COURS D'ANALYSE NUMÉRIQUE
M43

UNIVERSITÉ DU SUD TOULON VAR

ANNÉE 2007-2008

Cédric Galusinski

Introduction.

Ce polycopié est consacré à l'introduction des outils de l'analyse numérique. Le calcul d'intégrales, la résolution d'équations non linéaires, la résolution d'équations différentielles n'ont pas toujours (pas souvent) de solution analytique ou de solution analytique simple. On a alors recours à la recherche de solutions approchées.
Les physiciens, les mécaniciens, les ingénieurs, ... ont besoin d'évaluer les solutions de problèmes tels qu'évoqués précédemment. L'important n'est pas d'avoir une solution exacte, mais une solution approchée avec une erreur maîtrisée et quantifiée.
La modélisation de phénomènes riches et complexes amène un grand nombre de variables réelles inconnues à déterminer à l'aide d'outils d'analyse numérique. Un algorithme fait alors appel à un grand nombre d'opérations élémentaires, le mathématicien numéricien doit alors s'assurer que son calcul conduit bien au résultat souhaité et que son calcul est également efficace en terme de temps de calcul.

DÉFINITION .0.1
On appelle méthode numérique une suite finie d'opérations arithmétiques $(+, -, \star, /)$ *permettant d'avoir la solution d'un problème avec une précision arbitrairement fixée.*

On saura souvent prouver qu'une méthode assure la convergence "théorique" vers le résultat souhaité. Un processus itératif est associé à la méthode ; pour un nombre donné d'itérations, on obtient une solution approchée dont la précision dépend du nombre d'itérations et donc du nombre d'opérations effectuées. Si on s'autorise à faire tendre le nombre d'itérations vers l'infini, on s'attend à faire tendre l'erreur (la précision de la méhode) vers zéro. La méthode est dite convergente si la solution approchée converge vers la solution exacte, lorsque le nombre d'itération tend vers l'infini. Mais les opérations arithmétiques ne se faisant pas de façon exacte (erreur machine), on est amené à cumuler des erreurs. Un cumul important d'erreurs même petites peut s'avérer désastreux et conduire à un résultat erroné voire divergeant (le programme va planter sur la machine !), alors que la méthode semblait convergente sur le papier. On distinguera alors la notion de "convergence théorique" de la notion de "convergence effective". On parle aussi de robustesse ou stabilité de la méthode lorsque celle-ci n'est pas sensible au cumul des erreurs. C'est la notion la plus fine de l'analyse numérique sur laquelle nous insisterons à travers les diverses méthodes introduites dans ce cours.

Notion d'erreur machine

Les nombres réels sur lesquels nous effectuons des opérations sont stockés sur la machine à l'aide d'un nombre donné d'octets, soit un nombre fini de combinaisons possibles. On représente ainsi les réels sur une machine à l'aide d'un nombre fini de nombres alors même qu'il existe un nombre infini de réels, y compris sur un intervalle borné de$\mathbb{R}$.
Les nombres réels stockés sur 4 octets sont dits réels simple précision. Les nombres réels stockés sur 8 octets sont dits réels double précision. En analyse numérique, on utilisera les réels double précision, ils permettent de représenter les réels avec suffisamment de précision et sur une échelle de taille suffisamment large pour les problèmes qui se posent en analyse numérique, et compte tenu de la performance actuelle des ordinateurs. Les réels quadruple précision (16 octets) ne sont en général pas utilisés du fait d'un stockage mémoire plus important (on peut être amené à stocker plusieurs millions de réels lors

d'un calcul) et surtout d'un temps de calcul majoré pour des opérations arithmétiques rendant un résultat consistant avec l'erreur attendue.

DÉFINITION .0.2
On appelle erreur machine l'erreur maximale commise pour représenter un réel. L'erreur machine relative est l'erreur machine commise sur un réel rapportée à ce réel : si x_ε est le nombre stocké sur la machine pour représenter un nombre réel x, l'erreur relative est

$$\frac{|x - x_\varepsilon|}{|x|}.$$

REMARQUE .0.1 *Les opérations élémentaires (arithmétiques) ainsi que les quelques fonctions de base d'un language effectuent les calculs avec une précision suffisante pour ne pas introduire d'erreurs supplémentaires.*

