cases} x_0 \in \mathbb{R}, & \text{point de départ des itérations} \\ x_{k+1} = f(x_k), & k \geq 0. \end{cases}$$

Ces deux relations définissent les itérations de la méthode. C'est un exemple de méthode de point fixe : la limite $\alpha = \sqrt{a}$ de la suite ainsi définie par récurrence est telle que $\alpha = f(\alpha)$.

Ces deux relations restent imprécises sur deux points fondamentaux :

- le choix de x_0,
- le critère à vérifier pour arrêter les itérations : il n'est pas question de continuer indéfiniment les calculs.

On va voir que pour notre algorithme, ces deux points peuvent être élucidés facilement.

Choix d'une valeur initiale

Notre choix de x_0 était assez empirique, et pour tout dire pas très heureux : le premier rectangle est un peu trop allongé. On peut essayer de rechercher une meilleure valeur pour x_0

Pour cela, on va écrire a sous la forme $a = m \times 4^p$, avec $\dfrac{1}{4} \leq m \leq 1$. Attention, m n'est pas tout à fait une mantisse car il est écrit en base 10. On aura alors $\alpha = \sqrt{m} \times 2^p$, et $0.5 \leq \sqrt{m} \leq 1$.

Il nous suffit maintenant de savoir initialiser les itérations pour rechercher $\sqrt{m}$. La figure 3.2 montre que le courbes représentant $s(x) = \sqrt{x}$ et la droite d'équation $y = \dfrac{1+2x}{3}$ qui passe par les points $\left(\dfrac{1}{4},\ \dfrac{1}{2}\right)$ et $(1,\ 1)$ sont assez proches.

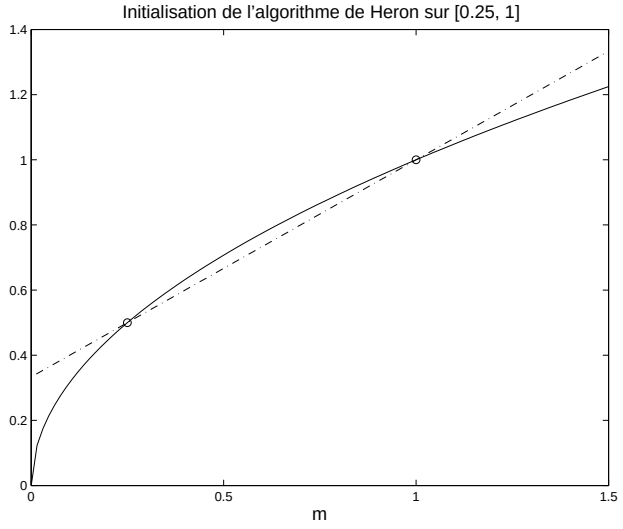


FIGURE 3.2 – Initialisation du calcul de la racine carrée.

On va donc choisir d'initialiser la recherche de $\sqrt{m}$ en utilisant la valeur $x_0 = \frac{1+2m}{3}$. Auparavant, il vaut trouver la valeur de p.

On remarque, mais l'ordinateur le fera très facilement pour nous, que

$$\frac{1}{4} \leq \frac{8000}{4^7} = 0.48828125 \leq 1,$$

de sorte que l'on choisira cette valeur pour m, avec $p = 7$. On obtient alors :

x_k	$\frac{m}{x_k}$
0.65885416666667	0.74110671936759
0.69998044301713	0.69756413178540
0.69877228740126	0.69877019853767
0.69877124296946	0.69877124296790
0.69877124296868	0.69877124296868

On retrouve $\sqrt{a} \simeq 0.69877124296868 \times 2^7 = 89.44271909999159$.

Arrêt des itérations

Supposons calculé un itéré x_k de notre suite, et évaluons :

$$x_{k+1} - \sqrt{m} = \frac{1}{2}\,(x_k + \frac{m}{x_k}) - \sqrt{m} = \frac{1}{2}\,\frac{x_k^2 - 2\,x_k\,\sqrt{m} + (\sqrt{m})^2}{(\sqrt{x_k})^2} = \frac{1}{2}\,\left(\frac{x_k - \sqrt{m}}{\sqrt{x_k}}\right)^2$$

de sorte que si $e_k = x_k - \sqrt{m}$ désigne l'erreur après k itérations, on aura :

$$e_{k+1} = \frac{1}{2\,x_k}\,e_k^2$$

Mais on peut montrer que tous les x_k sont dans l'intervalle [0.5, 1], de sorte que $2\,x_k \geq 1$ et :

$$e_{k+1} \leq e_k^2.$$

Par ailleurs, la fonction $e(x) = \sqrt{x} - \frac{1+2x}{3}$ a sa dérivée $e'(x) = \frac{1}{2\sqrt{x}} - \frac{2}{3}$ qui s'annule pour $x = \frac{9}{16}$. C'est donc pour $m = \frac{9}{16}$ que l'erreur e_0 sera maximale. Ce maximum de l'erreur est d'environ $0.041666 < 0.05$

On en déduit que $e_4 \leq e_3^2 \leq e_2^4 \leq e_1^8 \leq (0.05)^{16} \ < \ 2 \times 10^{-21}$, de sorte que 4 itérations suffisent à atteindre une précision meilleure que l'unité d'arrondi u (environ 10^{-16} en double précision).

3.2 Résolution d'une équation non linéaire

On recherche les racines positives de l'équation $F(x) = 0$, la fonction F étant définie par

$$F(x) = 0.01 \exp x + 10 \cos x - 3\,x. \tag{3.2}$$

3.2.1 Localisation des racines

La recherche d'une solution de cette équation ne peut se faire que par une méthode itérative. On peut utiliser deux types de méthodes. Certaines travaillent sur des intervalles contenant la racine cherchée, d'autres avec une seule valeur. La méthode devra être alimentée par une ou deux valeurs initiales qui seront si possible assez proche de la racine cherchée.

Représentation graphique

Une première façon de choisir une valeur initiale consiste à faire une représentation graphique. Voici une représentation graphique de F sur l'intervalle $[-1,\ 10]$.

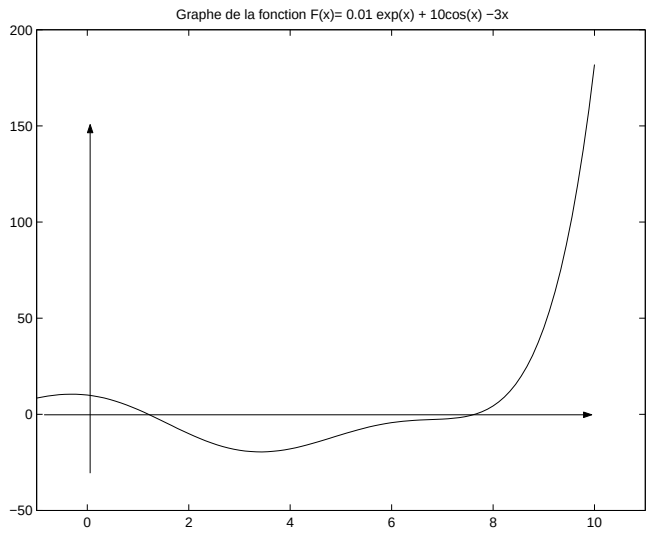


FIGURE 3.3 – Un graphique pour localiser les racines.

Notre équation possède apparemment deux racines positives :
- la première est dans l'intervalle $[0,\ 2]$, plutôt vers le milieu,
- la seconde est dans l'intervalle $[7,\ 8]$, plus près de 8.

Le tracé d'une telle figure est particulièrement facile avec le logiciel MATLAB . Voici les commandes qui permettent d'obtenir la figure ci-dessus :

```
x = linspace(-1,10);
plot(x, 0.01*exp(x)+10*cos(x)-3*x);
hold on ;
```

```
plot([-1,10],[0,0],10,0,'>');
plot([0,0],[-30,150],0,150,'^');
title('Graphe de la fonction F(x) = 0.01exp(x)+10 cos(x)-3x');
axis([-1 11 -50 200]);
```

Balayage systématique

Faute de possibilité de tracer facilement un graphe, on peut balayer systématiquement l'intervalle qui nous intèresse pour rechercher d'éventuels changements de signe de F.

Pour notre exemple, on peut calculer les valeurs de $F(x)$ pour les valeurs entières positives de x. Ces valeurs sont, dans l'ordre 10.01, 2.43, -10.08, -18.69, -17.99, -10.67, -4.36, -2.49, 4.35, 44.91, 181.87.

On observe un changement de signe entre 1 et 2 ainsi qu'entre 7 et 8.

Arguments physiques

Une connaissance *a priori* de la nature du problème vous permettra de situer grossièrement les racines qui vous intéressent. C'est d'ailleurs la première reflexion qu'il faut faire ; elle permet de délimiter l'intervalle d'étude de la fonction, avant même la représentation graphique.

3.2.2 Méthode de dichotomie

Principe

Si la fonction continue F change de signe sur un intervalle $[a,\ b]$, c'est-à-dire si $F(a)\,F(b) < 0$, alors F s'annule au moins une fois sur cet intervalle. Cette remarque peut être utilisée pour former une suite d'intervalles de plus en plus petits qui contiennent une racine.

En notant a_0, b_0 les bornes de cet intervalle et $m = \dfrac{a_0 + b_0}{2}$ le milieu de cet intervalle, on aura deux possibilités :

- soit $F(a_0)\,F(m) \leq 0$, et dans ce cas nous aurons une racine dans $[a_0,\ m]$; on pose $a_1 = a_0$ et $b_1 = m$;
- soit $F(b_0)\,F(m) \leq 0$, et dans ce cas nous aurons une racine dans $[m,\ b_0]$; on pose $a_1 = m$ et $b_1 = b_0$.

Dans les deux cas, le nouvel intervalle $[a_1,\ b_1]$ est de longueur $b_1 - a_1 = \dfrac{b_0 - a_0}{2}$. On recommence le même processus avec ce nouvel intervalle.

On construit ainsi une suite d'intervalles $[a_k,\ b_k]$ qui sont emboîtés et contiennent une racine. On arrêtera ces itérations lorsque la longueur $|a_k - b_k|$ de l'intervalle sera inférieure $\tau = 2\,\varepsilon$, qui correspond à un seuil d'erreur absolue ε choisi à l'avance. On prendra pour racine la valeur $\tilde{r} = \dfrac{a_k + b_k}{2}$.

Ceci correspond à l'algorithme suivant :

a ; b ; $fa \leftarrow F(a)$; $fb \leftarrow F(b)$;
erreur si $fa * fb > 0$;
Tant que $|b - a| > \tau$
  $c \leftarrow (a+b)/2$; $fc \leftarrow F(c)$;
  Si $fa * fc < 0$
    $b \leftarrow c$; $fb \leftarrow fc$;
  Sinon
    $a \leftarrow c$; $fa \leftarrow fc$;

Mise en oeuvre

Voici les résultats obtenus pour notre fonction F en partant des intervalles $[1,\ 2]$ et $[7,\ 8]$.

k	a_k	b_k	a_k	b_k
0	1.000000000	2.000000000	7.000000000	8.000000000
1	1.000000000	1.500000000	7.500000000	8.000000000
2	1.000000000	1.250000000	7.500000000	7.750000000
3	1.125000000	1.250000000	7.625000000	7.750000000
4	1.187500000	1.250000000	7.625000000	7.687500000
5	1.187500000	1.218750000	7.625000000	7.656250000
6	1.203125000	1.218750000	7.625000000	7.640625000
7	1.203125000	1.210937500	7.632812500	7.640625000
8	1.203125000	1.207031250	7.636718750	7.640625000
9	1.203125000	1.205078125	7.638671875	7.640625000
10	1.204101562	1.205078125	7.639648437	7.640625000
11	1.204589843	1.205078125	7.639648437	7.640136718
12	1.204589843	1.204833984	7.639648437	7.639892578
13	1.204589843	1.204711914	7.639770507	7.639892578
14	1.204589843	1.204650878	7.639831542	7.639892578

Les racines calculées sont alors respectivement $\tilde{r_1} = 1.204620361$ et $\tilde{r_2} = 7.639862060$.

Convergence et arrêt des itérations

Sur les deux racines calculées précédement, la précision garantie est la même : en effet la longueur des derniers intervalles calculés est, dans un cas 6.1035×10^{-5} et dans l'autre 6.1036×10^{-5}.

Cette précision peut être prévue à l'avance. A l'initialisation, l'intervalle $[a_0,\ b_0]$ a pour longueur 1 ; à la première itération, sa longueur sera $\frac{1}{2}$, et à la k-ième, elle sera $\frac{1}{2^k}$, et comme on choisit le milieu de l'intervalle, l'erreur maximale sera $\frac{1}{2^{k+1}}$.

Si l'on souhaite une précision de 10^{-4}, on cherchera k tel que $\frac{1}{2^{k+1}} \leq 10^{-4}$, soit encore $(k+1)\ln 2 \geq 4\ln 10$, c'est à dire $k \geq 4\,\frac{\ln 10}{\ln 2} - 1 \simeq 12.2877$. On peut que pour $k = 13$, la précision garantie était 6.1242×10^{-5}.

Ces remarques se généralisent, et peuvent s'énoncer de la façon suivante :

Théorème 3.1 *Si r désigne une racine de l'équation $F(x) = 0$ contenue dans l'intervalle $[a_0,\ b_0]$. La méthode de dichotomie appliquée à la recherche de cette racine est telle que :*

i) la racine approchée r_k calculée à l'itération k vérifie $|r_k - r| \leq \frac{|a_0 - b_0|}{2^{k+1}}$.

ii) le nombre k_ϵ d'itérations nécessaire pour atteindre une précision ϵ est tel que $(k_\epsilon + 1)\ln 2 \geq \ln \frac{|a_0 - b_0|}{\epsilon}$.

La borne de l'erreur est divisée par 2 à chaque itération.

Définition 3.1 : Ordre d'une méthode.
Si la suite des itérés $(x_k)_{k\geq 0}$ d'une méthode itérative est telle qu'il existe une constante $c > 0$ telle que pour k suffisament grand, les erreurs $e_k = |x_k - x|$ vérifient $e_{k+1} \leq Ce_k^p$, on dit que la méthode est d'ordre p.

La méthode de dichotomie est donc d'ordre 1.

3.2.3 Méthode de point fixe

Principe

Lors de la recherche de la racine carrée par la méthode de Héron, nous avons construit une suite définie par $x_{n+1} = f(x_n)$ et un x_0 donné. Nous avons alors montré que cette suite converge vers une valeur qui n'est autre que $\sqrt{a}$.

De façon générale, on peut chercher à suivre un principe d'approximations successives en construisant des suites de la forme $x_{n+1} = f(x_n)$ pour $n \in \mathbb{N}$, avec x_0 donné et nous nous intéresserons à la convergence de la suite $(x_n)_{n\in\mathbb{N}}$ ainsi définie. Un résultat de convergence est donné par le théorème du point fixe qui nous fournit des conditions suffisantes pour la convergence de la suite (x_n).

Théorème 3.2 *Soit $I = \mathbb{R}$ ou $I = [a, b]$ avec $a < b$ deux réels.*
Soit f une application de I dans $\mathbb{R}$ telle que
- *$f(I) \subset I$,*
- *il existe une constante $k \in [0, 1[$ telle que f est k-contractante, c'est-à-dire*

$$\forall (x, y) \in I^2,\ |f(x) - f(y)| \leq k|x - y|.$$

Alors la fonction f admet un unique point fixe $l \in I$ tel que $f(l) = l$. De plus, toute suite récurrente définie par

$$x_0 \in I \quad et \; \forall n \in \mathbb{N} \; x_{n+1} = f(x_n)$$

converge vers l.

Prenons un $x_0 \in I$. On veut prouver que la suite (x_n) converge dans I. On montre pour cela que la suite est de Cauchy.
Comme $f(I) \subset I$, par une récurrence immédiate, on a $\forall n \in \mathbb{N} \; x_n \in I$.
En prenant $x = x_{n+1}$ et $y = x_n$ dans l'inégalité, on obtient alors :

$$|x_{n+2} - x_{n+1}| = |f(x_{n+1}) - f(x_n)| \leq k|x_{n+1} - x_n|,$$

et par récurrence que $\forall n \in \mathbb{N}$

$$|x_{n+1} - x_n| \leq k^n |x_1 - x_0|.$$

On obtient ainsi $\forall (n, p) \in \mathbb{N} \times \mathbb{N}^*$

$$|x_{n+p} - x_n| \leq \sum_{i=0}^{p-1} |xn + i + 1 - x_{n+i}| \leq \sum_{i=0}^{p-1} k^{n+i} |x_1 - x_0| = k^n |x_1 - x_0| \frac{1 - k^p}{1 - k}.$$

Comme $0 \leq k < 1$, $0 < \dfrac{1 - k^p}{1 - k} < 1$ et $\forall (n, p) \in \mathbb{N} \times \mathbb{N}^*$

$$|x_{n+p} - x_n| \leq \frac{k^n}{1 - k} |x_1 - x_0|.$$

En conclusion, $\forall \varepsilon > 0$, soit N tel que $\dfrac{k^N}{1 - k} |x_1 - x_0| \leq \varepsilon$, alors pour tout $n \in \mathbb{N}$ et pour tout $p \in \mathbb{N}^*$, si $n \geq N$ alors $|x_{n+p} - x_n| \leq \varepsilon$. La suite (x_n) est de Cauchy, donc convergente. Notons l sa limite.

Comme la fonction f est continue sur I et que $x_{n+1} = f(x_n)$, on passe à la limite quand $n \to +\infty$ et on a $l = f(l)$. La limite de la suite x_n est donc un point fixe de f.
Montrons maintenant que f admet un unique point fixe sur I : supposons que f admet au moins deux points fixes l_1 et l_2 dans I. Comme f est k-contractante, $|l_1 - l_2| = |f(l_1) - f(l_2)| \leq k|l_1 - l_2|$. Comme $k \in [0, 1[$, on en déduit $l_1 = l_2$. Donc f admet un unique point fixe $l \in I$.

Application

Soit $I = [a, b]$ et $F : I \to \mathbb{R}$ une application de classe $\mathcal{C}^1$ pour laquelle on a localisé une **unique** racine $l \in I$. En fait, on supposera F monotone sur I, avec $F(a)\, F(b) < 0$.

– si F est strictement décroissante sur I, alors soit $M > 0$ tel que $\forall x \in I$, $-M \leq F'(x) < 0$. Définissons $f : I \to I$ par $f(x) = x + \dfrac{1}{M+1}F(x)$. f vérifie toutes les hypothèses du théorème de point fixe, donc toute suite récurrente sur I, $x_{n+1} = f(x_n)$, converge vers l'unique point fixe de f qui est l'unique racine de F.
– si F est strictement croissante sur I, alors soit $M > 0$ tel que $\forall x \in I$, $0 < F'(x) < M$. Définissons $f : I \to I$ par $f(x) = x - \dfrac{1}{M+1}F(x)$. f vérifie de même toutes les hypothèses du théorème de point fixe, donc toute suite récurrente sur I, $x_{n+1} = f(x_n)$, converge vers l'unique point fixe de f qui est l'unique racine de F.

Dans les deux situations évoquées par l'application ci-dessus, on montre que l'application f est contractante en appliquant le théorème des accroissements finis. La constante k du théorème 3.2 sera comprise entre $1 - \dfrac{M}{M+1}$ et 1 : elle peut-être assez proche de 1, et dans ce cas la convergence sera lente. Suivant que f sera croissante ou décroissante, la figure 3.4 montre les premières itérations d'une méthode de point fixe définie par f.

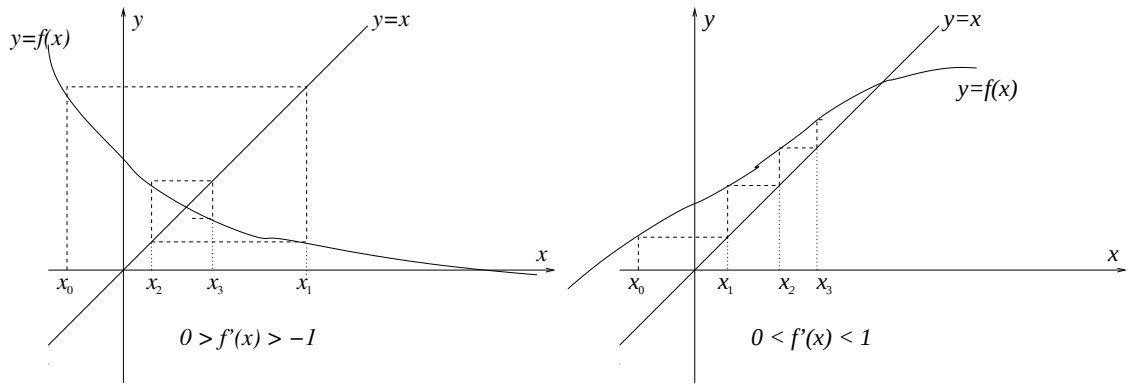


FIGURE 3.4 – Premiers itérés d'une méthode de point fixe.

Mise en œuvre

Pour la fonction F de (3.2), on a observé qu'on avait une racine dans l'intervalle [1, 2] et une racine dans l'intervalle [7, 8]. Une observation graphique de la dérivée de F sur ces intervalles permet de choisir, pour le premier $M = 13$ et pour le second, $M = 20$. Cependant, les performances de la méthode ne sont pas les mêmes dans les deux cas comme l'illustrent les itérations présentées dans le tableau 3.1, respectivemment initialisées à 8 et 2.

Exemple 3.1 *Lorsqu'elle converge, une méthode de point fixe est d'ordre 1.*

k	x_k	x_k
1	7.782271023383442	1.224032476578299
2	7.714930140950544	1.205607243906417
3	7.682084600165323	1.204670726723357
4	7.664303002069792	1.204620696293069
5	7.654223288962419	1.204618016850043
6	7.648372547353040	1.204617873329813
7	7.644931867641581	1.204617865642318
8	7.642893337517672	1.204617865230546
9	7.641680290776899	1.204617865208490
10	7.640956604817492	1.204617865207309
11	7.640524208045744	1.204617865207245
12	7.640265620540383	1.204617865207242
13	7.640110893202825	1.204617865207242
14	7.640018281330524	1.204617865207242

TABLE 3.1 – Deux comportements différents de la méthode du point fixe.

3.2.4 La méthode de Newton

Principe

Supposons maintenant que comme précédement, une représentation graphique nous ait indiqué que la valeur $x_0 = 8$ est proche d'une racine. Un développement de Taylor à l'ordre 1 en h au point x_0 s'écrit :

$$F(x_0 + h) = F(x_0) + h\,F'(x_0) + h\,\varepsilon(h) \tag{3.3}$$

La racine cherchée peut s'écrire $r = x_0 + h$ pour un h inconnu qui sera tel que :

$$F(x_0) + h\,F'(x_0) + h\,\varepsilon(h) = 0 \tag{3.4}$$

Comme h est assez petit, $\varepsilon(h)$ sera encore plus petit ; on va négliger le terme $h\,\varepsilon(h)$ et forcer l'égalité à 0 en posant

$$F(x_0) + h_1\,F'(x_0) = 0 \tag{3.5}$$

qui fournit $h_1 = -\dfrac{F(x_0)}{F'(x_0)}$.

On aura ainsi défini le premier itéré d'une nouvelle méthode en posant

$$x_1 = x_0 - \frac{F(x_0)}{F'(x_0)}.$$

Par ailleurs, on sait que la tangente à la courbe représentant $y = F(x)$ au point x_0 a pour équation

$$y = F(x_0) + (x - x_0)\, F'(x_0) \tag{3.6}$$

On remarque alors que cette tangente coupe l'axe des x au point x_1 : le choix de x_1 est donc la solution d'un modèle linéaire de notre problème au voisinage de x_0 !

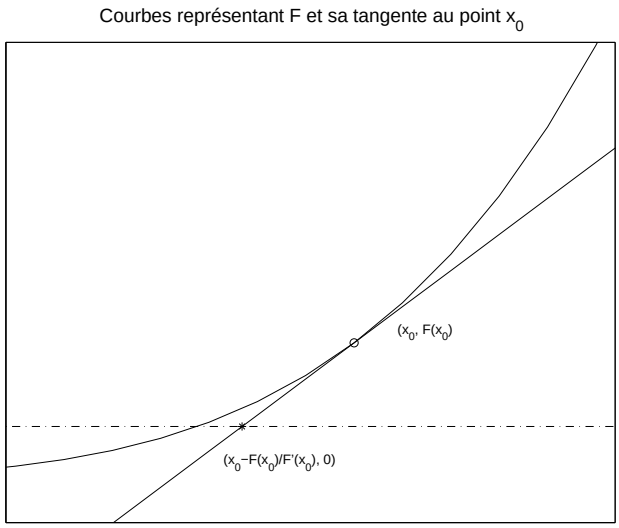


FIGURE 3.5 – La méthode de Newton choisit l'itéré suivant sur la tangente.

Mise en oeuvre

La méthode de Newton va être définie en itérant cette démarche. L'algorithme consiste donc, partant de x_0 voisin d'une racine, à calculer pour $k \geq 0$:

$$x_{k+1} = x_k - \frac{F(x_k)}{F'(x_k)} \tag{3.7}$$

L'algorithme suivant permet, à partir d'une valeur approchée x_0 de calculer une racine, à condition que la méthode converge vers cette racine. C'st une version très édulcorée, et un algorithme plus complet est donné pour les systèmes en section **3.3**.

Il n'utilise que deux variables pour calculer les itérés successifs : on les note x_{avant} et x_{apres}.

On teste la convergence à une précision relative ε sur 2 itérés successifs.

$x_{apres} \longleftarrow x_0$;
$x_{avant} \longleftarrow x_{apres} + 1$;
Tant que $|x_{apres} - x_{avant}| > 2 * \varepsilon * |x_{avant}|$
$\quad x_{avant} \longleftarrow x_{apres}$;
$\quad x_{apres} \longleftarrow x_{avant} - F(x_{avant})/F'(x_{avant})$;

Nous l'avons appliqué à la recherche des zéros positifs de la fonction F, partant respectivement des valeurs $x_0 = 2$ et $x_0 = 8$; les valeurs calculées sont présentées au tableau 3.2.

k	x_k	x_k
0	2.0000000000000000	8.0000000000000000
1	1.1607032574053444	7.7425762473069293
2	1.2049212788797627	7.6507460430283869
3	1.2046178784694039	7.6400156469715865
4	1.2046178652072419	7.6398801183259391
5	1.2046178652072419	7.6398800969514733
6	1.2046178652072419	7.6398800969514733

TABLE 3.2 – Deux comportements différents de la méthode du point fixe.

La suite converge à la précision de l'arithmétique en 4 ou 5 itérations. La méthode de Newton est beaucoup plus rapide que la précédente. On peut aussi le voir en regardant les courbes représentant l'évolution de l'erreur pour chaque méthode dans la recherche de la racine, sur la figure 3.6.

On y remarque que l'erreur pour chacune de ces méthodes est comparable à une courbe :

- pour la méthode de Newton, c'est la courbe représentant $e_n(k) = (\frac{1}{2})^{2^k}$,
- pour la méthode de dichotomie, c'est la courbe représentant $e_d(k) = (\frac{1}{2})^k$.

Pour la méthode de dichotomie, on retrouve le principe de la convergence linéaire : à chaque itération, l'erreur est divisée par 2. Pour la méthode de Newton, c'est le principe observé pour la méthode de Heron que l'on retrouve : en effet, ici $|x_0 - r| \simeq \frac{1}{2}$, et au-delà, pour $k \geq 0$, la relation $|x_{k+1} - r| \simeq |x_k - r|^2$ entraine donc

$$|x_{k+1} - r| \simeq \left(|x_{k-1} - r|^2\right)^2,$$

qui permet, en remontant jusqu'à $k = 0$, de retrouver

$$|x_k - r| \simeq (\frac{1}{2})^{2^k}.$$

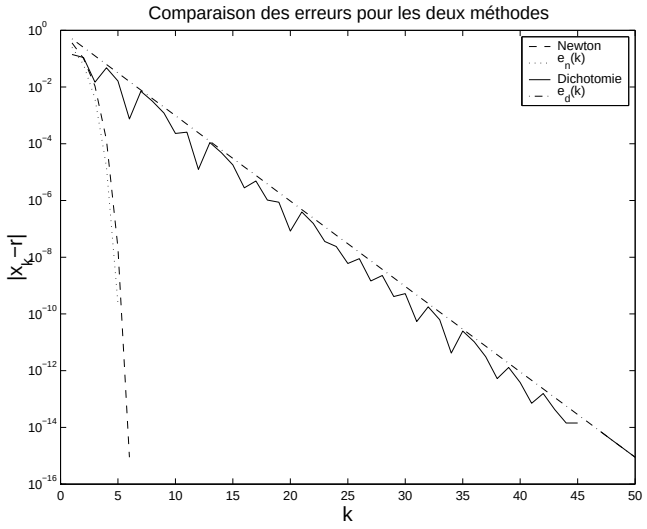


FIGURE 3.6 – Convergence comparée des méthodes de Newton et de dichotomie.

On peut montrer, que sous des hypothèses raisonnables, la méthode de Newton est d'ordre 2. C'est l'objet du théorème suivant.

Théorème 3.3 *On suppose que*
- *la fonction f est de classe $\mathcal{C}^2$ au voisinage d'une racine r,*
- *$|f'(x)| > 0$ dans un voisinage de r,*
- *les itérés $(x_k)_{k\geq 0}$ de la méthode de Newton convergent vers r.*

Alors, pour k suffisament grand, $|x_{k+1} - r| = \mathcal{O}(|x_k - r|^2)$.

La formule de Taylor-Young permet d'écrire

$$0 = F(r) = F(x_k) + (r - x_k)\, F'(x_k) + \frac{1}{2}(r - x_k)^2\, F"(x_k) + o\left((r - x_k)^2\right),$$

de sorte que

$$(r - x_k)\, f'(x_k) - f(x_k) = \frac{1}{2}(r - x_k)^2\, F"(x_k) + o\left((r - x_k)^2\right). \tag{3.8}$$

Par la d'éfinition (3.7) de la méthode $x_{k+1} = x_k - \dfrac{F(x_k)}{F'(x_k)}$, de sorte que

$$x_{k+1} - r = (x_k - r) - \frac{F(x_k)}{F'(x_k)} = \frac{(x_k - r\, F'(x_k)) - F(x_k}{)} F'(x_k).$$

de sorte qu'avec (3.8),

$$x_{k+1} - r = \frac{(r - x_k)^2\, F"(x_k) + o\left((r - x_k)^2\right)}{2\, F'(x_k)} = \mathcal{O}(|x_k - r|^2). \tag{3.9}$$

Remarque 3.1 *Il ne faut pas s'étonner de retrouver ici une propriété de la méthode de Heron. En effet, si l'on veut appliquer la méthode de Newton à la fonction* $F(x) = x^2 - 2$*, dont la dérivée est* $F'(x) = 2\,x$*, les itérations s'écrivent*

$$x_{k+1} = x_k - \frac{x_k^2 - 2}{2\,x_k} = \frac{x_k}{2} + \frac{1}{x_k} = \frac{1}{2}\,\left(x_k + \frac{2}{x_k}\right).$$

La méthode de Heron est une application particulière de la méthode de Newton.

Le comportement observé ci-dessus peut laisser penser que la méthode de Newton est préférable à la méthode de dichotomie. Cependant, pour l'exemple traité, le choix de x_0 a permis la convergence, et nous avons cependant précisé, en présentant l'algorithme, qu'il devait être assez voisin de la racine cherchée.

En fait, les choses sont un peu plus compliquées : on peut atteindre certaines racines en partant d'assez loin, et et les manquer alors que l'on part d'assez près, soit pour converger vers une autre racine, soit pour diverger !

Ainsi, en initialisant les itération à $x_0 = 3$, la méthode va converger vers une racine indésirable, approximativemùent égale à -2.35.

Considérons à présent la fonction

$$F(x) = 3\,\cos(x) - \log|2\,x|\,. \tag{3.10}$$

La figure (3.7) montre une courbe représentant cette fonction ainsi les points $(x_k,\ F(x_k))$ pour les itérés calculés avec deux choix différents de x_0.

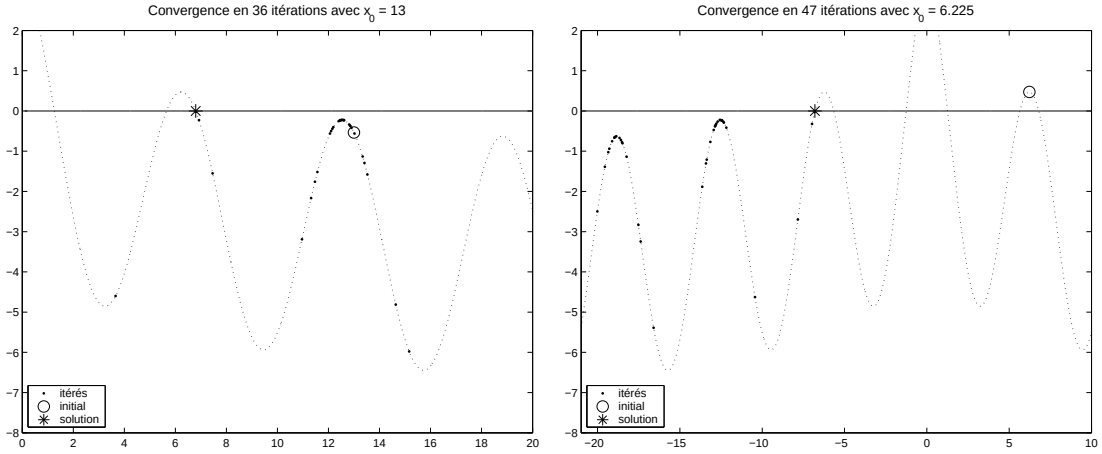


FIGURE 3.7 – On ne converge pas toujours vers la racine attendue !

En partant de $x_0 = 13$, la méthode hésite un certain temps autour de $x \simeq 12.53$ en restant dans l'intervalle $[0,\ 15]$ jusqu'à l'itération $k = 30$, avant de partir converger vers la solution approchée $x \simeq 6.7988$ en 37 itérations.

En partant de $x_0 = 6.225$, la méthode commence par aller visiter l'intervalle $[-20.5, -16.5]$, hésitant jusqu'à l'itération 19 autour d'un "pic" de la fonction $F(x)$ approximativement situé au point -18.8 ; les 20 itérations suivantes vont osciller autour de la valeur -12.5 dans l'intervalle $[-11.5, -13.5]$, et finalement la convergence aura lieu vers $x \simeq -6.7988$ en 47 itérations.

Dans les deux cas rapportés ci-dessus, la méthode a convergé, mais cela n'est pas garanti. Lorsque l'on part de $x_0 = 14$, la méthode hésite jusqu'à l'itération 18, en restant d'abord autour du pic situé aux environs de $x = 12.5$, puis autour de celui que l'on trouve aux environs de $x = 18.8$, avec $x_{18} \simeq 18.8307$, avant de s'échapper jusqu'à $x_{19} \simeq 198.744$. Au-dela, si l'on suit l'algorithme présenté ci-dessus, les calculs ne s'interrompent pas, mais les itérations vont faire du "ping-pong" sur l'intervalle $[198,\ 204]$, autour du pic qu'il contient. La figure 3.8 montre les points $(x_k,\ F(x_k))$ pour $k \leq 5000$! En fait, le point $x_{18} \simeq 18.8307$ est très proche d'un pic de $F(x)$, et $F'(x_{18} \simeq 0.0035$: la tangente à la courbe $y = F(x)$ en ce point est presque horizontale ; c'est ce qui explique que le point x_{19} est aussi éloigné de x_{18}.

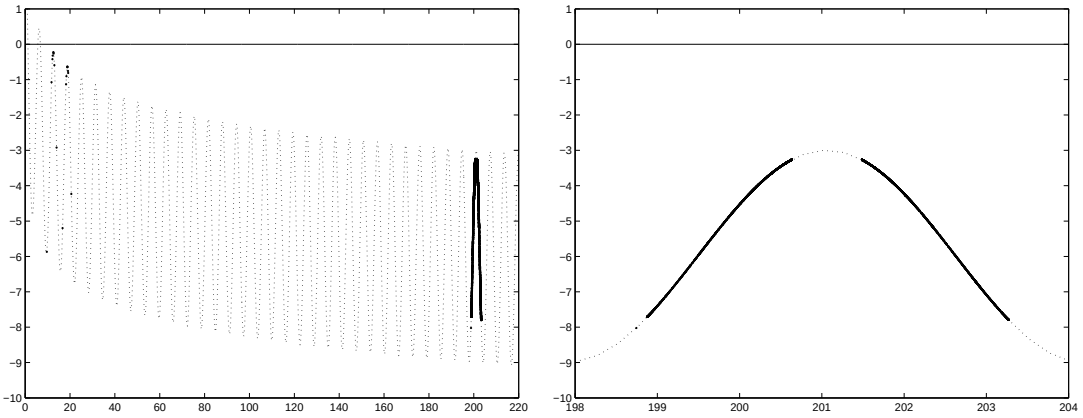


FIGURE 3.8 – A partir de la 19ème itération, les itérés restent autour d'un pic !

L'algorihme présenté ci-dessus n'est donc pas suffisant ; il faudra limiter le nombre d'itérations, et vérifier la convergence en sortie de boucle.

3.2.5 Combiner deux méthodes

La méthode de Newton est incontestablement la meilleure lorsqu'on est assez proche de la racine recherchée. Par contre elle n'est pas très robuste, et sa convergence vers la racine cherchée dépend de la fonction et de la solution.

La méthode de dichotomie est beaucoup plus lente, mais elle est plus robuste : lorsque l'on a isolé une racine avec changement de signe, on ne risque plus rien.

On peut combiner les deux méthodes pour tirer profit des avantages de chacune. A l'initialisation, on dispose d'un intervalle $[a_0,\ b_0]$ tel que $F(a_0)\,F(b_0) < 0$. On choisit une des bornes a_0 ou b_0 pour x_0.

L'algorithme qui suit indique le principe d'une implantation de cette méthode, avec le choix de a_0 pour x_0.

$a \quad ; \quad b \quad ; \quad fa \leftarrow F(a) \quad ; \quad fb \leftarrow F(b) \quad ;$
`erreur si` $fa * fb > 0$
$x \leftarrow a \quad ; \quad fx \leftarrow fa \quad ; \quad fpx \leftarrow F'(x) \quad ;$
`Tant que` $|b-a| > 2 * u * \max\{|a|, \ |b|\}$
$\quad x \leftarrow x - fx/fpx \quad ;$
$\quad$ `Si` $x < a$ `ou` $x > b$
$\quad\quad x \leftarrow (a+b)/2 \quad ;$
$\quad fx \leftarrow F(x) \quad ; \quad fpx \leftarrow F'(x) \quad ;$
$\quad$ `Si` $fa * fx < 0$
$\quad\quad b \longleftarrow x \quad ; \quad fb \longleftarrow fx$
$\quad$ `Sinon`
$\quad\quad a \leftarrow x \quad ; \quad fa \leftarrow fx$

3.2.6 Quand on veut éviter le calcul des dérivées

Un autre inconvénient de la méthode de Newton est que parfois les dérivées de F sont difficiles à calculer.

On va alors se rappeler que l'on peut approcher une dérivée par une formule de différences finies.

Supposons calculés les itérés jusqu'à l'étape k. La méthode de Newton demande de calculer $x_{k+1} = x_k - \dfrac{F(x_k)}{F'(x_k)}$.

L'idée de la méthode de la sécante consiste à remplacer $F'(x_k)$ par

$$d_k = \frac{F(x_k) - F(x_{k-1})}{x_k - x_{k-1}}.$$

Cette méthode remplace la tangente de la figure (3.5) par une sécante à la courbe $y = F(x)$ passant par les points $(x_{k-1},\ F(x_{k-1}))$ et $(x_k,\ F(x_k))$. Elle nécessite deux valeurs pour être initialisée. Partant de x_0, on calculera un d_0 en posant, pour un h convenablement choisi $d_0 = \dfrac{F(x_0+h) - F(x_0)}{h}$.

La courbe ci-dessous compare la méthode de Newton à celle de la sécante, initialisée en choisissant $h = -0.1$ pour $x_0 = 8$ dans le cas de la fonction (3.2).

Sur la figure 3.9, la courbe représentant $e_s(k)$ donne une idée de la convergence de la méthode de la sécante.Les valeurs $e_s(k)$ sont définies par $e_s(k) = (\dfrac{1}{2})^{d^k}$, avec $d = \dfrac{1+\sqrt{5}}{2}$.

Cette convergence n'est pas trop mauvaise par rapport à celle de la méthode de Newton, et un regard sur la figure 3.6 montre qu'elle est bien plus rapide que celle de la méthode de dichotomie.

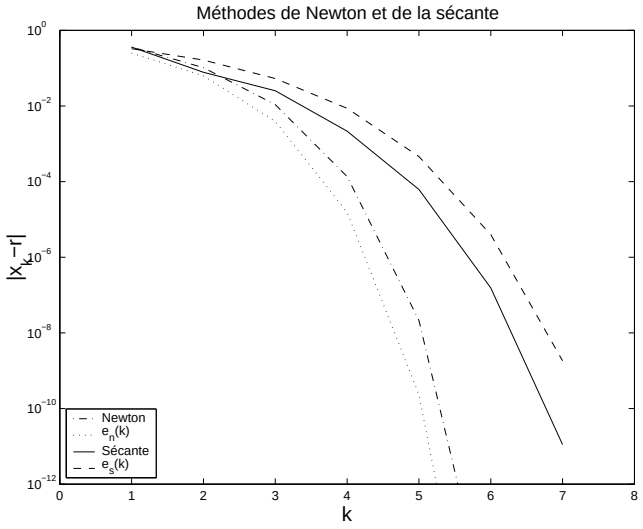


FIGURE 3.9 – Convergence comparée des méthodes de Newton et de la sécante.