En simple précision, l'erreur machine relative est de l'ordre de $\varepsilon \sim 10^{-7}$. En double précision, l'erreur machine relative est de l'ordre de $\varepsilon \sim 10^{-15}$.
Exemple de perte de précision : on note ε l'erreur machine relative. Un nombre de l'ordre de 10 est manipulée avec une erreur machine de l'ordre de 10ε. Ainsi l'opération $10 - 9$ est manipulée en réel avec une erreur cumulée de taille (situation au pire) $10\varepsilon + 9\varepsilon$. A l'issu de ce calcul, on récupère 1 à une erreur près d' un ordre 10 fois plus mauvais que l'erreur machine attendue pour stocker le réel 1.
Une telle perte de précision devient désastreuse si l'on itère cette perte de précision : soit $(u_n)_n$ la suite de réels définie par récurrence de la façon suivante :

$$\begin{aligned} u_0 &= 1/3 \\ u_1 &= 1/3 \\ u_{n+1} &= 10u_n - 1/u_{n-1}, \quad \text{pour } n \geq 2. \end{aligned}$$

La suite est évidemment constante ($u_n = 1/3$ pour tout n), mais l'ordinateur qui calcule cette suite va planter en calculant une suite divergente ! C'est un exemple typique d'instabilité lié au cumul d'erreur et à la dilatation des erreurs.
En revanche l'algorithme définie par la suite

$$\begin{aligned} u_0 &= 1 \\ u_1 &= 1 \\ u_{n+1} &= 0.4u_n + 0.6u_{n-1}, \quad \text{pour } n \geq 2, \end{aligned}$$

et calculé sur machine se montrera robuste. On trouvera $u_n = 1$ pour tout n, à l'erreur machine près.

Notion de consistance et stabilité

On a évoqué l'idée qu'une méthode numérique pouvait se révéler instable et produire un résultat aberrant du fait du cumul et/ou de la dilatation des erreurs au cours des itérations de l'algorithme. On rencontre un autre type d'erreur dans les méthodes numériques : l'erreur de consistance. C'est une erreur qui naît de la construction de la solution approchée puisqu'on introduit justement des méthodes pour palier à l'absence de méthode exacte de résolution. Cette erreur de consistance est répétée un grand nombre de fois au cours de l'algorithme. Ainsi la notion de stabilité de la méthode se pose pour contenir le cumul des

erreurs de consistance comme elle se pose pour contenir le cumul des erreurs machines. Pour diminuer l'erreur de consistance et ainsi gagner en précision, on sera amener à augmenter le nombre d'itérations de l'algorithme. Une méthode instable verra alors la solution numérique se détériorer alors même qu'on cherche à améliorer l'erreur de consistance. Il est ainsi essentiel d'assurer la consistance **et** la stabilité de la méthode numérique.
On insistera particulièrement sur ces deux notions : consistance et stabilité (ou robustesse), lors de l'étude des méthodes de calcul approché d'intégrales et de résolution approchée d'équations différentielles.

Structure du document

Le premier chapitre est consacré à la recherche de zéro de fonctions non-linéaires de $\mathbb{R}$ dans $\mathbb{R}$. Ce problème correspond à la résolution d'une équation non-linéaire que l'on ne sait pas résoudre de façon analytique. Les principaux algorithmes de résolution de ces problèmes seront présentés, la convergence sera étudiée ainsi que la vitesse de convergence.

Le second chapitre expose les outils de l'interpolation. Il s'agit de définir des équations de fonctions, le plus souvent polynomiales, pour approcher une fonction f donnée. Cette fonction f n'est éventuellement connue qu'en un nombre donné de points, par sa valeur et/ou ses dérivées. L'erreur entre f et une fonction interpolante sera quantifiée. La convergence de la fonction interpolante vers la fonction f sera étudiée lorsque le nombre de points d'interpolation tend vers l'infini.

Le troisième chapitre traite du calcul intégral. Ces calculs s'appuient sur les résultats d'interpolation de fonction du chapitre précédent. En effet, les fonctions polynomiales interpolantes sont facilement intégrales par un calcul exacte et conduisent à un calcul approché de la fonction interpolée. L'erreur commise entre les deux intégrales conduit à une erreur de consistance et la répétition de ce calcul sur une somme d'intervalles de petite taille amène à se poser la question de la stabilité (robustesse) de la méthode.

Le quatrième chapitre est consacré à l'approximation numérique des solutions d'équations différentielles. Après avoir brièvement introduit le sens à donner aux solutions des équations différentielles et aux quelques propriétés qualitatives qu'on saura exhiber des équations, on présentera les méthodes numériques permettant de construire des solutions approchées. On construira une suite discrète de valeurs, associé à un pas de discrétisation, pour définir une fonction approchée. Là encore, la notion de consistance et stabilité prend tout son sens lorsque le pas de discrétisation tend vers zéro. L'erreur de consistance tend vers zéro quand le pas de discrétisation tend vers zéro, mais le nombre de valeurs à calculer par récurrence pour définir la fonction approchée tend vers l'infini.

Le cinquième chapitre traite de la résolution de systèmes linéaires. Les problèmes à grand nombre d'inconnues se ramènent (après linéarisation) à la résolution de système linéaire de grande taille. On est alors amené à résoudre algorithmiquement ces systèmes de façon directe comme on le ferait "à la main", par exemple par la méthode du pivot de Gauss. L'écriture algorithmique de ces méthodes permet d'automatiser la résolution. Une approche itérative, dite indirecte, sera également proposée, elle consiste à réaliser une suite de vecteurs obtenus par la somme de vecteurs et de produits matrice vecteur. Cette suite converge vers la solution du système sous certaines conditions à établir. La perte de précision lors de certains calculs et le grand nombre d'opérations élémentaires inhérents à la résolution conduisent à

se poser la question de la précision de la solution. Les notions de perte de précision et cumul des erreurs seront abordés ici à travers la notion de conditionement de la matrice du système linéaire.

Table des matières

Chapitre I

Résolution de zéro de fonctions.

Dans ce chapitre, on s'intéresse en premier lieu à des problèmes à une inconnue réelle solution d'un problème non-linéaire. Ce dernier peut s'écrire sous la forme, trouver $x \in A \subset \mathbb{R}$ tel que

$$f(x) = 0. \tag{I.0.1}$$

On se limitera au cas des fonctions f continues sur A, voire $C^1(A)$ pour la méthode de Newton.
L'objectif est de mettre en évidence la convergence des méthodes employées pour déterminer x solution. L'objectif second est d'assurer le minimum d'évaluation de la fonction f pour obtenir une solution avec une précision donnée. Le coût de ces algorithmes peut paraître dérisoire pour des fonctions réelles définies explicitement. Cependant, il faut imaginer que l'évaluation de f peut provenir d'un processus très couteux comme par exemple le résultat d'un long calcul prenant plusieurs heures ou jours. Réduire le nombre d'appel de la fonction f devient alors essentiel.

Ce type de problème se rencontre en dimension supérieure à 1, on verra brièvement comment étendre les méthodes développées ci-après au cas de fonctions de $\mathbb{R}^n$ dans $\mathbb{R}^d$

1 Exemples d'algorithmes

Nous étudions trois algorithmes.

1.1 Dichotomie

PROPOSITION I.1.1
On suppose f continue de A dans $\mathbb{R}$. On suppose de plus que $[a, b] \subset A$ et que $f(a).f(b) < 0$. Alors l'équation (I.0.1) possède au moins une solution sur $[a, b] \subset A$.

Par le théorème des valeurs intermédiaires, il existe $x \in]a, b[$ solution de (I.0.1).

L'algorithme de dichotomie consiste à partir de 2 points a et b d'images par f de signe contraire, puis de découper l'intervalle en deux, en considérant, pour l'itération suivante, le demi-intervalle $[\alpha, \beta] = [a, (a + b)/2]$ ou $[\alpha, \beta] = [(a + b)/2, b]$ qui conserve la propriété $f(\alpha).f(\beta) < 0$.

Algorithme de Dichotomie :

```
epsilon=...;     ! epsilon est la précision souhaitée
!                   (supérieure à l'erreur machine)
a=...;b=...;     !on recherche un zéro entre a et b
Si f(a)*f(b)>0 alors
               écrire "redéfinir a et b";
               stop;
Fin de si
Tant que b-a>epsilon faire
     c=(a+b)/2.;
     Si f(c)*f(a) <0 alors
                     b=c;
        Sinon
                     a=c;
     Fin de si
Fin de Tant que
```