3.3 Systèmes d'équations non linéaires

On considère à présent une fonction F définie sur un domaine Ω de $\mathbb{R}^n$ à valeurs dans $\mathbb{R}^n$. On s'intéresse à la recherche d'une ou de plusieurs solutions du système d'équations non linéaires :

$$F(x) = 0 \Longleftrightarrow \begin{bmatrix} F_1(x_1, \ldots, x_n) \\ \ldots \\ F_n(x_1, \ldots, x_n) \end{bmatrix} = \begin{bmatrix} 0 \\ \ldots \\ 0 \end{bmatrix}. \tag{3.11}$$

Les éléments de $\mathbb{R}^n$ seront indifféremment considérés comme des points ou des vecteurs : lorsqu'on écrit un produit $M\,x$, c'est le vecteur x que l'on désigne ; les coordonnées d'une solution de (3.11) peuvent être celles d'un point.

3.3.1 La méthode de Newton

Principe

Supposons que la fonction F est de classe $\mathcal{C}^1$, et notons J_F sa matrice jacobienne :

$$J_F(x) = \begin{bmatrix} \dfrac{\partial F_1}{\partial x_1} & \cdots & \dfrac{\partial F_1}{\partial x_n} \\ \ldots & & \ldots \\ \dfrac{\partial F_n}{\partial x_1} & \cdots & \dfrac{\partial F_n}{\partial x_n} \end{bmatrix}.$$

Pour $x \in \mathbb{R}^n$ et $h \in \mathbb{R}^n$, on peut écrire un développement analogue à (3.3) :

$$F(x+h) = F(x) + J_F(x)\,h + \|h\|\,\varepsilon(h). \tag{3.12}$$

Cette fois, $\varepsilon(h)$ désigne une fonction définie dans un voisinage de $0 \in \mathbb{R}^n$ qui tend vers 0 avec $\|h\|$.

Supposons à présent que x soit voisin d'une solution d'une solution x^* de (3.11). On le note $x^{(0)}$. La solution x^* peut s'écrire $x^* = x^{(0)} + d$ pour un vecteur d inconnu qui sera tel que :

$$F(x^*) = 0 = F(x^{(0)}) + J_F(x^{(0)})\, d + \|d\|\, \varepsilon(d). \tag{3.13}$$

Si d est assez petit, $\|d\|\, \varepsilon(d)$ sera encore plus petit ; on va le négliger et forcer l'égalité à 0 dans (3.13) en remplaçant d par $d^{(0)}$ qui vérifie :

$$F(x^{(0)}) + J_F(x^{(0)})\, d^{(0)} = 0 \tag{3.14}$$

que l'on peut écrire également :

$$d^{(0)} = -\left(J_F(x^{(0)})\right)^{-1} F(x^{(0)}). \tag{3.15}$$

La relation (3.15) montre l'analogie avec la formule (3.7) qui définit les itérations de la méthode de Newton en dimension 1.

Cependant, c'est sous la forme (3.14) que l'on envisagera le calcul de $d^{(0)}$: on calculera la solution du système linéaire de matrice $J_F(x^{(0)})$ et de second membre $-F(x^{(0)})$!

On peut alors poser $x^{(1)} = x^{(0)} + d^{(0)}$: c'est le premier itéré de notre méthode.

Au-delà, elle sera définie en calculant, pour $k \geq 1$:

- le second membre $-F(x^{(k)})$ et la matrice $J_F(x^{(k)})$,
- la solution $d^{(k)}$ du systéme linéaire qu'ils définissent (factorisation LU),
- le nouvel itéré $x^{(k+1)} = x^{(k)} + d^{(k)}$.

Arrêt des itérations

Comme en dimension 1, cette méthode n'est pas assurée de converger. Il faut choisir un $x^{(0)}$ "assez voisin" d'une solution. Ce choix n'est pas aussi facile que pour la dimension 1, car on ne peut pas s'aider de figure ni envisager de balayage systématique. On ne peut pratiquement compter que sur une connaissance *a priori* du problème.

Comme la méthode ne garantit pas la convergence, il faut écrire l'algorithme en conséquence. On se fixera :

- un nombre maximum d'itérations,
- un seuil de convergence, qui peut être :
  - un seuil τ_x pour $\|d^{(k)}\|$: test de convergence de la suite, qui doit être relatif,
  - un seuil τ_F pour $\|F(x^{(k)})\|$: test de réalisation de l'objectif,
  - une combinaison des précédents.

On obtient par exemple l'algorithme suivant :

$itermax$; τ_x ; τ_F ;
$iter \leftarrow 0$;
x ; $N_x \leftarrow \|x\|$; | *vecteur* $x^{(0)}$
$err_x \leftarrow 2 * N_x * \tau_x$; $err_F \leftarrow 2 * \tau_F$
`Tant que` $(iter < itermax)$ `et` $(err_x > \tau_x * N_x)$ `et` $(err_F > \tau_F)$
‖ $M \leftarrow J_F(x)$; $s \leftarrow -F(x)$;
‖ `Résoudre` $Md = s$;
‖ $err_x \leftarrow \|d\|$; $err_F \leftarrow \|F(x)\|$;
‖ $x \leftarrow x + d$; $N_x \leftarrow \|x\|$;
‖ $iter \leftarrow iter + 1$;
`Si` $iter == itermax$
‖ `Pas de convergence. FIN`
`Solution acceptée` $\longleftarrow x$

Dans cet algorithme, on a proposé de tester à la fois la convergence de la suite des $x^{(k)}$ et la réalisation de l'objectif. Les logiciels proposent en général ces deux options, et par défaut, vérifient les deux.

En pratique, on privilégie la première lorsque F varie très lentement au voisinage de la solution et la seconde dans le cas contraire.

La suite que construit l'algorithme ne converge pas toujours. Cela dépend du choix de $x^{(0)}$. Mais lorsqu'elle converge, cette convergence est quadratique : il existe une constante positive c telle qu'à partir d'un certain rang, les itérés vérifient :

$$\left\|x^{(k+1)} - x^*\right\| \leq c \left\|x^{(k)} - x^*\right\|^2. \tag{3.16}$$

A partir de ce même rang, on peut donc s'attendre au doublement du nombre de chiffres exacts dans l'approximation de la solution x^* à chaque itération.

3.3.2 Un exemple

On considére le système d'équations non linéaires définies par

$$\left\{ \begin{array}{lcl} 3\,x_1 - 2\,x_1^2 - 2\,x_2 + 1 & = & 0, \\ -x_1 + 3\,x_2 - 2\,x_2^2 - 2\,x_3 + 1 & = & 0, \\ -x_2 + 3\,x_3 - 2\,x_3^2 + 1 & = & 0. \end{array} \right. \tag{3.17}$$

La fonction $F(x)$ définissant le problème (3.17) est définie sur $\mathbb{R}^3$ à valeurs dans $\mathbb{R}^3$. Sa matrice jacobienne est donnée par

$$J_F = \left[\begin{array}{ccc} 3 - 4\,x_1 & -2 & 0 \\ -1 & 3 - 4\,x_2 & -2 \\ 0 & -1 & 3 - 4\,x_3 \end{array} \right]. \tag{3.18}$$

En choisissant $x^{(0)} = (0,\ 0,\ 0)$ comme coordonnées de départ, la méthode converge assez rapidement. Le tableau ci-dessous représente les $x_3^{(k)}$ et les valeurs successives de err_x et err_F.

k	$x_3^{(k)}$	err_x	err_F
1	$-7.3333333333333328\ 10^{-1}$	$1.00000\ 10^{0}$	$1.73205\ 10^{0}$
2	$-4.6814543424982669\ 10^{-1}$	$6.87222\ 10^{-1}$	$4.00629\ 10^{0}$
3	$-4.1352485141794909\ 10^{-1}$	$2.07561\ 10^{-1}$	$6.82736\ 10^{-1}$
4	$-4.1032645197702611\ 10^{-1}$	$1.52546\ 10^{-2}$	$4.43842\ 10^{-2}$
5	$-4.1031222321822403\ 10^{-1}$	$8.11407\ 10^{-5}$	$2.39587\ 10^{-4}$
6	$-4.1031222286858426\ 10^{-1}$	$2.29702\ 10^{-9}$	$6.94778\ 10^{-9}$
7	$-4.1031222286858421\ 10^{-1}$	$5.89752\ 10^{-17}$	$3.14018\ 10^{-16}$

On y constate que :
- le comportement des deux termes d'erreurs révèle bien une convergence à l'ordre 2 lorsqu'on se rapproche de la solution,
- dès que la méthode commence à converger, on double le nombre de chiffres corrects pour la solution approchée à chaque itération.

Remarque 3.2 *Dans l'algorithme de la méthode, la dernière ligne n'est pas anodine. La plupart du temps, lorsque la méthode ne converge pas, le résultat qu'elle fournit sera clairement identifié comme incorrect (*``Inf'' *ou* ``NaN''*), mais cela n'est pas toujours le cas.*

Ainsi, pour le système (3.17), la méthode ne converge pas toujours, mais comme pour l'exemple (3.10), les itérations peuvent se poursuivre indéfiniment sans que les valeurs calculées n'explosent. La figure 3.10 montre la répartition de 5000 itérés calculés en initialisant les itérations respectivement par $x^{(0)} = (-1,\ 1,\ 0.5)$ *et* $x^{(0)} = (-1,\ -1,\ 2)$*. Le dernier itéré calculé est dans le premier cas* $(2.7042,\ 1.6605, 0.4225)$ *: rien n'indique qu'il est assez éloigné d'une solution, sauf la valeur de la variable* `errF` *qui est environ* 15.4 *!*

En fait les points $x^{(k)}$ *se répartissent assez curieusement dans* $\mathbb{R}^3$*. Ils semblent rester sur un segment de droite. Les deux segments que l'on observe sur la figure sont les seuls que l'on ait obtenu en faisant varier le choix de* $x(0)$*. Ils sont respectivement à peu près parallèles aux vecteurs* $[0.9582,\ -0.2633,\ -0.1121]^T$ *et* $[0.5361,\ 0.8106,\ -0.2357]^T$*. Dans les deux cas, il arrive que l'on passe assez près de la solution calculée précédemment, qui est représentée par une* $*$*.*

L'explication de ce phénomène dépasse largement le cadre de ce cours.

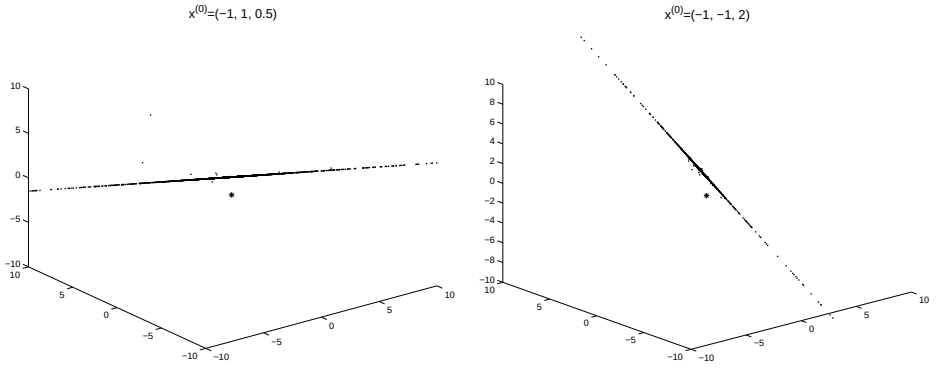


FIGURE 3.10 – La non convergence n'entraine pas l'explosion.

3.4 Annexe : Rappels de Calcul différentiel

3.4.1 Formules de Taylor

Pour une fonction $f : \mathbb{R} \longrightarrow \mathbb{R}$, les formules de Taylor s'écrivent

$$f(x+h) = f(x) + h\,f'(x) + \frac{h^2}{2}\,f''(x) + \ldots + \frac{h^n}{n!}\,f^{[n]}(x) + \mathcal{O}(h^{n+1}). \qquad (3.19)$$

– La dérivée $f'(x)$ au point x définit le terme linéaire de ce développement, $h \to f'(x)\,h$.
– La dérivée $f''(x)$ définit le terme quadratique de ce développement, $h \to \frac{1}{2}\,h\,f''(x)\,h$.

$$g(h) = \mathcal{O}(h^p) \;\Leftrightarrow\; \begin{cases} \exists a > 0,\ \exists M > 0, |h| < a \Rightarrow \left|\dfrac{g(h)}{h^p}\right| \leq M \ : \\ \text{“}g(h) \textit{ tend vers } 0 \textit{ avec } h \textit{ aussi vite que } h^p\text{“}. \end{cases}$$

$$g(h) = o(h^p) \;\Leftrightarrow\; \begin{cases} \lim\limits_{h\to 0} \dfrac{g(h)}{h^p} = 0, \\ \text{“}g(h) \textit{ tend vers } 0 \textit{ avec } h \textit{ plus vite que } h^p\text{“} \end{cases}$$

Pour une fonction $f : \mathbb{R}^p \longrightarrow \mathbb{R}$, les formules de Taylor s'écrivent

$$\begin{aligned} f(x+h) &= f(x) + \sum_{j=1}^{p} h_j\,\partial_{x_j} f(x) + \frac{1}{2}\sum_{j=1}^{p}\sum_{k=1}^{p} h_j\,h_k\,\partial^2_{x_j\,x_k} f(x) + \mathcal{O}(\|h^3\|), \\ &= f(x) + \sum_{j=1}^{p} h_j\,\partial_{x_j} f(x) + \frac{1}{2}\sum_{j=1}^{p} h_j \sum_{k=1}^{p} h_k\,\partial^2_{x_k\,x_j} f(x) + \mathcal{O}(\|h^3\|), \end{aligned}$$

- $J_f(x)$ = Matrice Jacobienne = $\big[\partial_{x_1} f(x), \ldots, \partial_{x_p} f(x)\big] = (\nabla f(x))^T$.
  Le gradient de f est le vecteur transposé de sa matrice jacobienne.
- $H_f(x)$ = Matrice Hessienne = $\Big[\partial^2_{x_i\, x_j} f(x)\Big]_{1\le i,j\le p} \in M_p(\mathbb{R})$. Elle est symétrique.

$$f(x+h) = f(x) + J_f(x)\, h + \frac{1}{2}\, h^T\, H_f(x)\, h + \mathcal{O}(\|h^3\|). \tag{3.20}$$

- La notation $A\,B$ désigne le produit matriciel. Dans le cas du produit d'une matrice ligne par une matrice colonne, on peut utiliser une notation de produit scalaire de deux matrices (vecteurs) colonnes

$$(3.20) \Leftrightarrow f(x+h) = f(x) + \langle \nabla f(x),\, h\rangle + \frac{1}{2}\, \langle H_f(x)\, h,\, h\rangle + \mathcal{O}(\|h^3\|). \tag{3.21}$$

Pour une fonction f : $\mathbb{R}^{\mathbf{p}} \longrightarrow \mathbb{R}^{\mathbf{n}}$, les formules de Taylor s'écrivent

$$f(x+h) = f(x) + L_f(h) + \frac{1}{2}\, Q_f(h) + \mathcal{O}(\|h^3\|). \tag{3.22}$$

- $x,\, h \in \mathbb{R}^p$ et $f(x) \in \mathbb{R}^n$: (3.22) est une égalité dans $\mathbb{R}^n$. $L_f(h)$ désigne un terme (vecteur de $\mathbb{R}^n$) linéaire en h, $Q_f(h)$ désigne un terme quadratique en h. Pour les expliciter, on peut écrire une formule de type (3.20) pour chaque composante f_i de f ; pour $1 \le i \le n$,

$$f_i(x+h) \;\; = \;\; f_i(x) + J_{f_i}(x)\, h + \frac{1}{2}\, h^T\, H_{f_i}(x)\, h + \mathcal{O}(\|h^3\|).$$

- En rassemblant toutes les lignes et suivant les règles du produit matrice-vecteur, on obtient $L_f(h) = J_f(x)\, h$ avec

$$J_f(x) = \text{Matrice Jacobienne} = \begin{bmatrix} J_{f_1}(x) \\ \cdots \\ \\ \cdots \\ J_{f_n}(x) \end{bmatrix} \in M_{n,p}(\mathbb{R}).$$

- Pour chaque i, le terme quadratique du développement de f_i est donné par $\dfrac{1}{2}\, h^T\, H_{f_i}(x)\, h$. En rassemblant toutes les lignes et en factorisant h à droite, on obtient :

$$\begin{bmatrix} h^T\, H_{f_1}(x)\, h \\ \cdots \\ \\ \cdots \\ h^T\, H_{f_n}(x)\, h \end{bmatrix} = \begin{bmatrix} \displaystyle\sum_{k=1}^{p} h_k\, \partial^2_{x_k\, x_1} f_1(x) & \ldots & \displaystyle\sum_{k=1}^{p} h_k\, \partial^2_{x_k\, x_p} f_1(x) \\ \cdots & & \cdots \\ & & \\ \cdots & & \cdots \\ \displaystyle\sum_{k=1}^{p} h_k\, \partial^2_{x_k\, x_1} f_n(x) & \ldots & \displaystyle\sum_{k=1}^{p} h_k\, \partial^2_{x_k\, x_p} f_n(x) \end{bmatrix} \begin{bmatrix} h_1 \\ \cdots \\ \\ \cdots \\ h_p \end{bmatrix}. \tag{3.23}$$

– On a obtenu une expression de $Q_f(h)$ qui permet d'écrire (3.22) sous la forme

$$f(x+h) = f(x) + \begin{bmatrix} J_{f_1}(x) \\ \cdots \\ \\ \cdots \\ J_{f_n}(x) \end{bmatrix} h + \frac{1}{2} \begin{bmatrix} h^T H_{f_1}(x) \\ \cdots \\ \\ \cdots \\ h^T H_{f_n}(x) \end{bmatrix} h + \mathcal{O}(\|h^3\|).$$

3.4.2 Application aux problèmes de moindres carrés

Problème de moindres carrés

– $F \ : \ \mathbb{R}^p \longrightarrow \mathbb{R}^n$, $n > p$: $F(x) = \begin{bmatrix} F_1(x_1, \ \ldots, \ x_p) \\ \ldots \ldots \\ F_n(x_1, \ \ldots, \ x_p) \end{bmatrix}$.

– $r(x) = \dfrac{1}{2} \|F(x)\|^2 = \dfrac{1}{2} \ \langle F(x), \ F(x) \rangle$.

– On cherche $x^* = \text{argmin} \left\{ r(x) \right\} \Leftrightarrow r(x^*) = \min \left\{ \dfrac{1}{2} \displaystyle\sum_{i=1}^{n} F_i^2(x), \ x \in \mathbb{R}^p \right\}$.

Utilisation de la méthode de Newton

– Condition nécessaire d'optimalité du premier ordre : $\nabla r(x^*) = 0$.
– Résolution par la méthode de Newton : pour $k \geq 0$,

$$x^{(k+1)} = x^{(k)} - \left(H_r(x^{(k)}) \right)^{-1} \nabla r(x^{(k)}). \tag{3.24}$$

Calcul de $\nabla r(x)$: on peut le faire en **identifiant de la partie linéaire** d'un développement au voisinage de x :

$$\begin{aligned} 2 \ r(x+h) &= \langle F(x+h), \, F(x+h) \rangle \, , \\ &= \langle F(x) + J_F(x) \, h + \mathcal{O}(\|h\|^2), \, F(x) + J_F(x) \, h + \mathcal{O}(\|h\|^2), \rangle \, , \\ &= \langle F(x), \, F(x) \rangle + \langle F(x), \, J_F(x) \, h \rangle + \langle J_F(x) \, h, \, F(x) \rangle + \mathcal{O}(\|h\|^2). \end{aligned}$$

– Comme en dimension 1, on assimile les termes qui seront au moins $\mathcal{O}(\|h\|^2)$: $\left\langle F(x), \, \mathcal{O}(\|h\|^2) \right\rangle + \ \langle J_F(x) \, h, \, J_F(x) \, h \rangle + \ldots = \mathcal{O}(\|h\|^2)$.
– Pour $A \in M_{n,p}(\mathbb{R})$, $x \in \mathbb{R}^n$ et $y \in \mathbb{R}^p$, on a $\langle x, \, A \, y \rangle = \left\langle A^T \, x, \, y \right\rangle$:

$$r(x+h) \ = \ r(x) + \left\langle J_F(x)^T \, F(x), \, h \right\rangle + \mathcal{O}(\|h\|^2).$$

– Par identification,

$$\nabla r(x) = J_F(x)^T \, F(x). \tag{3.25}$$

Calcul de $\nabla r(x)$: on peut le faire **en calculant les dérivées partielles** :

– Pour tout i, $1 \le i \le p$,

$$\partial_{x_i} r(x) = \partial_{x_i} \left(\frac{1}{2} \sum_{j=1}^{n} F_j^2(x) \right) = \sum_{j=1}^{n} \left(\partial_{x_i} F_j(x) \right) F_j(x).$$

– En prenant en compte toute les lignes,

$$\begin{bmatrix} \partial_{x_1} r(x) \\ \dots \\ \partial_{x_p} r(x) \end{bmatrix} = \begin{bmatrix} \partial_{x_1} F_1(x) & \dots & \partial_{x_1} F_n(x) \\ \dots & & \dots \\ \partial_{x_p} F_1(x) & \dots & \partial_{x_p} F_n(x) \end{bmatrix} \begin{bmatrix} F_1(x) \\ \dots \\ F_n(x) \end{bmatrix} = \sum_{j=1}^{n} (\nabla F_j)\ F_j = J_F^T\, F.$$

Calcul de $H_r(x)$: on peut le faire par **identification de la partie quadratique** d'un développement au voisinage de x :

$$\begin{aligned} r(x+h) &= \frac{1}{2} \left\langle F(x) + J_F(x)\, h + \frac{1}{2} Q_r(h) + \mathcal{O}(\|h\|^3),\ F(x) + J_F(x)\, h + \frac{1}{2} Q_F(h) + \mathcal{O}(\|h\|^3), \right\rangle, \\ &= r(x) + \langle \nabla r(x),\ h \rangle + \frac{1}{2}\ \langle J_F(x)\, h,\ J_F(x)\, h \rangle \dots \\ \dots &+ \frac{1}{4}\ \langle F(x),\ Q_F(h) \rangle + \frac{1}{4}\ \langle Q_F(h),\ F(x) \rangle + \mathcal{O}(\|h\|^3), \end{aligned} \tag{3.26}$$

– La partie quadratique de (3.26) s'identifie selon

$$\langle H_r(x)\, h,\ h \rangle = \left\langle J_F(x)^T\, J_F(x)\, h,\ h \right\rangle + \frac{1}{2}\ \langle F(x),\ Q_F(h) \rangle$$

– Suivant (3.23), $\langle F(x),\ Q_F(h) \rangle = \left(\sum_{i=1}^{n} F_i(x)\, h^T\, H_{F_i}(x) \right) h$, d'où

$$H_r = J_F^T\, J_F + \sum_{i=1}^{n} F_i\, H_{F_i}. \tag{3.27}$$

Calcul de $H_r(x)$: on peut le faire **en dérivant $\nabla r(x)$** :

– $\nabla r = \sum_{j=1}^{n} (\nabla F_j)\ F_j$.

– On utilise la règle de dérivation d'un produit : $(\nabla r)' = \sum_{j=1}^{n} \left(H_{F_j}\, F_j + (\nabla F_j)\, J_{F_j} \right)$.

– $J_{F_j} = (\nabla F_j)^T$, et $\sum_{j=1}^{n} (\nabla F_j)\ (\nabla F_j)^T = \left[\ \nabla F_1 \ \middle|\ \dots \ \middle|\ \nabla F_n\ \right] \begin{bmatrix} \nabla F_1^T \\ \dots \\ \nabla F_n^T \end{bmatrix} = J_F^T\, J_F$.

Calcul de $H_r(x)$: on peut **laborieusement** le faire en calculant les dérivées partielles secondes ; on ne le fait pas.

3.4.3 Exemples

– $F(x) = A\,x - b$, avec $A \in M_{n,p}(\mathbb{R})$: on obtient le système des équations normales

$$A^T\,A\,x^* = A^T\,b.$$

– Pour $1 \leq i \leq n$, $F_i(x_0,\, x_1,\, x_2) = c_i - x_0 - x_1\,e^{-x_2\,t_i}$. On a donc $F : \mathbb{R}^3 \longrightarrow \mathbb{R}^n$.

$$\nabla F_i = \begin{bmatrix} -1 \\ -e^{-x_2\,t_i} \\ x_1\,t_i\,e^{-x_2\,t_i} \end{bmatrix} \text{ et } H_{F_i} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & t_i\,e^{-x_2\,t_i} \\ 0 & t_i\,e^{-x_2\,t_i} & -x_1\,t_i^2\,e^{-x_2\,t_i} \end{bmatrix}.$$

$$J_r(x) = \sum_{i=1}^{n} (\nabla F_i(x))\; F_i(x), \text{ et } H_r = J_F(x)^T\,J_F(x) + \sum_{i=1}^{n} H_{F_i}(x)\,F_i(x).$$

Chapitre 4

ÉQUATIONS DIFFÉRENTIELLES ORDINAIRES

4.1 Equations Différentielles

4.1.1 Différents types de problèmes différentiels

Les équations différentielles constituent, avec les équations aux dérivées partielles, l'outil de base pour la description des modèles mathémathiques de phénomènes rencontrés dans la nature. Le terme d'**équations différentielles ordinaires** est employé pour désigner les équations différentielles qui peuvent se mettre sous la forme

$$y^{(p)}(t) = f(t,\, y(t),\, \ldots,\, y^{(p-1)}(t)) \tag{4.1}$$

par opposition à celles qui s'expriment seulement de façon implicite, appelées équations différentielles algébriques,

$$f(t,\, y(t),\, y'(t), \ldots,\, y^{(p)}(t)) = 0.$$

L'équation (4.1) est d'ordre p ; elle peut admettre une solution générale, mais il faut ajouter p conditions pour espérer une solution unique.

Un problème différentiel sera toujours constitué d'équation(s) et de condition(s). Plus que l'ordre de l'équation, c'est la nature des conditions qui caractérise le type de problème.

- Pour des équations ou des systèmes du premier ordre, la condition fixe la valeur en un point de la solution. On peut avoir un **problème à valeur initiale** :

$$\left\{ \begin{array}{lcl} y'(t) & = & f(t, y(t))\,; \quad t \in [0,\, T], \quad y \in \mathbb{R}^d, \\ y(0) & = & \alpha \in \mathbb{R}^d. \end{array} \right. \tag{4.2}$$

- Les problèmes du second ordre peuvent être des **problèmes aux limites** comme par exemple :

$$\begin{cases} y''(x) = f(x,\, y(x),\, y'(x))\ ;\ x \in [a,\, b], \\ y(a) = \alpha\ ;\ y(b) = \beta. \end{cases} \tag{4.3}$$

- Les problèmes du second ordre, ou d'ordre supérieur peuvent être également des problèmes à valeur(s) initiale(s) ; on les transforme alors en problèmes d'ordre 1 de taille supérieure. Partant d'une équation différentielle, on forme un **système différentiel** ; c'est le cas pour le problème :

$$\begin{cases} y''(t) = f(t,\, y(t),\, y'(t))\,;\ t \in [0, T], \\ y(0) = \alpha\,;\ y'(0) = \beta. \end{cases}$$

que l'on écrira sous la forme :

$$\begin{cases} Y'(t) &= F(t,\, Y(t))\ ;\ t \in [0,\, T], \\ Y(0) &= Y_0, \end{cases}$$

avec $Y = \begin{bmatrix} y(t) \\ y'(t) \end{bmatrix} = \begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix}$ et $Y_a = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$. On obtient :

$$Y'(t) = \begin{bmatrix} y_2(t) \\ f(t,\, y_1(t),\, y_2(t)) \end{bmatrix}$$

Ce chapitre est consacré aux problèmes à valeurs initiales, et on n'y taitera pas les problèmes de la forme (4.3). On a choisi d'appeler t la variable dans ce cas, car, le plus souvent, elle représente souvent le temps. Les problèmes à valeur(s) initiale(s) seront toujours étudiés sur un intervalle de la forme $[0,\, T]$.

Exemple 4.1 *Le problème*

$$\begin{cases} y''(t) - 4\,y'(t) + 4\,y(t) = \exp(2\,t),\ 0 \le t \le 1, \\ y(0) = 0\ ;\ y'(0) = 1. \end{cases}$$

s'écrit sous la forme $Y'(t) = \begin{bmatrix} 0 & 1 \\ -4 & 4 \end{bmatrix} Y(t) + \begin{bmatrix} 0 \\ \exp(t) \end{bmatrix}$, *avec* $Y(0) = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

Exemple 4.2 : Le problème du pendule.

Le mouvement d'un pendule de masse m, suspendu à un point O par un fil non pesant de longueur l, en rotation d'angle $\theta(t)$ autour de O est gouverné par l'équation :

$$\theta''(t) = -\frac{g}{l}\,\sin(\theta(t)). \tag{4.4}$$

L'angle θ est mesuré par rapport à une verticale passant par O. On s'intéresse au mouvement entre les instants $t_0 = 0$ et un instant final T. Les conditions initiales

peuvent être $\theta_0 = \dfrac{\pi}{3}$ *rad. et* $\theta'(0) = 0$ rad s^{-1}. *En général, on introduit* $\omega^2 = \dfrac{g}{l}$. *Ce problème est un problème non linéaire.*

L'équation (4.4) est d'ordre 2, mais les conditions données en font un problème à valeurs initiales ; il doit donc être transformé en un système de deux équations différentielles du premier ordre. On pose $y_1(t) = \theta(t)$ *et* $y_2 = \theta'(t)$. *On obtient :*

$$\begin{array}{rcl} y_1'(t) & = & y_2(t) \\ y_2'(t) & = & -\omega^2 \sin(y_1(t)) \end{array}$$

Ce système entre dans le cadre de l'équation (4.2) avec $Y = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}$ *et*

$$F = \begin{bmatrix} F_1(t,\, Y) \\ F_2(t,\, Y) \end{bmatrix} = \begin{bmatrix} y_2 \\ -\omega^2 \sin y_1 \end{bmatrix} \tag{4.5}$$

Exemple 4.3 : Modèles proie-prédateur.

Un modèle simple à deux espèces est le **modèle de Lotka-Volterra**. *Il s'agit d'un modèle de système biologique à deux espèces, une proie et un predateur. On peut par exemple imaginer des lapins et des renards. Si* $L(t)$ *et* $R(t)$ *désignentles populations respectives de lapins et renards, ce sont des fonctions à valeurs dans* $\mathbb{N}$. *On préfère donc travailler avec des densités, que l'on notera* $l(t)$ *et* $r(t)$.

$$\begin{cases} \dfrac{d\, l(t)}{dt} & = & k_1\ l(t) - k_2\, l(t)\, r(t), \\ \dfrac{d\, r(t)}{dt} & = & e\, k_2\, l(t)\, r(t) - k_3\, r(t). \end{cases}$$

On peut expliciter ce modèle :

- k_1 *désigne le taux de croissance de la proie, qui est constant en l'absence de prédateurs ; dans ce cas, la population croit exponentiellement vite (dynamique de Malthus).*
- *le taux* k_2 *de disparition des proies est proportionnel au nombre de rencontres, lui même proportionnel aux deux populations.*
- *le taux* k_3 *de disparition des prédateurs en l'absence de proies est lui aussi constant*
- *il y a un coefficient d'éfficacité* e *de l'assimilation des proies par les prédateurs.*

En posant $y_1(\tau) = \dfrac{e\, k_2}{k_3}\, l(t),\ y_2(\tau) = \dfrac{k_2}{k_1}\, r(t),\ \gamma = \dfrac{k_3}{k_1}$ *avec* $\tau = k_1\, t$, *on obtient une forme analogue à (4.2) avec :*

$$F(\tau,\, Y) = \begin{bmatrix} F_1(\tau,\, Y) \\ F_2(\tau,\, Y) \end{bmatrix} = \begin{bmatrix} y_1(\tau)\ (1 - y_2(\tau)) \\ \gamma\, y_2(\tau)\, (y_1(\tau) - 1) \end{bmatrix} \tag{4.6}$$

Avec des constantes adéquates, et une valeur initiale que n'est pas un état d'équilibre ($[y_1,\, y_2] = [0,\, 0]$ *ou* $[1,\, 1]$*), on obtient une solution périodique. On peut s'en convaincre en représentant l'espace des phases, ici plan de phases car on a un système de deux équations, c'est-à-dire en représentant les populations l'une par rapport à l'autre.*

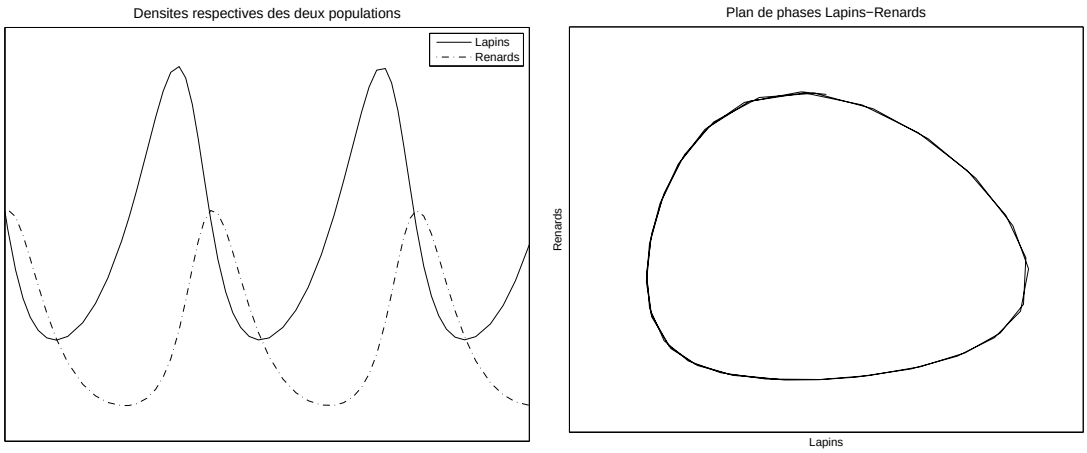


FIGURE 4.1 – Le modèle prédateur-proie de Lokta-Volterra.

On peut enrichir se modèle en supposant qu'en absence de prédateurs, les proies ont une croissance de type logistique, en remplaçant le terme $k_1\, l(t)$ par $k_1\, l(t) \left(1 - \dfrac{l(t)}{\kappa_l}\right)$; la constante κ_l, dite capacité biotique, représente la densité limite des proies en absence de prédateurs. On obtient le **modèle de proie-prédateur logistique**. C'est un **exercice** de l'écrire sous une forme du type (4.2).

4.1.2 Problèmes de Cauchy

Lorsque la fonction f est continue sur $[0,\, T] \times \mathbb{R}^d$, les problèmes du type (4.2) s'appellent aussi problèmes de Cauchy.

Théorème 4.1 *Si f est continue sur $\Omega = [0,\, T] \times \mathbb{R}^d$ et s'il existe une constante $L \geq 0$ telle que $\|f(t,\, y) - f(t,\, z)\| \leq L\, \|y - z\|$, $\forall t \in [a,\, b]$, $\forall (y,\, z) \in \mathbb{R}^{2d}$, le problème (4.2) admet une solution unique pour toute donnée initiale $\alpha \in \mathbb{R}^d$.*

La seconde condition s'appelle **condition de Lipschitz** par rapport à la variable y. Dans $\mathbb{R}^d$, on peut choisir la norme que l'on veut. Bien que ce théorème nous assure de l'existence d'une solution, il est souvent impossible de la calculer explicitement.

Il existe d'autres version de ce théorème, et en particulier, la condition de Lipschitz peut n'être que locale, avec alors existence d'une solution unique seulement sur un intervalle $[0,\, T_e[$, où T_e désigne un "temps d'explosion", $T_e \leq T$.

Exemple 4.4 *Le problème du pendule satisfait les hypothèses du théorème 4.1. La fonction $F(t,\, Y) = \begin{bmatrix} y_2 \\ -\omega^2 \sin y_1 \end{bmatrix}$ de (4.5) est de classe $\mathcal{C}^1$, et on peut utiliser le théorème des accroissements finis pour établir la condition de Lipschitz locale :*

$$\partial_Y F(t,\, Y) = J_Y = \begin{bmatrix} \dfrac{\partial F_1}{\partial y_1} & \dfrac{\partial F_1}{\partial y_2} \\ \dfrac{\partial F_2}{\partial y_1} & \dfrac{\partial F_2}{\partial y_2} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\omega^2 \cos y_1 & 0 \end{bmatrix}.$$

Le théorème des accroissement finis permet d'écrire

$$\|F(t,\, Y) - F(t,\, Z)\| \leq \max_{x \in \mathbb{R}^2} \|J_Y(x)\|\, \|Y - Z\|,$$

d'où le résultats, avec $\|J_Y\|_1 = \|J_Y\|_\infty = \|J_Y\|_2 \leq \max\{1,\, \omega^2\} = L$.

Si on a peur du calcul différentiel et des normes matricielle, on peut aussi écrire :
- $|F_1(Y) - F_1(Z)| = |y_2 - z_2| \leq \|Y - Z\|$,
- $|F_2(Y) - F_2(Z)| = \omega^2\, |\sin(y_1) - \sin(z_1)| \leq \omega^2\, \|Y - Z\|$, *puisque*

$$|\sin(y_1) - \sin(z_1)| = 2\, |\sin(\frac{y_1 - z_1}{2})\, \cos(\frac{y_1 + z_1}{2})| \leq |y_1 - z_1| \leq \|Y - Z\|.$$

On ne peut cependant pas expliciter la solution de ce problème à l'aide de fonctions usuelles. Cette solution peut s'exprimer à l'aide d'une fonction spéciale appelée fonction "sinus amplitude" de Jacobi, mais c'est beaucoup plus compliqué que d'en rechercher une approximation numérique en utilisant une des méthodes que nous allons exposer dans ce cours.

Exemple 4.5 *Le probléme* $\left\{ \begin{array}{lcl} y'(t) & = & y(t)^{1/2}\,;\ t \in [0,\, 1], \\ y(0) & = & 0, \end{array} \right.$ *ne satisfait pas les hypothèses du théorème 4.1. Il admet au moins deux solutions distinctes* $y_1(t) = 0$ *et* $y_2(t) = \left(\dfrac{t}{2}\right)^2$.

Exemple 4.6 *Le problème (4.6) est un problème de Cauchy. La fonction* F *qui définit ce problème ne satisfait pas une condition de Lipschitz globale* $\|F(t,\, y) - F(t,\, z)\| \leq L\, \|y - z\|$ *pour tous* y *et* z *dans* $\mathbb{R}^2$*, mais elle satisfait une condition de Lipschitz locale ; c'est toujours le cas lorsque* F *est continument dérivable par rapport à* Y*, et ici, la matrice jacobienne de* F *(relativement à* y*) s'écrit :*

$$J_Y = \begin{bmatrix} 1 - y_2 & -y_1 \\ \gamma\, y_2 & \gamma\, (y_1 - 1) \end{bmatrix}.$$

4.1.3 Approximation numérique des solutions

On cherche à calculer des valeurs approchées de la solution $y(t)$ du problème (4.2). Pour cela, on se donne une partition $0 = t_0 < t_1 < ... < t_N = T$ et on cherche des y_i aussi voisins que possible des $y(t_i)$.

Les **pas de discrétisation** sont les réels $h_i = t_{i+1} - t_i$. Lorsque l'on n'aura pas expressément besoin d'un pas variable, on le supposera constant, et on le notera $h = \dfrac{T}{N}$.

Les méthodes d'Euler

Construction des méthodes d'Euler : En intégrant l'équation (4.2) sur l'intervalle $[t_n, t_{n+1}]$, on obtient :

$$y(t_{n+1}) = y(t_n) + \int_{t_n}^{t_{n+1}} y'(s)\, ds = y(t_n) + \int_{t_n}^{t_{n+1}} f(s, y(s))\, ds. \tag{4.7}$$

On peut remplacer l'intégrale par une formule approchée, et si on choisit la formule des rectangles à gauche :

$$\int_{t_n}^{t_{n+1}} f(s, y(s))\, ds \simeq h\, f(t_n,\, y(t_n)),$$

on obtient la formule d'Euler explicite :

$$y_{n+1} = y_n + h\, f(t_n, y_n)\ ;\ 0 \leq n \leq N-1.$$

On peut choisir une formule des rectangles à droite, et dans ce cas, c'est la méthode d'Euler implicite que l'on obtient :

$$y_{n+1} = y_n + h\, f(t_{n+1}, y_{n+1})\ ;\ 0 \leq n \leq N-1.$$

Cette méthode est dite implicite car y_{n+1} ne s'obtient qu'après la résolution d'une équation.

Ces deux méthodes sont dites à un pas, car le calcul de y_{n+1} se fait seulement en fonction des valeurs à l'étape n, $(y_n,\, t_n)$ et du pas h.

Exemple 4.7 *La méthode d'Euler explicite se construit aussi en considérant un développement de Taylor de la forme*

$$y(t_{n+1}) = y(t_n) + h\, y'(t_n) + \mathcal{O}(h^2) = y(t_n) + h\, f(t_n,\, y(t_n)) + \mathcal{O}(h^2). \tag{4.8}$$

On néglige le terme $\mathcal{O}(h^2)$ et on remplace les valeurs éxactes $y(t_n)$ par des valeurs approchées y_n. On obtient de façon analogue la méthode implicite.