1.2 Newton

On suppose f de classe C^1 définie sur $\mathbb{R}$. L'algorithme de Newton consiste à partir d'un point x_0 donné, puis de définir par récurrence une suite $(x_n)_n$. Pour x_n donné on construit x_{n+1} en cherchant le zéro du linéarisé de f en x_n. Ainsi sur la figure ci-après, on construit la tangente à la courbe de f au point x_n, puis on définit x_{n+1} par intersection de la tangente et de l'axe des abscisses. L'équation de la tangente est

$$y = f(x_n) + f'(x_n)(x - x_n).$$

Ainsi, en $x = x_{n+1}$, on a

$$y = 0 = f(x_n) + f'(x_n)(x_{n+1} - x_n).$$

Finalement,

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}.$$

REMARQUE I.1.1 *La suite n'est plus définie à partir du rang n_0 dès lors que $f'(x_{n_0-1}) = 0$. La définition de la suite et la convergence de la suite obligera à considérer des hypothèses sur f et/ou le premier itéré.*

REMARQUE I.1.2 *Une variante de la méthode de Newton est la méthode de la corde. Elle consiste à s'affranchir du calcul de la dérivée de f en remplaçant $f'(x_n)$ par le coefficient directeur de la droite passant par $(x_n, f(x_n))$ et $(x_{n-1}, f(x_{n-1}))$. Soit,*

$$x_{n+1} = x_n - f(x_n)\frac{x_n - x_{n-1}}{f(x_n) - f(x_{n-1})}.$$

Algorithme de Newton :

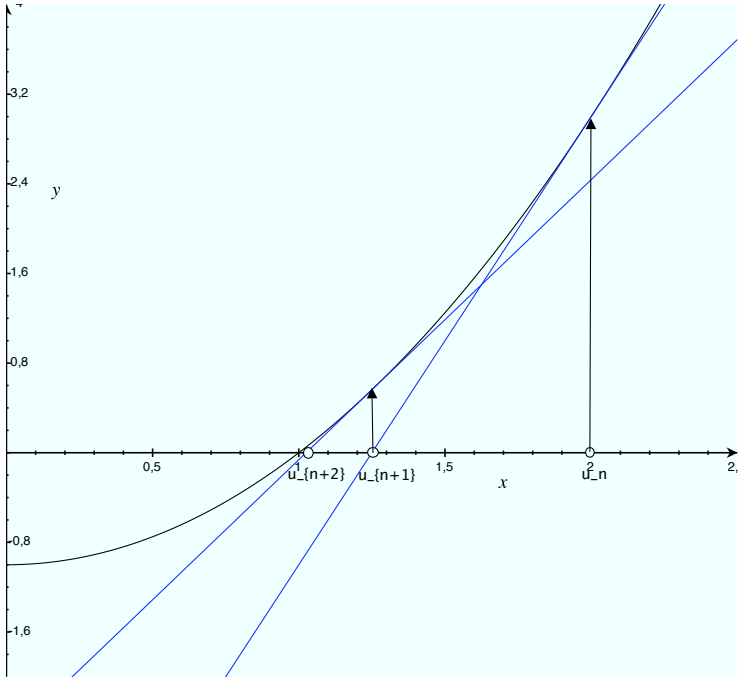

FIG. 1.1 – Méthode de Newton.

```
epsilon=...;      ! epsilon est la précision souhaitée
!                    (supérieure à l'erreur machine)
x=...;            ! choix du premier itéré
Si f'(x)==0 alors
            écrire "algorithme mal défini";
            stop;
sinon
        x9=x-f(x)/f'(x);
Fin de Si
Tant que abs(x9-x)>epsilon faire
     x=x9;
     Si f'(x)==0 alors
            écrire "algorithme mal défini";
            stop;
     sinon
        x9=x-f(x)/f'(x);
     Fin de Si
Fin de Tant que
```

1.3 Lagrange

La méthode de Lagrange est une variante de la méthode de dichotomie. Plutôt que de diviser en deux intervalles de même longueur l'intervalle $[a, b]$, on découpe $[a, b]$ en $[a, c]$ et $[c, b]$ où c est l'abscisse du point d'intersection de la droite passant par $(a, f(a))$ et $(b, f(b))$ et l'axe des abscisses. Soit,

$$b - c = (f(b) - 0)\frac{b - a}{f(b) - f(a)}.$$

Algorithme de Lagrange :

```
epsilon=...;     ! epsilon est la précision souhaitée
!                   (supérieure à l'erreur machine)
a=...;b=...;     !on recherche un zéro entre a et b
Si f(a)*f(b)>0 alors
               écrire "redéfinir a et b";
               stop;
Fin de si
Tant que b-a>epsilon faire
     c=b+f(b)*(a-b)/(f(b)-f(a));
     Si f(c)*f(a) <0 alors
                     b=c;
        Sinon
                     a=c;
     Fin de si
Fin de Tant que
```