Une mise en œuvre réussie : Le problème

$$\left\{ \begin{array}{rcl} y'(t) & = & y(t),\ 0 \leq t \leq 1, \\ y(0) & = & 1, \end{array} \right. \tag{4.9}$$

admet comme unique solution $y(t) = e^t$.

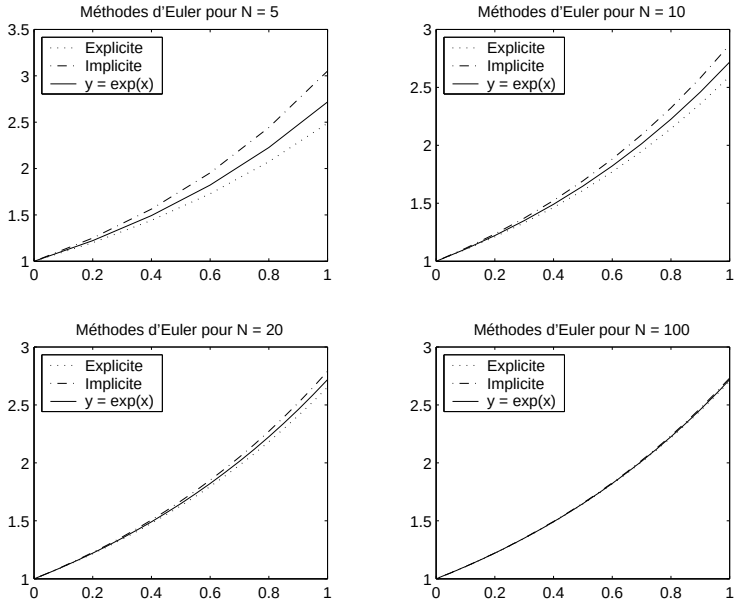


FIGURE 4.2 – Les méthodes d'Euler pour le problème (4.9)

La méthode d'Euler explicite consiste à calculer, pour $n \geq 0$, $y_{n+1} = y_n + h\, y_n = (1+h)\, y_n$, de sorte partant de $y_0 = 1$, on obtient $y_{n+1} = (1+h)^{n+1}$. Avec $h = \dfrac{1}{N}$, on retrouve pour y_N une valeur approchée de $y(1) = e$ puisque

$$\lim_{N\to\infty} y_N = \lim_{N\to\infty} (1 + \frac{1}{N})^N = e.$$

La méthode implicite fournit quant à elle $y_N = \dfrac{1}{(1 - \dfrac{1}{N})^N}$ qui converge également vers e lorsque N tend vers l'infini.

La figure 4.2 illustre cette convergence des deux méthodes ; on y constate que la méthode explicite calcule des valeurs légèrement inférieures aux $y(t_i)$, alors que la méthode implicite calcule des valeurs supérieures.

En fait, ces comportements s'expliquent par un développement limité :

$$\begin{cases} y_{N,e} &=& e^{N\,\ln(1+\frac{1}{N})} &=& e\left(1 - \dfrac{1}{2\,N} + \mathcal{O}(\dfrac{1}{N^2})\right), \\ y_{N,i} &=& e^{-N\,\ln(1-\frac{1}{N})} &=& e\left(1 + \dfrac{1}{2\,N} + \mathcal{O}(\dfrac{1}{N^2})\right). \end{cases}$$

Cette estimation est confortée par les valeurs données dans le tableau suivant, où e_N désigne l'erreur absolue pour la valeur $t = 1$ de la variable et pour chacune des méthodes.

	Euler explicite		**Euler implicite**	
N	ϵ_N	$\frac{\epsilon_N}{h}$	ϵ_N	$\frac{\epsilon_N}{h}$
10	0.1245	1.2454	0.1497	1.4969
25	0.0524	1.3111	0.0564	1.4110
50	0.0267	1.3347	0.0277	1.3845
100	0.0135	1.3468	0.0137	1.3717
250	0.0054	1.3542	0.0055	1.3641
500	0.0027	1.3567	0.0027	1.3616
1000	0.0014	1.3579	0.0014	1.3604

Les valeurs du rapport $\frac{e_N}{h}$ encadrent de mieux en mieux $\frac{e}{2} \simeq 1.3591$.

Une mise en œuvre défaillante : Considérons le problème suivant :

$$\left\{ \begin{array}{rcl} y'(t) & = & 3\,y(t) - 3\,t\ ;\ 0 \leq t \leq T, \\ y(0) & = & \frac{1}{3}, \end{array} \right. \tag{4.10}$$

dont la solution est $y(t) = t + \frac{1}{3}$ et appliquons la méthode d'Euler pour en chercher la solution approchée à l'instant T, pour différentes valeurs de T et différents choix du pas h. Les résultats obtenus peuvent surprendre. Le tableau suivant donne les erreurs dans le calcul de y_N :

h	$T = 5$	$T = 10$	$T = 20$
1	$1.8651\ 10^{-14}$	$1.9403\ 10^{-11}$	$2.0345\ 10^{-05}$
0.5	$4.9560\ 10^{-13}$	$4.7278\ 10^{-09}$	$4.2999\ 10^{-01}$
0.1	$5.2402\ 10^{-14}$	$2.6318\ 10^{-08}$	$6.5252\ 10^{+03}$
0.05	$1.5525\ 10^{-11}$	$1.8232\ 10^{-05}$	$2.5142\ 10^{+07}$

On constate que l'erreur semble croître avec le pas, contrairement à ce que l'on espérait. Par ailleurs, pour $T = 20$, cette erreur atteint des valeurs astronomiques !

Pour chercher à comprendre ce qui s'est passé, nous calculons la solution générale de l'équation différentielle $y'(t) = 3y(t) - 3t$. La solution de l'équation homogène est $y(t) = Ke^{3t}$. La méthode de variation de la constante fournit pour notre équation

$$\begin{array}{rcl} K'(t) & = & -3t\,e^{-3t}, \\ K(t) & = & (t + \frac{1}{3})\,e^{-3t} + k, \\ y(t) & = & t + \frac{1}{3} + k\,e^{3t}. \end{array}$$

La constante k peut être évaluée à l'aide de la valeur de y en 0, avec $k = 0$ si $y(0) = \frac{1}{3}$. Mais si $y(0) = \frac{1}{3} + \epsilon$, on obtient $k = \epsilon$ de sorte que la solution réelle du problème devient $y_\epsilon(t) = \frac{1}{3} + t + \epsilon\, e^{3t}$.

Les valeurs de e^{3T} sont respectivement :
- pour $T = 5$, $e^{3T} \simeq 3.269\ 10^6$,
- pour $T = 10$, $e^{3T} \simeq 1.068\ 10^{13}$,
- pour $T = 20$, $e^{3T} \simeq 1.142\ 10^{26}$.

Avec une unité d'arrondi de l'ordre de $1.1102\ 10^{-16}$, on retrouve l'ordre de grandeur des erreurs constatées ! Le problème (4.10) est typiquement un problème instable ; une petite perturbation de la donnée initiale entraine une grande perturbation de la solution.

4.1.4 Stabilité et comportement des solutions des problèmes linéaires

L'exemple (4.10) nous montre que le comportement des solutions du problème homogène associè à une équation différentielle linéaire peut avoir une grande importance dans le calcul approchè d'une solution de cette équation. La solution générale de l'équation linéaire homogène du premier ordre $y' = a\, y$ est $y(t) = k\, e^{a\, t}$, la constante k étant déterminée par la condition initiale. On constate que toute solution tendra vers l'infini avec t si $a > 0$, vers 0 si $a < 0$ et restera constante si $a = 0$.

Plaçons nous à présent dans le cas d'un problème de Cauchy défini par une équation du second ordre $y'' + \alpha\, y' + y = 0$.

Cette équation s'écrit comme un système du premier ordre,

$$Y' = \begin{bmatrix} y \\ y' \end{bmatrix}' = \begin{bmatrix} 0 & 1 \\ -1 & -\alpha \end{bmatrix} \begin{bmatrix} y \\ y' \end{bmatrix}. \tag{4.11}$$

Les valeurs propres $\lambda_{1,2}$ de la matrice $\begin{bmatrix} 0 & 1 \\ -1 & -\alpha \end{bmatrix}$ sont les racines du polynôme caractéristique de l'équation du second ordre, $r_{1,2} = \dfrac{-\alpha \pm \sqrt{\alpha^2 - 4}}{2}$. Les deux composantes de la solution auront le comportement représenté sur la figure 4.3.

La forme des solutions **à valeurs réelles** de cette équation va dépendre de α,
- si $\alpha^2 < 4$, les racines sont complexes et $y(t) = e^{-\frac{\alpha\, t}{2}}\ (a \cos \beta t + b \sin \beta t)$; le système a une composante périodique qui sera amortie si α est positif (cas stable) et amplifiée jusqu'à l'explosion sinon ;
- si $\alpha^2 > 4$, les racines sont réelles et le système n'a plus de comportement périodique. Sa solution est de la forme $y(t) = e^{-\frac{\alpha\, t}{2}}\ (a\, e^{\beta t} + b\, e^{-\beta t})$, avec $|\beta| < \left|\dfrac{\alpha}{2}\right|$, et le système reviendra ou non à son état d'équilibre selon que α sera négatif ou positif.

En présence d'un second membre, ces comportements ne se retrouvent pas nécessairement dans la solution exacte ; cela dépend en particulier des conditions initiales.

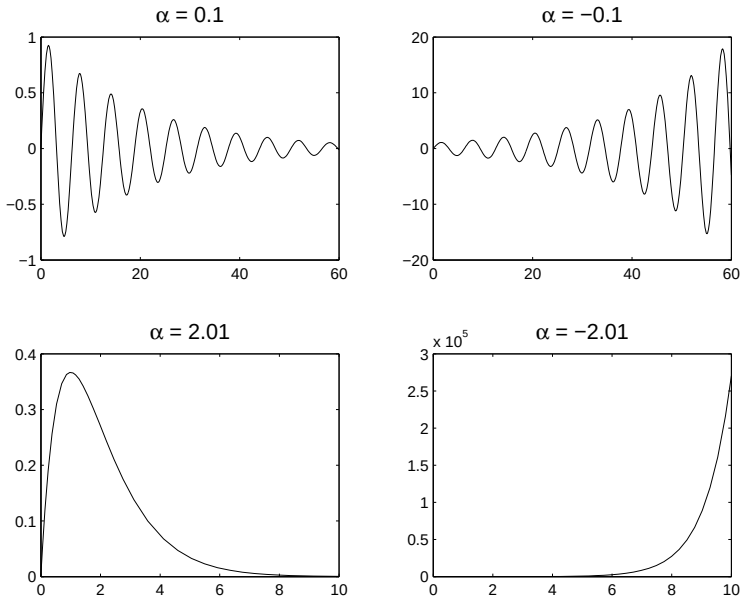


FIGURE 4.3 – Comportement des solutions du système (4.11)

Lorsque le système différentiel est de dimension supérieure à 2, de la forme $Y'(t) = A\,Y(t)$, le comportement des différentes composantes de la solution dépend encore des valeurs propres de la matrice A. Lorsque cette matrice est diagonalisable, c'est-à-dire lorsqu'il existe une matrice P inversible telle que $P^{-1}\,A\,P = \Lambda$ soit une matrice diagonale, dont les éléments diagonaux sont les valeurs propres de A, on obtient

$$P^{-1}\,Y'(t) = P^{-1}\,A\,P\,P^{-1}\,Y(t) = \Lambda\,P^{-1}\,Y(t),$$

de sorte que si $X(t) = P^{-1}\,Y(t)$ représente la fonction $Y(t)$ dans la base des colonnes de P, ses composantes $x_i(t)$ seront de la forme $x_i(t) = k_i e^{\lambda_i\,t}$. La encore, il apparait qu'une petite perturbation de ce problème, par exemple de ses conditions initiales, pourra produire une grosse perturbation des solutions si une des valeurs propres de la matrice définissant le système est de partie réelle positive.

Définition 4.1 *On dira qu'une équation ou un système différentiel linéaire est stable si toutes les valeurs propres de la matrice qui le définit sont à partie réelle négative ou nulle.*

Dans le cas non linéaire, ce comportement pourra varier au cours du temps. En fait, pour les solutions d'un système $Y'(t) = F(t, Y(t))$, c'est la matrice jacobienne de F relativement à Y qui détermine localement ce comportement. On ne dispose donc généralement pas de cette information.

Exemple 4.8 *Le comportement des solutions de (4.11) est déterminé par la valeur de α comme on l'a expliqué. Le cas où $0 < \alpha < 2$ correspond à un système oscillant avec amortissement de type visqueux (proportionnel à la vitesse). Le cas d'un système oscillant non amorti, pour lequel on aurait $\alpha = 0$ n'est pas trés réaliste physiquement. Si maintenant on remplace cette constante α par une fonction de y, qui serait négative pour y petit et positive pour y grand, on fera vivre durablement la solution, sans explosion ni amortissement : en choisissant par exemple $\alpha(y) = -\mu\,(1 - y^2)$, avec $\mu > 0$, on obtient l'équation*

$$y'' - \mu\,(1 - y^2)\,y' + y = 0 \tag{4.12}$$

Cette équation est connue sous le nom d'équation de Van der Pol. Ses solutions ne peuvent pas s'exprimer à l'aide de fonctions usuelles. La figure 4.4 montre la solution obtenue pour $\mu = 20$ avec comme conditions initiales $y(0) = 2$ et $y'(0) = 0$.

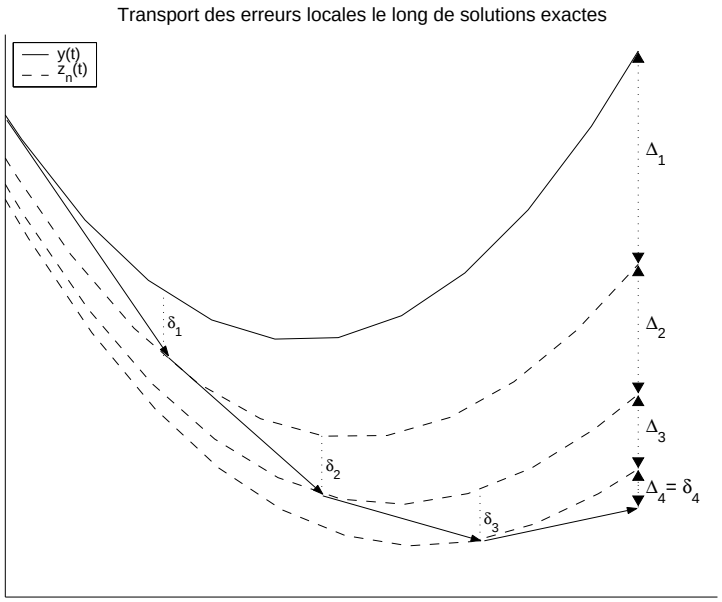


FIGURE 4.4 – Un exemple de solution de (4.12).

L'équation (4.12) est équivalente au système $Y' = \begin{bmatrix} y_1 \\ y_2 \end{bmatrix}' = \begin{bmatrix} y_2 \\ \mu\,(1 - y_1^2)\,y_2 - y_1 \end{bmatrix}$.
Les oscillations obtenues sont remarquablement stables, au point que ce modèle d'oscillateur a été utilisé pour la réalisation de stimulateurs cardiaques.

4.1.5 Les méthodes explicites de type Runge-Kutta

Forme générale des méthodes explicites à un pas

Nous restons dans le cadre général du problème (4.2) en supposant que les hypothéses du théoréme 4.1 sont vérifiées. On parle de méthode à un pas lorsque le

calcul de la valeur approchée y_{n+1} de $y(t_{n+1})$ ne fait intervenir que y_n, t_n et le pas h_n. Par opposition, les méthodes multipas utilisent aussi des valeurs calculées aux étapes précédentes.

Dans le cas d'un pas constant $h = \dfrac{T}{N}$, la forme générale des schémas à un pas est donc la suivante :

$$\left\{ \begin{array}{lcl} y_0 & = & \alpha, \\ y_{n+1} & = & y_n + h\,\Phi(t_n,\, y_n,\, y_{n+1},\, h), \quad 0 \leq n \leq N-1. \end{array} \right. \tag{4.13}$$

C'est la fonction Φ qui caractérise le schéma. Pour les méthodes explicites, elle ne dépend pas de y_{n+1} ; on la note $\Phi(t,\, y,\, h)$. Pour la méthode d'Euler explicite, $\Phi(t,\, y,\, h) = f(t,\, y)$.

Exemple 4.9 *On peut penser à utiliser la formule des trapèzes pour approcher l'intégrale figurant dans (4.7). On obtient alors un schéma implicite défini par :*

$$y_{n+1} = y_n + \frac{h}{2}\left(f(t_n,\, y_n) + f(t_{n+1},\, y_{n+1})\right). \tag{4.14}$$

C'est la méthode implicite du trapèze, ou **méthode de Crank-Nicholson**. *Si l'on souhaite éviter le caractère implicite de ce schéma, on remplace $f(t_{n+1},\, y_{n+1})$ par l'estimation qu'en donne le schéma d'Euler. On obtient la méthode du trapèze explicite :*

$$y_{n+1} = y_n + \frac{h}{2}\left(f(t_n,\, y_n) + f(t_{n+1},\, y_n + h\, f(t_n,\, y_n))\right). \tag{4.15}$$

Pour cet exemple, on peut expliciter $\Phi(t,\, y,\, h) = \dfrac{1}{2}\,\left(f(t,\, y) + f(t+h,\, y + h\, f(t,\, y))\right)$.

Les méthodes de type Runge-Kutta

Notation algorithmique

Le schéma explicite (4.15) possède une écriture très compacte, et on peut l'allèger en introduisant une **notation algorithmique** :

$$\left\{ \begin{array}{lcl} k_1 & = & f(t_n,\, y_n), \\ k_2 & = & f(t_n + h,\, y_n + h\, k_1), \\ \Phi(t_n,\, y_n,\, h) & = & \dfrac{1}{2}(k_1 + k_2), \\ y_{n+1} & = & y_n + h\,\Phi(t_n,\, y_n,\, h). \end{array} \right. \tag{4.16}$$

Exemple 4.10 *On peut proposer une autre méthode à un pas sous une forme du même type :*

$$\left\{ \begin{array}{lcl} k_1 & = & f(t_n,\, y_n), \\ k_2 & = & f(t_n + \dfrac{h}{2},\, y_n + \dfrac{h}{2}\, k_1), \\ \Phi(t_n,\, y_n,\, h) & = & k_2, \\ y_{n+1} & = & y_n + h\,\Phi(t_n,\, y_n,\, h). \end{array} \right. \tag{4.17}$$

Cette méthode est connue sous le nom de méthode d'Euler modifiée ou **méthode du point milieu**. *La fonction qui le définit est* $\Phi(t,y,h) = f(t+\frac{h}{2},\, y+\frac{h}{2}f(t,y))$ *Le calcul de* y_{n+1} *en fonction de* y_n *s'écrit donc également*

$$y_{n+1} = y_n + h\, f(t_n + \frac{h}{2}, y_n + \frac{h}{2}\, f(t_n, y_n)).$$

Ce type de construction peut être généralisé de la façon suivante. On définit un schéma de Runge-Kutta à q étages en se donnant :

- une partition $0 \leq c_1 \leq \ldots \leq c_q \leq 1$ de $[0,1]$,
- pour chaque c_i, une formule d'intégration approchée sur $[0,\, c_i]$:

$$\int_0^{c_i} g(s)\, ds \simeq \sum_{j=1}^{i} a_{i,j}\, g(c_j), \tag{4.18}$$

- une formule d'intégration approchée sur $[0,\, 1]$:

$$\int_0^{1} g(s)\, ds \simeq \sum_{j=1}^{q} b_j\, g(c_j). \tag{4.19}$$

Pour écrire le schéma, on va utiliser les formules (4.19) pour évaluer l'intégrale de l'expression (4.7), avec le changement de variable qui envoie l'intervalle $[t_n,\, t_{n+1}]$ sur $[0,\, 1]$:

$$\int_{t_n}^{t_{n+1}} y'(s)\, ds = h \int_0^1 y'(t_n + s\, h)\, ds \simeq h \sum_{i=1}^{q} b_i\, y'(t_n + c_i\, h).$$

Mais pour chaque i, les $y'(t_n + c_i\, h) = f(t_n + c_i\, h, y(t_n + c_i\, h))$ sont inconnus, et on utilise les formules (4.18) pour fournir des approximations intermédiaires

$$y(t_n + c_i\, h) \simeq y_{n,i} = y_n + h \sum_{j=1}^{q} a_{i,j}\, f(t_n + c_j\, h, y_{n,j}), \quad 1 \leq i \leq q.$$

On généralise la notation (4.16) en introduisant les variables $k_i \simeq f(t_n + c_i\, h, y_{n,i})$. Le schéma s'écrit alors :

$$\left\{ \begin{array}{lcl} k_i & = & f(t_n + c_i\, h,\, y_n + h \sum_{j=1}^{q} a_{i,j}\, k_j), \quad 1 \leq i \leq q, \\ \\ y_{n+1} & = & y_n + h \sum_{j=1}^{q} b_j\, k_j. \end{array} \right. \tag{4.20}$$

On ne va s'intéresser de façon générale qu'aux schémas explicites, et on imposera $c_1 = 0$. Dans ce cas, pour chaque i, les $a_{i,j}$ sont nuls si $j \geq i$: la matrice $A = (a_{i,j})_{1 \leq i,\, j \leq q}$ est alors strictement triangulaire inférieure. Les k_i sont des approximations de $f(t,y)$ aux instants intermédiaires $t_n + c_i\, h$, $1 \leq i \leq q$. On impose en outre une condition qui rend consistantes ces approximations :

$$c_i = \sum_{j=1}^{q} a_{i,j}, \;\; 1 \leq i \leq q. \tag{4.21}$$

Notation synthétique

Pour une présentation synthétique de la méthode, on utilise généralement le diagramme suivant

$$\begin{array}{c|ccccccc} 0 & & & & & & & \\ c_2 & a_{21} & & & & & & \\ c_3 & a_{31} & a_{32} & & & & & \\ \vdots & \vdots & \vdots & \ddots & & & & \\ c_i & a_{i1} & a_{i2} & \dots & a_{ii-1} & & & \\ \vdots & \vdots & \vdots & \dots & \vdots & \ddots & & \\ c_q & a_{q1} & a_{q2} & \dots & a_{qi-1} & \dots & a_{qq-1} & \\ \hline & b_1 & b_2 & \dots & b_{i-1} & \dots & b_{q-1} & b_q \end{array} \tag{4.22}$$

On retrouve par exemple les méthodes (4.17) et (4.16) à travers leurs diagrammes respectifs :

$$\begin{array}{c|cc} 0 & & \\ 1/2 & 1/2 & \\ \hline & 0 & 1 \end{array} \qquad\qquad \begin{array}{c|cc} 0 & & \\ 1 & 1 & \\ \hline & 1/2 & 1/2 \end{array}$$

Exemple 4.11 : La méthode "classique" de Runge-Kutta. *C'est la méthode habituellement désignée par le nom de méthode de Runge-Kutta. C'est une méthode excellente pour la plupart des problèmes de Cauchy, en tous cas souvent la première à essayer. Elle est à 4 étages, définie par la fonction* $\Phi(t, y, h) = \dfrac{1}{6}(k_1 + 2k_2 + 2k_3 + k_4)$ *avec :*

$$\begin{aligned} k_1 &= f(t, y), \\ k_2 &= f(t + \frac{h}{2},\, y + \frac{h}{2}\, k_1), \\ k_3 &= f(t + \frac{h}{2},\, y + \frac{h}{2}\, k_2), \\ k_4 &= f(t + h,\, y + h\, k_3)). \end{aligned}$$

Explicitez la présentation synthétique de cette méthode sous la forme (4.22).