2 Convergence des méthodes

2.1 Dichotomie

PROPOSITION I.2.1

On suppose f continue de A dans $\mathbb{R}$. On suppose de plus que $[a, b] \subset A$ et que $f(a).f(b) < 0$. Alors l'algorithme de dichotomie converge vers une solution de (I.0.1) sur $[a, b] \subset A$.

A l'itération n de l'algorithme, on note a_n la borne gauche de l'intervalle de dichotomie et b_n la borne droite de l'intervalle de dichotomie. Pour tout n, on a la propriété suivante qui découle de la découpe de $[a_n, b_n]$ en deux intervalles de même longueur,

$$b_{n+1} - a_{n+1} = \frac{1}{2}(b_n - a_n).$$

Ainsi,

$$b_n - a_n = \frac{1}{2^n}(b_0 - a_0).$$

Comme $f(b_n)f(a_n) \leq 0$, par le théorème des valeurs intermédiaires (f est continue !), il existe un zero de f sur $[a_n, b_n]$. Et comme les suites $(a_n)_n$ et $(b_n)_n$ sont convergentes en tant que suite monotone bornée et que $\lim_{n\to\infty} b_n - a_n = 0$, alors $(a_n)_n$ et $(b_n)_n$ convergent vers le même zéro de f.

2.2 Newton

La convergence de la méthode de Newton consiste à démontrer la convergence de la suite définie par récurrence :

$$x_{n+1} = g(x_n) = x_n - \frac{f(x_n)}{f'(x_n)}. \tag{I.2.1}$$

Si l'algorithme converge, $x_n \to x$ et x vérifie $x = g(x)$ en supposant g continue. On est ramené à un problème de point fixe.

PROPOSITION I.2.2
Si la fonction g définie de l'intervalle $[a, b]$ dans $[a, b]$ est continue, alors g admet un point fixe.

Preuve : On définit la fonction h par $h(x) = g(x) - x$. On a

$$h(a) = g(a) - a \geq 0, \quad h(b) = g(b) - b \leq 0.$$

Comme h est continue, d'après le théorème des valeurs intermédiaires,

$$\exists x \in [a, b] \text{ tel que } h(x) = 0, \text{ soit, } g(x) = x.$$

REMARQUE I.2.1 *L'hypothèse d'invariance de $[a, b]$ par g assure l'existence d'un point fixe, mais ne prouve pas qu'une suite définie par récurrence par $x_{n+1} = g(x_n)$ et $x_0 \in [a, b]$ converge.*

PROPOSITION I.2.3
Soit X une partie fermée de $\mathbb{R}$. Soit g définie de X dans X, lipschitzienne de constante K ($K < 1$) (ie g est contractante).
Alors, il existe un unique point fixe x de g sur X et toute suite vérifiant $x_{n+1} = g(x_n)$ et $x_0 \in X$ converge vers x.

Ce résultat, appliqué au cas particulier de g définie par (I.2.1) donne un premier résultat de convergence de la méthode de Newton.

COROLLAIRE I.2.4
Soit g de $\mathbb{R}$ dans $\mathbb{R}$ de classe $\mathscr{C}^1$ possédant un point fixe x.
Si $|g'(x)| < 1$ alors x est limite de toute suite définie par récurrence par $x_{n+1} = g(x_n)$ pourvu que x_0 soit suffisamment proche de x.
Si $|g'(x)| > 1$ alors x est limite d'aucune suite définie par récurrence par $x_{n+1} = g(x_n)$.

COROLLAIRE I.2.5
Soit f de $\mathbb{R}$ dans $\mathbb{R}$ de classe $\mathscr{C}^2$ possédant un zéro x.
Si $f'(x) \neq 0$ alors x est limite de toute suite définie par récurrence par (I.2.1) pourvu que x_0 soit suffisamment proche de x.

Ces résultats de convergence suppose de partir d'un premier itéré proche du résultat, en pratique, cela est limitatif. Pour une fonction f convexe, on a des résultats de convergence moins restrictif. La convergence de la méthode de Newton est donc soumise à des hypothèses restrictives, c'est le défaut de cette méthode, en revanche sa vitesse de convergence (si convergence il y a) sera son intérêt.