4.1.6 Erreurs locales de discrétisation

Une méthode de la forme (4.13) sera satisfaisante si les valeurs y_n qu'elle permet de calculer se rapprochent des valeurs $y(t_n)$ de la solution de (4.2) aux instants t_n, au moins pour un pas h suffisament petit.

Pour caractériser cette éventuelle "convergence", on introduit les **erreurs locales**

$$e_n = |y(t_n) - y_n|, \tag{4.23}$$

qui sont dominées par l'**erreur globale**

$$E_N = \max_{0 \leq n \leq N} \{e_n\} . \tag{4.24}$$

Comme la solution $y(t)$ est inconnue, ces erreurs e_n ne seront pas accessibles. Nous allons chercher à évaluer la contribution de chaque étape du calcul à l'erreur globale, et réfléchir à sa propagation. Plaçons nous à l'étape n du calcul. Les calculs précédents nous ont fourni une valeur y_n, à laquelle est associée une erreur locale e_n, et nous devons calculer y_{n+1} à laquelle correspondra l'erreur e_{n+1}. Cette erreur e_{n+1} sera donc la somme d'une contribution δ_{n+1} de l'étape courante et des contributions précédentes.

Pour essayer de caractériser la contribution δ_{n+1} de l'étape courante, on considère le problème

$$\left\{ \begin{array}{lcl} z_n'(t) & = & f(t,\, z_n(t)) \; ; \quad t_n \leq t \leq T, \\ z_n(t_n) & = & y_n. \end{array} \right. \tag{4.25}$$

Définition 4.2 *On appelle* **erreur locale de discrétisation** *à l'étape n la quantité*

$$\delta_{n+1} = z_n(t_{n+1}) - y_{n+1} \tag{4.26}$$

Les $z_n(t)$ sont des solutions de la même équation différentielle que $y(t)$, chacune d'entre elles étant celle qui passe par y_n à l'instant t_n. On obtient alors

$$\delta_{n+1} = \left\{ \begin{array}{l} z_n(t_{n+1}) - z_n(t_n) - h\,\Phi(t_n,\, z_n(t_n),\, h), \\ z_n(t_{n+1}) - y_n - h\,\Phi(t_n,\, y_n,\, h). \end{array} \right. \tag{4.27}$$

Exemple 4.12 *Montrez que dans le cas de l'équation différentielle $y'(t) = a\,y(t)$, les $z_n(t)$ sont de la forme*

$$z_n(t) = y_n\, e^{a\,(t-t_n)}. \tag{4.28}$$

La contribution δ_{n+1} ajoutée à l'erreur par l'étape n pourra être amplifiée ou réduite dans les calculs ultérieurs selon que le problème aura tendance à amplifier ou à atténuer les perturbations. Cette tendance pourra varier dans le cas de problèmes non linéaires.

Pour les équations linéaires, cette tendance ne varie pas et caractérise la **stabilité** au sens de la définition 4.1 du problème. Considérons par exemple :

$$\left\{ \begin{array}{lcl} y'(t) & = & y - t^2 + \dfrac{7\,t - 3}{2} \; ; \quad 0 \leq t \leq 1.6, \\ y(0) & = & 0. \end{array} \right.$$

La solution de ce problème est $y(t) = t^2 - \dfrac{3\,t}{2}$.

La figure 4.5 montre 4 étapes de la méthode d'Euler explicite appliquée à ce problème, avec un pas $h = 0.4$. Les valeurs y_n, $1 \leq n \leq 4$ sont pointées par les flèches qui illustrent la trajectoire de la méthode. Cette figure fait également apparaitre les

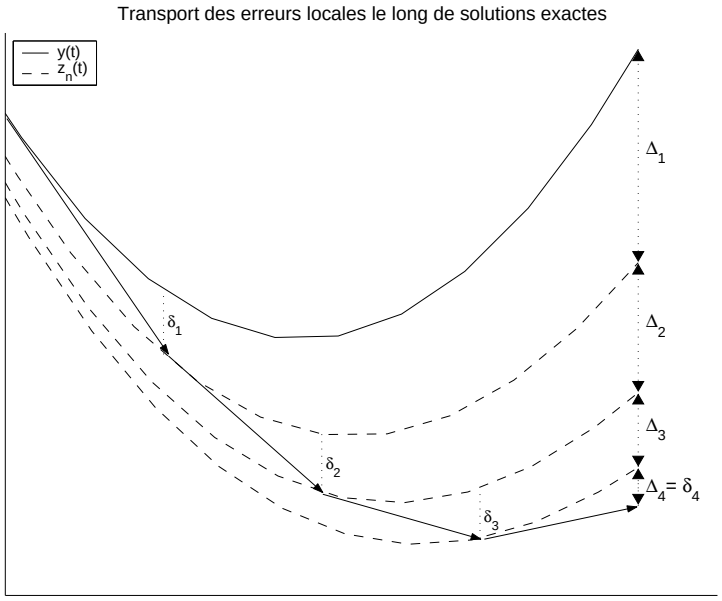


FIGURE 4.5 – Les z_n transportent les erreurs locales de discrétisation

trajectoires des différentes solutions exactes $z_n(t),\ 0 \leq n \leq 3$ aux problèmes (4.25), avec bien sur $z_0(t) = y(t)$.

Les erreurs locales (4.26) ne sont pas simplement additionnées pour contribuer à l'erreur globale : elles sont "transportées" par les solutions z_n pour fournir des contributions Δ_n à l'erreur finale e_4 qui vérifie :

$$e_4 = \sum_{n=1}^{4} \Delta_n.$$

Sur notre exemple, ce "transport" a plutôt enrichi les erreurs locales.

Exemple 4.13 *On applique la méthode d'Euler explicite au problème (4.9). Montrez que les erreurs locales de discrétisation vérifient*

$$\delta_{n+1} \simeq y_n \frac{h^2}{2}.$$

Montrez que si l'on applique la méthode du trapèze (4.15), cette erreur vérifie

$$\delta_{n+1} \simeq y_n \frac{h^3}{6}.$$

4.1.7 Consistance d'un schéma

Définition

Suivant la définition (4.27), l'erreur locale de discrétisation δ_{n+1} est une erreur locale de consistance : elle mesure avec quelle précision le schéma approche localement

une solution de l'équation différentielle. L'exemple 4.13 montre que son comportement dépend du schéma.

Définition 4.3 *On dit que le schéma (4.13) est consistant à l'ordre p s'il existe une constante $K > 0$ telle que*

$$|\delta_n| \leq K\, h^{p+1}. \tag{4.29}$$

Cette condition garantit que pour une méthode à un pas consistante (à l'ordre 1), on aura

$$\lim_{h\to 0} \sum_{n=1}^{N} |\delta_n| = 0.$$

En effet, cette somme fait intervenir $N = \dfrac{T}{h}$ termes, de sorte que $\displaystyle\sum_{n=1}^{N} |\delta_n| \leq K\, T\, h$.

L'ordre d'une méthode n'est réalisé que si le problème que l'on veut résoudre le permet. L'exemple **4.16** montre qu'un défaut de régularité du problème interdit de réaliser la précision espérée. Le théorème suivant donne une condition simple pour assurer qu'une méthode est consistante.

Théorème 4.2 *La méthode (4.13) est consistante si et seulement si, pour tout $t \in [0, T]$ et tout $y \in \mathbb{R}$:*

$$\Phi(t, y, 0) = f(t, y).$$

Exemple 4.14 *On suppose que f et Φ sont suffisament régulières pour autoriser un développement à l'ordre 2 de $z_n(t_{n+1})$. Montrez que, pour un $\eta \in [t_n,\, t_{n+1}]$,*

$$\delta_{n+1} = h\ (f(t_n,\, y_n) - \Phi(t_n,\, y_n,\, h)) + \frac{h^2}{2}\, z_n''(\eta).$$

Utiliser un développement à l'ordre 1 en h de $\Phi(t_n,\, y_n,\, h)$ pour démontrer le théorème 4.2.

Cas des méthodes de type Runge-Kutta

L'étude des schémas de Runge-Kutta est facilitée par leur présentation matricielle. On note A la matrice des paramètres $a_{i,\,j}$ de la méthode introduite dans la défition (4.20). On définit le vecteur $b = [b_1, \ldots, b_q]^T$, la matrice $C \in M_q(\mathbb{R})$, diagonale, dont les éléments diagonaux sont les c_i et le vecteur $1 \in \mathbb{R}^q$ tel que $1 = [1, \ldots, 1]^T$. La condition (4.21) s'écrit alors $A\,1 = C\,1$. Ces éléments permettent d'énoncer le théorème suivant :

Théorème 4.3 *La méthode de Runge-Kutta caractérisée par les éléments A, b, C et 1 définis ci-dessus est :*

- *consistante si et seulement si $b^T\, 1 = 1$,*

- *d'ordre 2 si en outre* $b^T C\,1 = b^T A\,1 = 1/2$,
- *d'ordre 3 si en outre* $\left\{ \begin{array}{rcl} b^T C^2\,1 & = & 1/3, \\ b^T A\,C\,1 & = & 1/6, \quad (= b^T A^2\,1), \end{array} \right.$
- *d'ordre 4 si en outre* $\left\{ \begin{array}{rcl} b^T C^3\,1 & = & 1/4, \\ b^T A\,C^2\,1 & = & 1/12, \\ b^T A^2\,C\,1 & = & 1/24, \quad (= b^T A^3\,1), \\ b^T C\,A\,C\,1 & = & 1/8. \end{array} \right.$

Exemple 4.15 *La méthode classique de l'Exemple 4.11 est d'ordre 4. On la désignera désormais sous le nom de méthode* **RK4**.

Exemple 4.16 *Calculer la solution du problème*

$$\left\{ \begin{array}{l} y' = 1 + |y-1|\,, \quad t > 0, \\ y(0) = 0. \end{array} \right.$$

On calcule la solution approchée aux points $T = 1/2$ et $T = 2$ avec différents pas h, en utilisant la méthode RK4. On note e_N l'erreur aux points T. Le tableau ci-dessous fournit dans chaque cas les valeurs des rapports $\dfrac{e_N}{h^4}$. Commentez les.

T	$h = 1/8$	$h = 1/16$	$h = 1/32$	$h = 1/64$
1/2	$5.61\ 10^{-3}$	$5.32\ 10^{-3}$	$5.18\ 10^{-3}$	$5.12\ 10^{-3}$
2	7.70	38.83	54.93	343.74

Stabilité du problème

Le transport des erreurs locales de discrétisation par les solutions $z_n(t)$ ne les enrichit pas nécessairement, et cela ne dépend pas de la solution $y(t)$.

Exemple 4.17 *Montrez que la fonction $y(t) = t^2 - \dfrac{3\,t}{2}$ est également solution des problèmes :*

$$\left\{ \begin{array}{rcl} y'(t) & = & \dfrac{4}{3}(y - t^2) + 4t - \dfrac{3}{2}\,; \\ y(0) & = & 0. \end{array} \right. \qquad \left\{ \begin{array}{rcl} y'(t) & = & \dfrac{4}{3}(t^2 - y) - \dfrac{3}{2}\,; \\ y(0) & = & 0. \end{array} \right.$$

En appliquant à nouveau la méthode d'Euler implicite avec un pas $h = 0.4$ à ces deux problèmes, on obtient les résultats représentés par la figure 4.6. On constate que dans un cas les erreurs locales de discrétisation sont clairement amplifiées, et dans l'autre clairement atténuées.

Ces deux problèmes sont définis par des équations différentielles linéaires. La dérivée partielle $\partial_y f$ est constante, et c'est elle qui permet de savoir si les erreurs

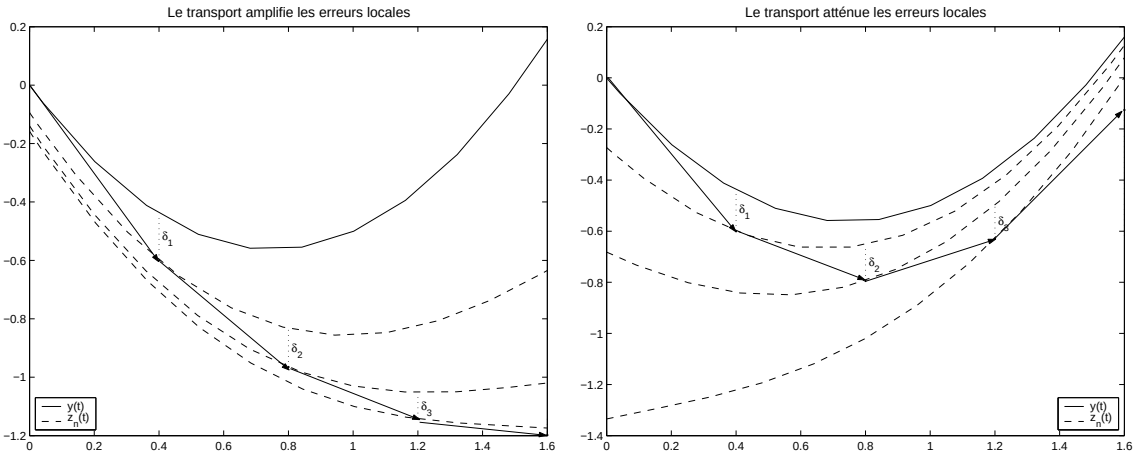


FIGURE 4.6 – Les erreurs locales de discrétisation peuvent être amplifiées ou atténuées

locales seront amplifiées ou atténuées. Dans le cas "stable", elle est égale à $-\frac{4}{3}$, et dans le cas "instable" à $+\frac{4}{3}$.

Dans le cas général, ce comportement va varier au cours du calcul. On peut montrer qui si $\partial_y f(t,\, y) < 0$, alors on aura $|\Delta_n| \leq |\delta_n|$ pour tout n.

Exemple 4.18 *En utilisant (4.28), montrez que pour l'équation différentielle* $y'(t) = a\, y(t)$, *on a*

$$\Delta_{n+1} = \delta_{n+1}\, e^{a\,(T-t_{n+1})}. \tag{4.30}$$

4.1.8 Notion de A-stabilité

L'expression (4.30) montre que pour un problème linéaire stable, les erreurs locales de discrétisation seront transportées de façon "favorables". Comme leurs contributions sur des temps longs seront écrasées, on peut espérer dans ce cas mener des calculs en temps long avec des pas de temps pas trop petits.

Considérons par exemple le problème

$$\left\{ \begin{array}{rcl} y'(t) & = & -a\ (y(t) - \sin(t)) + \cos(t), \quad 0 < t < 2\,\pi, \\ y(0) & = & 1. \end{array} \right. \tag{4.31}$$

La solution générale de l'équation différentielle qui le définit est donnée en fonction du paramètre a par $y(t) = c\, \exp(-a\, t) + \sin(t)$. Si le paramètre a est positif, et par exemple pour $a = 10$, c'est un problème stable. La condition initiale permet de calculer la constante $c = y(0) = 1$.

Exemple 4.19 *Les solutions $z_n(t)$ de (4.25) dans le cas du problème (4.31) sont données par*

$$z_n(t) = (y_n - \sin(t_n))\, e^{a(t_n - t)} + \sin(t).$$

Les erreurs locales de discrétisation pour la méthode d'Euler explicite vérifient donc

$$\delta_{n+1} \simeq \frac{a^2\, h^2}{2}\, (y_n - \sin(t_n)) - \frac{h^2}{2} \sin(t_n). \tag{4.32}$$

On se place dans le cas où $a = 10$. Compte-tenu de (4.32) et (4.30), on peut tenter des calculs avec un pas raisonnable, au moins pour les temps un peu longs. Ainsi, pour un pas $h = 0.01$, de sorte que $a\, h = 0.1$, l'erreur globale E_N est d'environ 0.0211, réalisée en debut de calcul ; l'erreur au point $T = 2\,\pi$ est d'environ 0.0005 ! La figure 4.7 montre la solution calculée avec ce pas, ainsi qu'avec trois autres pas, et fait apparaitre également la solution exacte.

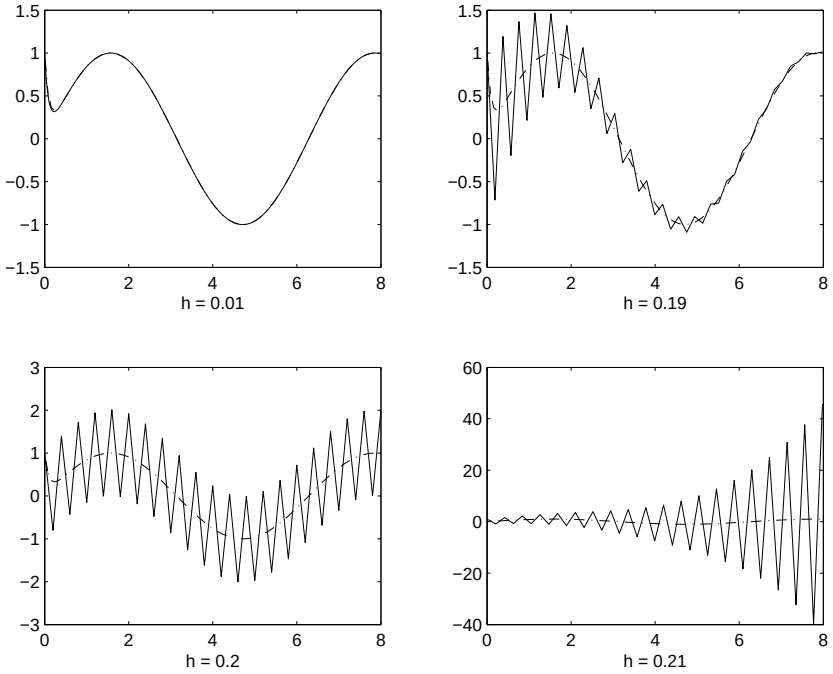


FIGURE 4.7 – Apparition d'instabilités pour certaines valeurs du pas.

Pour le pas $h = 0.19$, la solution calculée commence par osciller autour des valeurs de la solution exacte avant de se stabiliser.

Pour le pas $h = 0.20$, la solution oscille autour de la solution exacte sans se stabiliser, et finalement, pour le pas $h = 0.21$, les oscillations initiales sont constament amplifiées au cours du calcul ! La valeur 0.20 semble bien être un seuil de stabilité pour ce calcul.

On va donc essayer de comprendre ce qui se passe en s'intéressant au calcul d'une solution de l'équation homogène associée au problème (4.31) :

$$y'(t) = -10\, y(t). \tag{4.33}$$

Sa solution générale est $y(t) = y(0)\, e^{-10t}$ et tend vers 0 lorsque t tend vers l'infini. Il faut que le schéma qu'on utilise pour calculer la solution de (4.31) respecte ce comportement.

Après avoir choisi $y_0 = y(0)$, le schéma d'Euler calcule

$$y_1 = y_0 - 10\, h\, y_0 = (1 - 10\, h)\, y_0.$$

et au-delà, pour tout n, $y_n = (1 - 10\, h)^n\, y_0$.

y_n ne tendra vers 0 lorsque n tend vers l'infini qu'à la condition que $|1 - 10\, h| < 1$, et comme le pas h est positif, il devra vérifier $10\, h < 2$, d'ou $h < \dfrac{2}{10} = 0.2$. Cette valeur constitue le **seuil de stabilité** de la méthode d'Euler explicite pour le problème (4.33), et par suite pour le problème (4.31). Cette notion est généralisée par le théorème suivant.

Théorème 4.4 *Lorsqu'on applique une méthode à un pas à l'équation différentielle $y'(t) = -a\, y(t)$, en utilisant le pas h, cette méthode calcule des approximations y_n de $y(tn)$ sous la forme*

$$y_n = R(a\, h)^n\, y_0,\ n > 0. \tag{4.34}$$

La fonction $R(x)$ s'appelle **fonction de stabilité** *de la méthode.*

La méthode est **inconditionellement stable** *si $|R(x)| < 1$ pour tout $x > 0$. Sinon, on appelle* **rayon de stabilité** *de cette méthode le nombre $r > 0$ défini par*

$$r = \max\left\{\rho,\ \rho > 0, \text{ tel que } 0 < x < \rho \Longleftrightarrow |R(x)| < 1\right\}. \tag{4.35}$$

Le seuil de stabilité de la méthode pour un problème donné sera donc $s = \dfrac{r}{a}$. Toute méthode à un pas d'ordre p sera telle que $R(x) = \displaystyle\sum_{k=0}^{p} (-1)^k\, \frac{x^k}{k!} + \mathcal{O}(x^{p+1})$.

Exemple 4.20 *La fonction de stabilité des méthodes (4.16) et (4.17) est $R(x) = 1 - x + \dfrac{x^2}{2}$. Le rayon de stabilité est $r = 2$.*

Exemple 4.21 *Pour la méthode d'Euler implicite, la fonction $R(x)$ est donnée par*

$$R(x) = \frac{1}{1 + x}.$$

Cette fonction est telle que $|R(x)| < 1$ pour tout $x > 0$. Il n'y a donc pas de restriction à la stabilité de cette méthode : elle est inconditionnellement stable.

Remarque 4.1 *Pour des systèmes différentiels, le réel a est remplacé par une matrice $A \in M_n(\mathbb{R})$ symétrique définie positive. La solution du problème $y' = -A\, y$ est donnée par $y(t) = \exp(-t\, A)\, y(0)$, où l'exponentielle est une exponentielle de matrice.*

L'exponentielle de matrice est définie, par analogie avec la fonction exponentielle, par la relation

$$\exp(-t\,A) = \sum_{k=0}^{\infty} \frac{(-t)^k}{(k!)}\,A^k.$$

Lorsque la matrice A est diagonalisable, selon $A = P\,\Lambda\,P^{-1}$, Λ désignant une matrice diagonale dont les éléments diagonaux sont les valeurs propres λ_i de A, et P une matrice dont les colonnes forment une base de vecteurs propres, on obtient :

$$\exp(-t\,A) = P \begin{bmatrix} \exp(-\lambda_1\,t) & & & \\ & \exp(-\lambda_2\,t) & & 0 \\ & & \ldots & \\ 0 & & & \exp(-\lambda_n\,t) \end{bmatrix} P^{-1}.$$

Si A est symétrique définie positive, elle est diagonalisable, et ses valeurs propres sont strictement positives : $y(t)$ tend donc vers 0 lorsque t tend vers l'infini. Le seuil de stabilité est alors $s = \dfrac{r}{\lambda_M(A)}$, où $\lambda_M(A)$ désigne la plus grande valeur propre de A.

4.2 Pour en savoir plus...

4.2.1 Convergence des méthodes à un pas

Au-delà de l'observation empirique de la convergence de certains schémas vers la solution du problème différentiel qu'on cherche à approcher, il n'est pas inutile d'établir formellement cette propriété.

Définition 4.4 *On dit que la méthode (4.13) est* **convergente à l'ordre** *p s'il existe une constante $C > 0$ telle que*

$$E_N \leq C\,h^p. \tag{4.36}$$

La consistance à l'ordre p ne suffit pas à garantir cette convergence. Il faut ajouter une condition de stabilité. A priori, c'est la méthode qui doit être stable, et une méthode à un pas consistante à l'ordre p sera toujours stable si le problème satisfait les hypothèse du théorème 4.1.

Théorème 4.5 *Si la fonction f vérifie les hypothèses du théorème 4.1, la méthode à un pas (4.13) vérifiant la condition de consistance à l'ordre p (4.29) est convergente à l'ordre p. L'erreur globale E_N satisfait l'inégalité*

$$E_N \leq \frac{K}{L}\,h^p\left(e^{L\,T} - 1\right).$$

On a remarqué que les erreurs δ_n sont transportées par les solutions de (4.25). Nous allons essayer de donner une majoration des contributions Δ_n à l'erreur finale e_N.

A un instant $t > t_{n+1}$, Δ_{n+1} s'écrit comme une fonction de t :

$$\Delta_{n+1}(t) = z_n(t) - z_{n+1}(t),$$

$$|\Delta_{n+1}(t)| = \left| z_n(t_{n+1}) - z_{n+1}(t_{n+1}) + \int_{t_{n+1}}^{t} \left(z_n'(s) - z_{n+1}'(s) \right) \, ds \right|.$$

De la définition (4.26) et de (4.25), on tire

$$|\Delta_{n+1}(t)| \leq |\delta_{n+1}| + \int_{t_{n+1}}^{t} |f(s,\, z_n(s)) - f(s,\, z_{n+1}(s))| \, ds.$$

En supposant que f satisfait les hypothèses du théoreme 4.1, on obtient

$$|\Delta_{n+1}(t)| \leq |\delta_{n+1}| + L \int_{t_{n+1}}^{t} |z_n(s) - z_{n+1}(s)| \, ds. \tag{4.37}$$

On pose alors $v_n(t) = \displaystyle\int_{t_{n+1}}^{t} |z_n(s) - z_{n+1}(s)| \, ds$. La fonction v_n est telle que $v_n(t_{n+1}) = 0$, et vérifie :

$$v_n'(t) = |\Delta_{n+1}(t)| \leq |\delta_{n+1}| + L\, v_n(t). \tag{4.38}$$

(4.38) s'écrit également $v_n'(t) - L\, v_n(t) \leq |\delta_{n+1}|$ qui fournit, en multipliant par e^{-Lt} :

$$(v_n'(t) - L\, v_n(t))\; e^{-Lt} = \frac{d}{dt} \left(v_n(t)\, e^{-Lt} \right) \leq |\delta_{n+1}|\, e^{-Lt}.$$

En intégrant de t_{n+1} à t, et comme $v_n(t_{n+1}) = 0$:

$$v_n(t)\, e^{-Lt} \leq \frac{|\delta_{n+1}|}{L} \left(e^{-Lt_{n+1}} - e^{-Lt} \right).$$

$$v_n(t) \leq \frac{|\delta_{n+1}|}{L} \left(e^{L(t-t_{n+1})} - 1 \right).$$

L'inégalité (4.37) fournit finalement :

$$|\Delta_{n+1}(t)| \leq |\delta_{n+1}| + L\, \frac{|\delta_{n+1}|}{L} \left(e^{L(t-t_{n+1})} - 1 \right) = |\delta_{n+1}|\, e^{L(t-t_{n+1})}. \tag{4.39}$$

La relation (4.39) va nous permettre de majorer l'erreur $e_N = \displaystyle\sum_{n=1}^{N} \Delta_n(T)$ pour une méthode d'ordre p, avec $T = t_N = N\, h$. Dans ce cas, suivant (4.29),

$$|e_N| \leq K\, h^{p+1} \sum_{n=1}^{N} e^{L(T-t_n)} = K\, h^p\, e^{LT} \sum_{n=1}^{N} h\, e^{-L\, t_n}.$$

En remarquant que la somme de droite est la formule composée des rectangles à droite pour l'intégrale $\int_0^T e^{-L\,s}\,ds = \dfrac{1-e^{-L\,T}}{L}$, et que cette intégrale domine la somme, on obtient

$$|e_N| \leq \frac{K}{L}\,h^p\left(e^{L\,T}-1\right).$$

Remarque 4.2 *Dans la définition du schéma, on suppose que y_0 est donné exactement. On verra ci-dessous que dans le cas d'erreurs d'arrondi, la convergence n'est plus garantie en cas d'erreur initiale..*

4.2.2 Calculs en arithmétique finie

On va supposer que les calculs sont perturbés par des erreurs d'arrondi, ce qui va introduire un terme supplémentaire aux erreurs locales de discrétisation. On aura à présent

$$|\delta_{n+1}| \leq K\,h^{p+1} + |\epsilon_{n+1}|.$$

On aura également une erreur initiale ϵ_0. Cette erreur correspond à l'écriture de la condition initiale en arithmétique finie : c'est par exemple l'écart entre $\dfrac{1}{3}$ et sa valeur approchée par l'arithmétique de l'ordinateur pour l'exemple (4.10). Cette erreur sera transportée par la solution du problème perturbé, pour fournir une contribution $\Delta_0 = \epsilon_0\,e^{L\,T}$ à l'erreur finale, selon le principe de la démonstration précédente ou de l'analyse de l'exemple (4.10).

En supposant que les erreurs d'arrondi des étapes 1 à N sont majorées en valeur absolue par σ, on obtiendra alors une majoration de la forme

$$|e_N| \leq |\epsilon_0|\,e^{L\,T} + (K\,h^p + \frac{\sigma}{h})\,e^{L\,T}\sum_{n=1}^{N} h\,e^{-L\,t_{n+1}} \leq e^{L\,T}\left(|\epsilon_0| + C\,h^p + \frac{\sigma}{h}\right). \quad (4.40)$$

On remarque un terme en $\dfrac{1}{h}$ qui tend vers l'infini lorsque h tend vers 0.

Exemple 4.22 *On applique la méthode du point-milieu (4.17) au problème (4.9) On réalise alors les calculs en utilisant* MATLAB*, c'est-à-dire en arithmétique IEEE pour laquelle l'unité d'arrondi est $u \simeq 1.11\,10^{-16}$. On convient de choisir $\epsilon_0 = \sigma = u$. La fonction qui donne la majoration empirique de l'erreur résultant de (4.40) est alors, avec ici $L = T = 1$:*

$$e(h) = e * \left(u + \frac{e}{6}\,h^2 + \frac{u}{h}\right).$$

La figure 4.8 représente les erreurs observées pour 200 valeurs du pas h régulièrement réparties entre 10^{-7} et $2\,10^{-5}$: elle fait clairement apparaitre que le comportement des erreurs observées est globalement bien représenté par la courbe empirique.

On constate aussi que l'influence des erreurs d'arrondi est négligeable : il est inutile de choisir un pas trop petit et les risques d'erreurs proviennent surtout de l'éventuelle instabilité du problème.

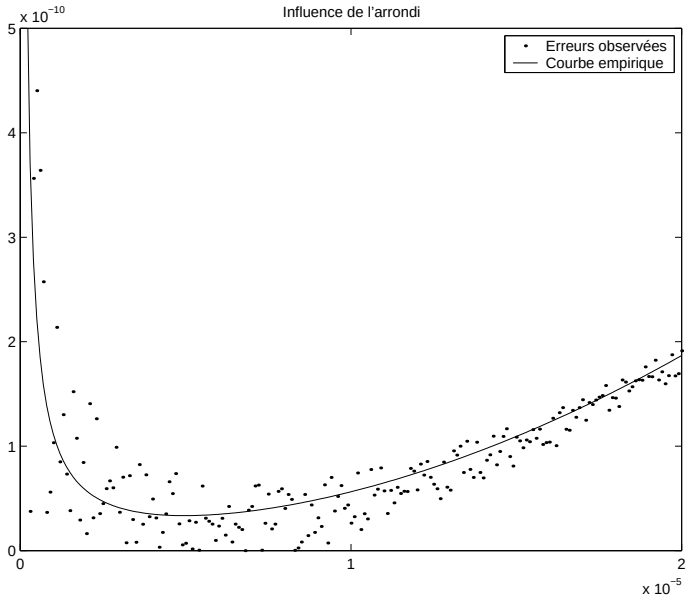


FIGURE 4.8 – Estimation et observation des erreurs d'arrondi

4.2.3 Problèmes raides

Qu'est-ce qu'un problème raide ?

Reprenons l'exemple (4.31). Sa solution est une combinaison linéaire de deux fonctions :

- $y_R(t) = \exp(-a\,t)$ est la solution générale de l'équation homogène ; lorsque le paramètre a devient grand, elle décrit un phénomène rapide ; son effet sur la solution a une durée de vie de l'ordre de $\dfrac{1}{a}$.
- $y_L(t) = \sin(t)$ qui dépend ici directement du second membre décrit un phenomène plus lent.

La figure 4.9 ilustre ce comportement ; on y constate que la contribution de la fonction y_R à la solution devient vite négligeable.

Cet exemple est typique des problèmes de phase transitoire, lorsqu'une contribution telle que y_R n'existe que sur un temps très petit. Pour ces temps petits, la formule (4.32) montre que, si l'on utilise une méthode d'Euler, on devra imposer une contrainte de précison de la forme $(a^2+1)\,h^2 \leq 2\,\epsilon$ pour garantir localement une précison de l'ordre de ϵ, et on pourrait aussi le vérifier pour la méthode implicite. Pour $t >> \dfrac{1}{a}$, cette contrainte n'est plus valide. C'est alors la contrainte de stabilité $a\,h \leq 2$ qui va prendre le relai, sauf si l'on utilise une méthode implicite.

En fait, aucune méthode explicite ne peut être A-stable, alors que les méthodes implicites peuvent l'être. C'est ce qui les rend indispensables pour les problèmes raides.

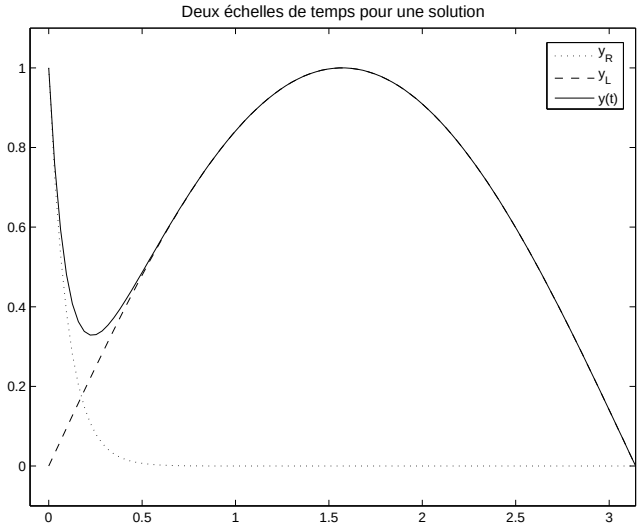


FIGURE 4.9 – Le problème (4.31) est raide si a est grand.

Un exemple de problème raide

La flamme de l'allumette : Voici un autre exemple de problème raide. Il est dû à L. Shampine. Il s'agit d'un modèle simplifié de propagation de flamme : lorsque vous grattez une allumette, la boule de flamme va grossir très rapidement avant d'atteindre une taille critique, pour laquelle la quantité d'oxygène consommée à l'intérieur de cette boule est égale à celle qui est disponible en surface. La surface d'une sphère de rayon y est $4\,\pi\,y^2$, et son volume est $\dfrac{4}{3}\,\pi\,y^3$. Cette loi d'équilibre est grossièrement décrite par le problème :

$$\begin{cases} y' = y^2 - y^3\,;\, 0 \leq t \leq \dfrac{2}{\rho} \\ y(0) = \rho. \end{cases} \tag{4.41}$$

Ce problème semble très anodin, et il l'est en effet lorsque le rayon initial ρ n'est pas trop petit. Mais si ρ est petit, il devient assez raide. La figure 4.12 nous montre les courbes calculées par MATLAB pour 2 valeurs du rayon.

Les solveurs de MATLAB La figure 4.12 semble indiquer que MATLAB a bien réussit à approcher la solution du problème raide. L'appel d'un solveur MATLAB pour obtenir la courbe de droite peut s'écrire :

```
>> fonction = inline('y.^2-y.^3','t','y');
>> delta = 0.0001;
>> intervalle =  [0, 2/delta];
>> [t, y] = ode-solveur(fonction, intervalle, delta);
>> plot(t, y);
```

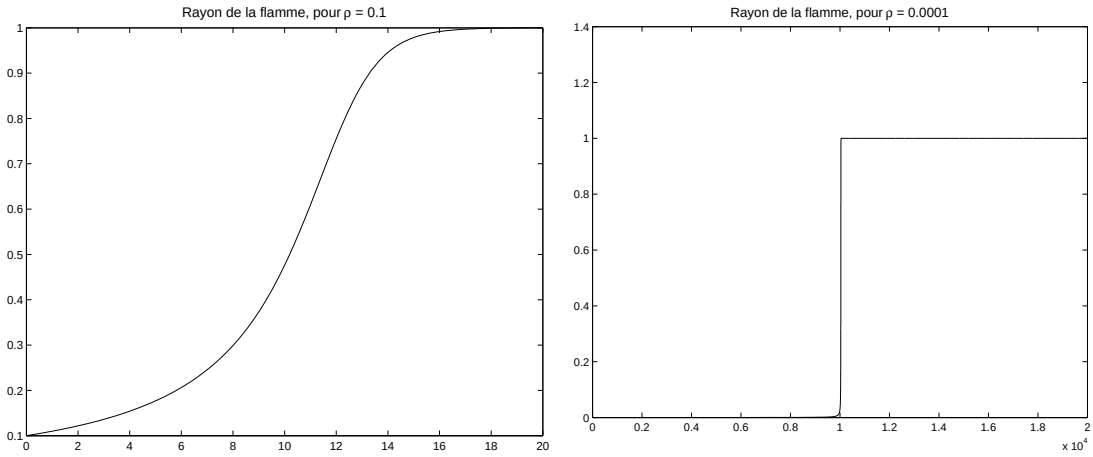


FIGURE 4.10 – Le problème se raidit quand ρ devient petit.

Dans ces lignes, `ode-solveur` sera remplacé successivement par `ode113`, `ode23`, `ode45` et `ode23s`.

En fait, il convient de regarder les choses d'un peu plus près. Tous ces solveurs ne se comportent pas de la même façon, et la figure 4.11 nous montre les courbes obtenues avec une échelle définie par la commande

```
>> axis([0.98/delta 1.12/delta 1-30*delta 1+30*delta]);
```

On constate alors, que parmi les solveurs testés, seul `ode23s` semble se comporter correctement.

Les courbes associées à `ode23` et `ode45` présentent une zone très chaotique, mais les valeurs calculées respectent la tolérance relative par défaut qui est de 10^{-3}. Ces solveurs ont donc bien fonctionné. Ils ne sont simplement pas adaptés au problème.

`ode113` ne réussit pas à respecter cette tolérance. C'est normal car les méthodes multipas linéaires ne sont pas adaptées aux problèmes raides. Elles conviennent à des problèmes assez réguliers pour lesquels le coût d'évaluation de f est important. Elles sont aussi mieux adaptées que les méthodes à un pas pour les calculs de trajectoires à long terme.