3 Ordre de convergence

On s'intéresse à la vitesse de convergence des méthodes à travers la notion d'ordre. Les méthodes d'ordre élevé converge plus vite, au sens où moins d'itérations de la méthode sont nécessaires pour obtenir la solution pour une précision donnée.

DÉFINITION I.3.1
Soit $(x_n)_n$ une suite de réels et $x \in \mathbb{R}$ tel que $x_n \neq x$ pur tout n et $\lim_{n\to\infty} x_n = x$.
On appelle ordre de convergence de la suite $(x_n)_n$, le plus petit réel $r \geq 1$ (s'il existe) tel que

$$\lim_{n\to\infty} \frac{|x_{n+1} - x|}{|x_n - x|^r} = c \text{ avec } c \neq 0 \text{ et fini.}$$

REMARQUE I.3.1 *L'ordre 1 est obtenu par exemple pour les suites géométriques de raison K ($|K| < 1$).*

DÉFINITION I.3.2
On dit que la suite $(x_n)_n$ converge plus vite vers x que la suite $(y_n)_n$ si

$$\lim_{n\to\infty} \frac{|x_n - x|}{|y_n - x|} = 0.$$

PROPOSITION I.3.3
Si la suite $(x_n)_n$ convergent vers x est d'ordre r_1 , si la suite $(y_n)_n$ convergent vers x est d'ordre r_2, avec $r_1 > r_2$ alors la suite $(x_n)_n$ converge plus vite que $(y_n)_n$.

3.1 Dichotomie

PROPOSITION I.3.4
La dichotomie est une méthode d'ordre 1.

En effet, la suite des longueurs de l'intervalle de dichotomie est géométrique.

3.2 Newton

PROPOSITION I.3.5
La méthode de Newton est une méthode d'ordre 2 au moins.

4 Exercices

Exercice 1.1. On cherche à évaluer $\sqrt{a}$ à l'aide d'algorithmes autorisant toutes les opérations élémentaires. Soient $(x_n)_n$ et $(y_n)_n$ deux suites définies par récurrence :

$$x_{n+1} = \frac{2ax_n}{x_n^2 + a}, \quad y_{n+1} = \frac{y_n}{2a}(3a - y_n^2).$$

1. Montrer la convergence des suites $(x_n)_n$ et $(y_n)_n$ pour x_0 et y_0 bien choisis.
2. Déterminer l'ordre de convergence de ces suites.

3. Choisissez x_0 et y_0 pour $a = 3$.

Exercice 1.2. Soit f une fonction de classe $\mathcal{C}^2$ sur $[a, b]$ avec $f(a)f(b) < 0$. On suppose de plus que f' et f'' ne s'annulent pas sur $[a, b]$.

1. Montrer que f est strictement monotone.
2. Montrer que il existe un unique s élément de $]a, b[$ tel que $f(s) = 0$.
3. Montrer que le graphe de f est du même côté par rapport toute tangente à la courbe sur $[a, b]$.
4. Si $x_0 \in [a, b]$ vérifie $f(x_0)f''(x_0) > 0$, montrer que la suite $(x_n)_n$ définie par $x_{n+1} = g(x_n) = x_n - \frac{f(x_n)}{f'(x_n)}$ est convergente vers s. Quel est le nom de l'algorithme associé à cette suite ? Caractériser l'ensemble des x_0 vérifiant l'hypothèse.

Exercice 1.3. Soit $f(x) = x^3 - x^2 - 1$. On se propose de trouver les racines réelles de f.

1. Situer la ou les racines de f. Montrer qu'il y a une racine l comprise entre 1 et 2.
2. Soit les méthodes itératives suivantes, $x_0 \in [1, 2]$,

$$x_{n+1} = x_n^3 - x_n^2 + x_n - 1 \tag{I.4.1}$$

$$x_{n+1} = \frac{1}{x_n^2 - x_n} \tag{I.4.2}$$

$$x_{n+1} = (x_n^2 + 1)^{\frac{1}{3}}. \tag{I.4.3}$$

   (a) Examiner la convergence et la limite de ces méthodes.

   (b) Préciser l'ordre des méthodes convergentes.

Exercice 1.4. On dispose d'une calculatrice qui ne sait faire que des additions, des soustractions et des multiplications. On cherche à l'utiliser pour calculer l'inverse $1/a$ d'un réel $a > 0$. Cela revient à calculer un zéro de la fonction $f : x \mapsto x^{-1} - a$.

1. Ecrire la méthode de Newton pour cette fonction. Cette algotithme peut-il être implémenté sur la calculatrice ?
2. Dans le cas $a = 5$, calculer les termes x_1 et x_2 pour $x_0 = 0, 1$, $x_0 = 1$ puis $x_0 = 2$.

Exercice 1.5. On définit f sur $\mathbb{R}_+^*$ par $x \to x - e-(1 + x)$.

1. Montrer qu'il existe un unique zéro l de f. Localiser l entre deux entiers successifs.
2. La méthode suivante converge-t-elle vers l ?

$$x_0 \text{ donné}, \quad x_{n+1} = g(x_n), \text{ où } g(x) = e^{-(1+x)}.$$

3. La méthode suivante converge-t-elle vers l ?

$$x_0 \text{ donné}, \quad x_{n+1} = h(x_n), \text{ où } h(x) = x^2 e^{1+x}.$$

4. La méthode suivante converge-t-elle vers l ?

$$x_0 \text{ donné}, \quad x_{n+1} = k(x_n), \text{ où } k(x) = -1 - \ln(x).$$

5. Choisir x_0 et donner l'ordre de la première méthode.
6. Entre ces méthodes et celle de Newton, quelle est la plus efficace ?

Exercice 1.6. On définit f sur $\mathbb{R}_+^*$ par $f(x) = \ln(x) + x^2 + 2x - 5$.

1. Montrer qu'il existe un unique zéro l de f. Localiser l entre deux entiers successifs a et $a+1$.
2. Montrer que la méthode suivante ne converge pas :

$$x_0 \in [a, a+1], \quad x_{n+1} = x_n - f(x_n).$$

3. On propose la méthode,

$$x_0 \in [a, a+1], \quad x_{n+1} = x_n - \alpha f(x_n) = g_\alpha(x_n).$$

Quelles sont les valeurs de α qui assurent la convergence de la suite vers l quel que soit le choix de $x_0 \in [a, a+1]$. Quel est le choix optimal de α.

Chapitre II

Interpolation.

L'interpolation permet d'associer au graphe d'une fonction dont l'évaluation et/ou quelques dérivées sont connues en un nombre donné de points, une fonction définie en tout point par une expression analytique. L'interpolation polynomiale en est un exemple classique. Outre l'intérêt visuel du graphe, on peut définir les dérivées, les primitives d'une telle fonction interpolée ou simplement évaluer la fonction interpolée en un point quelconque.

1 Interpolation de Lagrange

C'est l'interpolation polynomiale la plus simple, elle consiste à interpoler un nuage de points appartenant à un graphe.

1.1 Définition

Soit $n+1$ points de $\mathbb{R}^2$ de coordonnées $(x_i, f_i = f(x_i))$, $i = 0 \cdots n$, avec $x_i \neq x_j$ si $i \neq j$.
On cherche à interpoler f par un polynôme p :

THÉORÈME II.1.1
Il existe un unique polynôme p de degré au plus n tel que

$$\forall 0 \leq i \leq n, \quad p(x_i) = f_i.$$

Le polynôme p s'appelle le polynôme de Lagrange (associé aux points $(x_i, f_i = f(x_i))$, $i = 0 \cdots n$). On dira que p interpole f en x_i, $i = 0 \cdots n$.

Preuve : on note l_i le polynôme défini par

$$l_i(x) = \frac{\prod\limits_{\substack{j=0\cdots n \\ j\neq i}} (x - x_j)}{\prod\limits_{\substack{j=0\cdots n \\ j\neq i}} (x_i - x_j)}.$$

On remarque que l_i est de degré n et que

$$l_i(x_j) = \delta_{ij}.$$

On rappelle que l'espace vectoriel P_n des polynômes de degré au plus n est de dimension $n+1$. La famille $(l_i)_{i=0\cdots n}$ est une famille libre de P_n : si,

$$\forall x \in \mathbb{R}, \quad \sum_{i=0}^{n} \alpha_i l_i(x) = 0,$$

alors, en choisissant $x = x_i$ pour tout $i = 0 \cdots n$, on a $\alpha_i l_i(x_i) = \alpha_i = 0$.
La famille $(l_i)_{i=0\cdots n}$ constituée de $n+1$ éléments est donc une base de P_n. Ainsi, pour tout $q \in P_n$, q se décompose de façon unique sur la base $(l_i)_{i=0\cdots n}$. Finalement, le poynôme $p = \sum_{i=0}^{n} f_i l_i$ est l'unique polynôme de P_n tel que $p(x_i) = f_i$ pour $i = 0 \cdots n$.