Nous pouvons obtenir des informations sur le déroulement des calculs en déclarant :

```
>> options = odeset('Stats','on');
>> [t, y] = ode-solveur(fonction, intervalle, delta, options);
```

Voici deux des réponses obtenues :

- pour `ode45` :

```
3028 successful steps
762 failed attempts
22741 function evaluations
0 partial derivatives
0 LU decompositions
0 solutions of linear systems
```

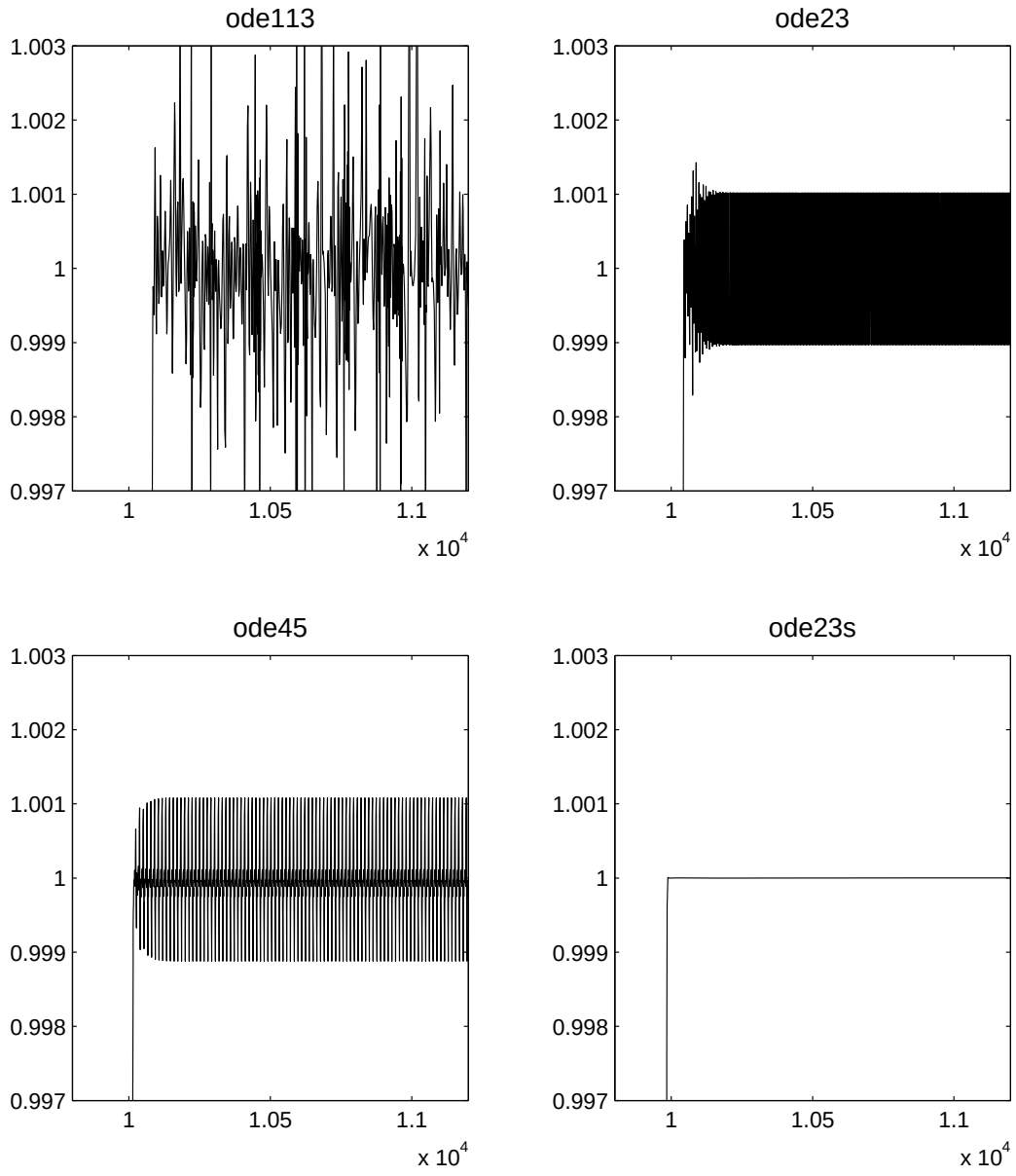


FIGURE 4.11 – Quelques solveurs de MATLAB pour un problème raide.

- pour `ode23s` :

```
55 successful steps
14 failed attempts
250 function evaluations
55 partial derivatives
69 LU decompositions
207 solutions of linear systems
```

La statistique relative à `ode23s` mentionne des évaluations de dérivées partielles, qui sont faites par différences finies. Par contre, s'agissant ici d'une équation et non d'un système, les résolutions de systèmes linéaires sont de simples divisions. On constate qu'il faut environ 7 évaluations de fonctions par étape pour `ode45` qui est une méthode à 7 étages.

Le tableau suivant nous donne rappelle le nombre d'évaluations de la fonction f, nous donne la taille du vecteur `t` et le temps CPU (en secondes) utilisé par l'ordinateur sur lequel nous avons effectué le calcul.

Solveur	`ode113`	`ode23`	`ode45`	`ode23s`
Evaluations	19906	12070	22741	250
CPU	8.7667	1.5833	2.8167	0.0667
Taille	9248	4012	12113	56

4.3 Pour en savoir encore plus !

4.3.1 Contrôle du pas pour les méthodes explicites

La figure 4.11 montre les hésitations de la méthode `ode45` sur un problème raide. Ces hésitations s'expliquent par sa stratégie de contrôle du pas.

Estimation de l'erreur

En général, les solutions peuvent varier lentement à certains endroits, et plus rapidement à d'autres. Un pas constant n'est donc pas toujours adapté. Il est souhaitable de pouvoir décider à chaque instant d'un pas qui permet de respecter une tolérance fixée à priori. C'est le comportement local de la solution qui permettra de décider, et on va se servir des erreurs locales de discrétisation (4.26).

Ces erreurs font *a priori* intervenir une solution exacte du problème, qui est inaccessible. Supposons cependant que l'on dispose de deux méthodes, une d'ordre p_1 et l'autre d'ordre p_2, avec $p_1 > p_2$. Partant de la valeur y_n calculée à l'étape n, elle fournissent suivant (4.26) et (4.29)

$$\begin{aligned} y_{n+1}^{(1)} = z_n(t_{n+1}) + \delta_{n+1}^{(1)} &\simeq z_n(t_{n+1}) + C_1\, h^{p_1+1} \\ y_{n+1}^{(2)} = z_n(t_{n+1}) + \delta_{n+1}^{(2)} &\simeq z_n(t_{n+1}) + C_2\, h^{p_2+1} \end{aligned}$$

On déduit $|y_{n+1}^{(2)} - y_{n+1}^{(1)}| \simeq |C_2\, h^{p_2+1} - C_1\, h^{p_1+1}| \sim |C_2\, h^{p_2+1}|$ au voisinage de $h = 0$.

On se fixe une tolérance tol, et on souhaite que $\epsilon = |y_{n+1}^{(2)} - y_{n+1}^{(1)}|$ soit aussi voisin que possible de cette tolérance, sans la dépasser. On choisira par exemple de calculer un pas nouveau h_{nouv} de façon que $tol \simeq C_2\, h_{nouv}^{p_2+1}$.

On a donc $\dfrac{tol}{\epsilon} = \dfrac{C_2\, h_{nouv}^{p_2+1}}{C_2\, h^{p_2+1}}$, de sorte que $\dfrac{h_{nouv}}{h} = \left(\dfrac{tol}{\epsilon}\right)^{\frac{1}{p_2+1}}$.

On choisit alors, avec une petite marge de sécurité, le nouveau pas :

$$h_{nouv} = 0.9\, h \left(\frac{tol}{\epsilon}\right)^{\frac{1}{p_2+1}} \tag{4.42}$$

Le seul problème est que l'on doit mettre en oeuvre 2 méthodes pour mener à bien ces calculs.

Méthodes emboitées

L'idée des méthodes de Runge-Kutta emboitées est d'utiliser les mêmes calculs pour les deux méthodes. Pour illustrer ce principe, observons les diagrammes la méthode d'Euler améliorée et de la méthode de Runge-Kutta d'ordre 4 :

0		
1/2	1/2	
	0	1

0				
1/2	1/2			
1/2	0	1/2		
1	0	0	1	
	1/6	1/3	1/3	1/6

Ces deux diagrammes peuvent se rassembler en un seul :

0				
1/2	1/2			
1/2	0	1/2		
1	0	0	1	
$y^{(1)}$	1/6	1/3	1/3	1/6
$y(2)$	0	1	0	0

La mise en oeuvre de la méthode d'ordre $p_1 = 4$ conduit à calculer, à l'étape n, pour le pas courant h :

$$\begin{aligned}
k_1 &= f(t_n, y), \\
k_2 &= f(t_n + \frac{h}{2},\, y_n + \frac{h}{2}\, k_1), \\
k_3 &= f(t_n + \frac{h}{2},\, y_n + \frac{h}{2}\, k_2), \\
k_4 &= f(t_n + h,\, y_n + h\, k_3)),
\end{aligned}$$

avant de poser $y_{n+1}^{(1)} = y_n + \dfrac{h}{6}(k_1 + 2k_2 + 2k_3 + k_4)$. Ces mêmes calculs peuvent être utilisés pour calculer $y_{n+1}^{(2)} = y_n + h\,k_2$. Cette fois, la méthode est d'ordre $p_2 = 2$, et suivant (4.42), le nouveau pas sera donné par :

$$h_{nouv} = 0.9\,h\,\left(\frac{tol}{\epsilon}\right)^{\frac{1}{3}}$$

Remarque 4.3 *En fait, on travaille plutôt avec des erreurs relatives,*

$$\epsilon = \frac{|y_{n+1}^{(2)} - y_{n+1}^{(1)}|}{\max\{|y_{n+1}^{(2)}|,\, |y_{n+1}^{(1)}|\}}.$$

A l'étape n, on commence par comparer ϵ à tol, et le calcul n'est conservé que lorsque $\epsilon < tol$. La valeur choisie pour la suite sera $y_{n+1} = y_{n+1}^{(1)}$.

La paire de Bogacki et Shampine

Les fonctions `ode23` et `ode45` de MATLAB emboîtent respectivement des méthodes d'ordres (2, 3) et (4, 5). `ode45` est à 7 étages, connue dans la littérature sous le nom de méthode de Dormand-Prince d'ordres (4, 5).

`ode23` est caractérisée par le diagramme suivant :

0				
1/2	1/2			
3/4	0	3/4		
1	2/9	1/3	4/9	
$y^{(1)}$	2/9	1/3	4/9	0
$y(2)$	7/24	6/24	8/24	3/24

Elle est connue sous le nom de méthode de Bogacki et Shampine. On vérifiera en exercice qu'elle est bien d'ordres (2, 3). C'est la méthode 1 qui est d'ordre 3, de sorte que l'on ne calcule pas $y_{n+1}^{(2)}$.

Avec $y_{n+1}^{(1)} = y_n + \dfrac{h}{9}(2\,k_1 + 3\,k_2 + 4\,k_3)$, on doit calculer $k_4 = f(t_{n+1},\, y_{n+1}^{(1)})$ pour estimer

$$y_{n+1}^{(2)} = y_n + \frac{h}{24}\left(7\,k_1 + 6\,k_2 + 8\,k_3 + 3k_4\right),$$

puis faire la différence $\epsilon = y_{n+1}^{(1)} - y_{n+1}^{(2)}$. On préfère calculer directement

$$\epsilon = \frac{h}{72}\left(-5\,k_1 + 6\,k_2 + 8\,k_3 - 9\,k_4\right).$$

4.3.2 Méthodes implicites

Les méthodes implicites nécessitent à chaque pas la résolution d'une équation (ou d'un système) non linéaire. Cette équation est définie par :

$$y_{n+1} = y_n + h\,\Phi(t_{n+1},\ y_{n+1},\ h). \tag{4.43}$$

Stratégie de prédiction-correction (PECE)

On considère la méthode d'Euler implicite. Il faut résoudre l'équation $x = y_n + h\,f(t_{n+1},\ x)$, où x désigne l'inconnue y_{n+1}.

On peut imaginer une méthode de point fixe définie par la fonction $g(x) = y_n + h\,f(t_{n+1},\ x)$. Si f satisfait les hypothèses du théorème 4.1, on aura

$$|g(y) - g(z)| = |h\,f(t_{n+1},\ y) - h\,f(t_{n+1},\ z)| \leq h\,L\,|y - z|.$$

Choisissons alors $x^{(0)}$, si possible voisin de la solution, et calculons

$$x^{(1)} = y_n + h\,f(t_{n+1},\ x^{(0)}) = g(x^{(0)}) \tag{4.44}$$

Comme $g(x) = x$, on aura

$$|x^{(1)} - x| \leq h\,L\,|x^{(0)} - x|. \tag{4.45}$$

Il faudra choisir un pas h tel que $h < \dfrac{1}{L}$ pour que la fonction g soit une application contractante et donc que $x^{(1)}$ soit meilleur que $x^{(0)}$. Cette contrainte impose des conditions pour la convergence qui sont du même ordre que les containtes de stabilité de la méthode explicite !

On préfère travailler uniquement avec $x^{(0)}$ et $x^{(1)}$ en mettant en place une stratégie de contrôle du pas. La première étape est un bon choix de $x^{(0)}$: on le calcule en utilisant la méthode d'Euler explicite, et on calcule $x^{(1)}$ suivant (4.44). Cette stratégie est connue sous le nom de **stratégie de prédiction-correction**.

En supposant y_n calculé, et en notant $f_n = f(t_n, y_n)$, le calcul de y_{n+1} se fait selon :

Prédiction	$y_{n+1}^{(P)} = y_n + h\,f_n$
Evaluation	$f_{n+1}^{(P)} = f(t_{n+1},\ y_{n+1}^{(P)})$
Correction	$y_{n+1}^{(C)} = y_n + h\,f_{n+1}^{(P)}$
Evaluation	$f_{n+1} = f(t_{n+1},\ y_{n+1}^{(C)})$

Remarque 4.4 *Cette stratégie est également utilisée dans le cadre de méthodes multipas. Les plus utilisées sont les méthodes d' Adams :*

- *les méthodes d'Adams-Basforth sont explicites,*
- *les méthodes d'Adams-Moulton sont implictes.*

On peut donc les associer dans une stratégie PECE avec un contrôle du pas. La fonction `ode113` *de* MATLAB *implante cette méthode, pour des ordres pouvant varier de* 1 *à* 13.

Contrôle du pas des méthodes PECE

Pour décider de garder $x^{(1)}$, il faut s'assurer que la majoration (4.45) est acceptable. On le fait de façon empirique en évaluant l'écart

$$\Delta = |y_{n+1}^{(P)} - y_{n+1}^{(C)}|$$

Pour tirer profit de cette évaluation, il faut un peu plus d'informations sur l'erreur locale de discrétisation δ_n (4.26). Pour les méthodes d'Euler, on a

- dans le cas explicite, $z_n(t_{n+1}) = y_n + h\, f(t_n,\, y_n)) + \dfrac{h^2}{2} z_n''(t_{n,1})$,
- dans le cas implicite, $z_n(t_{n+1}) = y_n + h\, f(t_n,\, z_n(t_{n+1})) - \dfrac{h^2}{2} z_n''(t_{n,2})$.

$t_{n,1}$ et $t_{n,2}$ sont tous deux dans l'intervalle $[t_n,\ t_{n+1}]$. On peut donc espérer que

$$\begin{aligned} z_n(t_{n+1})) &\simeq y_{n+1}^{(P)} + \frac{h^2}{2} y''(t_{n,1}), \\ z_n(t_{n+1})) &\simeq y_{n+1}^{(C)} - \frac{h^2}{2} y''(t_{n,2}). \end{aligned}$$

Par soustraction, on obtient $\Delta \simeq Kh^2$, de sorte que

$$|y_{n+1}^{(C)} - z_n(t_{n+1})| \simeq \frac{\Delta}{2}.$$

Pour une tolérance *tol* choisie, si $\dfrac{\Delta}{2} < tol$, on accepte la valeur $y_{n+1} = \dfrac{1}{2}\,(y_{n+1}^{(P)} + y_{n+1}^{(C)})$. Dans tous les cas, on choisit alors un nouveau pas h_{nouv} selon, par exemple,

$$h_{nouv} = 0.9\, h \sqrt{\frac{2\, tol}{\Delta}}.$$

4.3.3 Application à un problème raide

On rappelle que l'équation de Van der Pol (4.12) définit un problème raide lorsque μ devient grand.

La figure 4.12 montre quelques tentatives de calcul de la solution du problème défini par cette équation pour $\mu = 10$, en choisissant comme conditions initiales $y(0) = 2$ et $y'(0) = 0$.

En fait ce problème n'est pas encore très raide, mais il permet de comparer les comportements de la méthode d'Euler implicite avec contrôle du pas, et de la méthode d'Euler explicite.

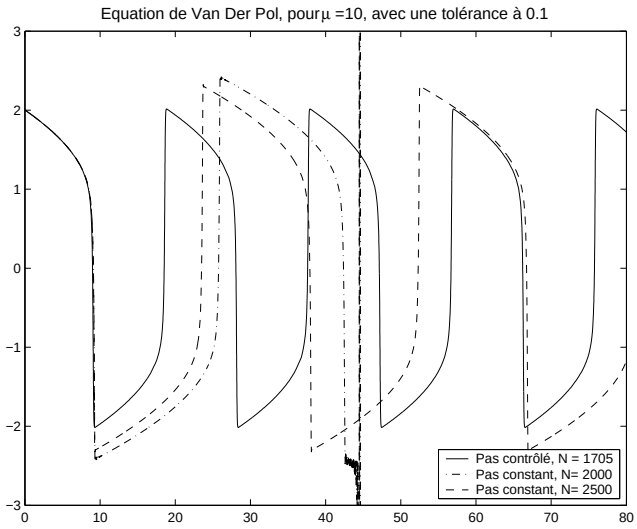


FIGURE 4.12 – La méthode implicite permet de calculer la solution d'un problème raide.

Sur l'intervalle [0, 80], pour une tolérance $tol = 0.1$, la méthode implicite calcule la solution en respectant son comportement périodique : on peut le constater en observant les cycles parcouru par les points $(y(t), y'(t))$.

Pour cette tolérance, le nombre de pas pour le calcul est de $N \simeq 1700$. Avec un pas constant défini par $h = \dfrac{T}{N} \simeq 0.0471$, la méthode explicite explose assez vite.

On peut alors essayer des pas plus importants :

- pour $h = 0.04 = \dfrac{T}{2000}$, la méthode explose aux environs de $t = 45$,
- pour $h = \dfrac{T}{2500} = 0.032$, elle n'explose pas sur l'intervalle considéré ; cependant, elle ne respecte pas le comportement périodique attendu : les "périodes" sont plus longues que celle du calcul correct.

La figure 4.13 nous montre comment la méthode explicite commence à être destabilisée lorsqu'elle rencontre le premier changement brutal de la dérivée $y'(t)$.

Elle commence alors à surévaluer la trajectoire de $y(t)$, et dépasse le nouveau changement très brutal de $y'(t)$. Lorsque la méthode commence à prendre en compte ce changement, elle est légèrement destabilisée, mais les oscillations s'amplifient de façon irrémédiable ! Sur la partie gauche de la figure, on peut observer que la méthode implicite calcule beaucoup moins de valeurs sur l'intervalle [7.5, 8.5] que la méthode explicite. Par contre dès que les variations sont importantes, elle raffine le pas de façon à pouvoir mieux prendre en compte les changements de variation de la fonction $y(t)$.

Remarque 4.5 *On peut aussi chercher à résoudre l'équation (4.43) en utilisant la méthode de Newton, ou une de ses variantes. C'est ce que font les méthodes connues sous le nom de méthodes de Rosenbrock qui sont construites à partir de méthodes de*

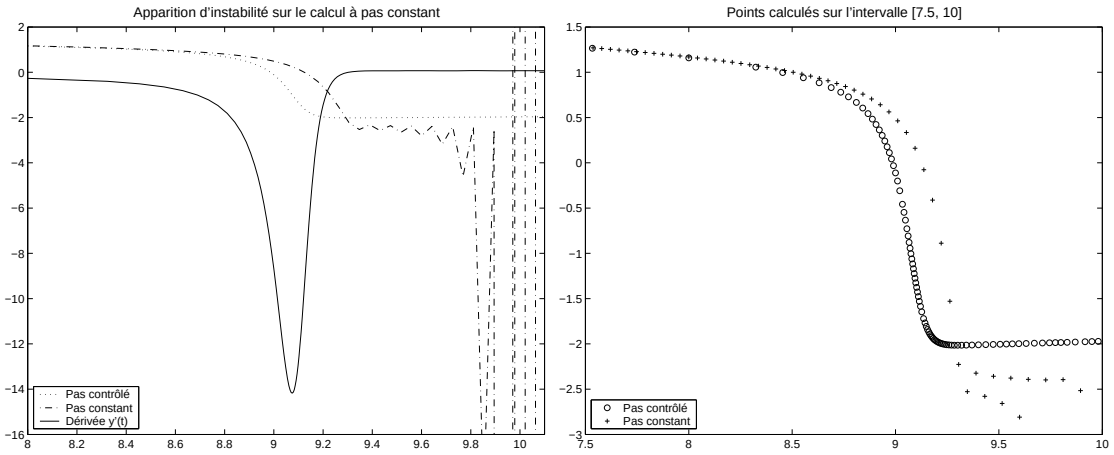


FIGURE 4.13 – Apparition d'instabilités pour la méthode explicite.

type Runge-Kutta semi-implicites. La fonction `ode23s` *de* MATLAB *met en œuvre deux méthodes de Rosenbrock d'ordre 2 et 3, emboitées sur 3 étages. Elle est beaucoup plus efficace que la méthode que nous avons décrite, mais un peu plus difficile à analyser. Comme pour les méthodes de Runge-Kutta emboitées, sa mise en œuvre n'est pas très compliquée.*