1.2 Représentation polynomiale et coût de l'évaluation

On a vu qu'on pouvait représenter le polynôme de Lagrange interpolant $(x_i, f_i = f(x_i))$, $i = 0 \cdots n$, par la formule

$$p(x) = \sum_{i=0}^{n} f_i \frac{\displaystyle\prod_{\substack{j=0\cdots n \\ j\neq i}} (x - x_j)}{\displaystyle\prod_{\substack{j=0\cdots n \\ j\neq i}} (x_i - x_j)}. \tag{II.1.1}$$

Pour tout x donné, on peut évaluer numériquement $p(x)$ par la formule (II.1.1). Cependant, si on est amené à évaluer souvent cette quantité en différents points, cette expression va s'avérer peu juducieuse, du fait du grand nombre d'opérations élémentaires intervenant dans (II.1.1).
On peut déjà remarquer qu'il est inutile de reproduire le calcul des constantes

$$a_i = \prod_{\substack{j=0\cdots n \\ j\neq i}} (x_i - x_j).$$

On calculera donc au préalable les a_i pour $i = 0 \cdots n$. Le calcul

$$p(x) = \sum_{i=0}^{n} f_i \frac{\displaystyle\prod_{\substack{j=0\cdots n \\ j\neq i}} (x - x_j)}{a_i}. \tag{II.1.2}$$

reste néanmoins encore bien coûteux avec un nombre d'opérations toujours en $O(n^2)$.
On peut reécrire

$$\prod_{\substack{j=0\cdots n \\ j\neq i}} (x-x_j) = \frac{1}{x-x_i} \prod_{j=0\cdots n} (x-x_j),$$

ainsi,

$$p(x) = \sum_{i=0}^{n} f_i \frac{1}{x-x_i} \prod_{j=0\cdots n} (x-x_j), \tag{II.1.3}$$

on est ainsi ramené, pour $x \neq x_i$, au calcul de $\sum_{i=0}^{n} f_i \frac{1}{x-x_i}$ qui nécessite un $0(n)$ opérations et au calcul de $\prod_{j=0\cdots n} (x-x_j)$ qui nécessite également un $0(n)$ opérations.

1.3 Estimation d'erreur d'interpolation

Soit une fonction f donnée, et soit p le polynôme de Lagrange interpolant f en x_i, $i = 0\cdots n$, $x_0 < x_1 < \cdots < x_n$. On cherche à quantifier l'écart entre f et p en fonction de x afin de mesurer l'erreur d'interpolation.

THÉORÈME II.1.2
On suppose que $x_i \in [a,b]$ pour $i = 0\cdots n$.
Si f est $C^{n+1}([a,b])$, alors il existe $\xi \in [\min(x,x_0), max(x,x_n)]$ tel que

$$f(x) - p(x) = \prod_{i=0\cdots n} \frac{(x-x_i)}{(n+1)!} f^{(n+1)}(\xi). \tag{II.1.4}$$

REMARQUE II.1.1 *Si les points x_i sont voisins, le produit $\prod_{i=0\cdots n} (x-x_i)$ est petit pour $x \in [\min(x_i), \max(x_i)]$.*
On pourrait alors être tenté d'augmenter le nombre de points d'interpolation pour réduire l'erreur, mais l'erreur croît avec l'étalement des points d'interpolation et le facteur $\frac{f^{(n+1)}(\xi)}{(n+1)!}$ peut croître (comme décroître) selon f.

2 Interpolation d'Hermite

On construit une interpolation polynômiale à l'aide de points d'un graphe et des dérivées en ces points.

2.1 Définition

On se donne $n+1$ triplets $(x_i, f_i, f_i') \in \mathbb{R}^3$ pour $i = 0\cdots n$.
On cherche à construire un polynôme p tel que $p(x_i) = f_i$ et $p'(x_i) = f_i'$ pour $i = 0\cdots n$.

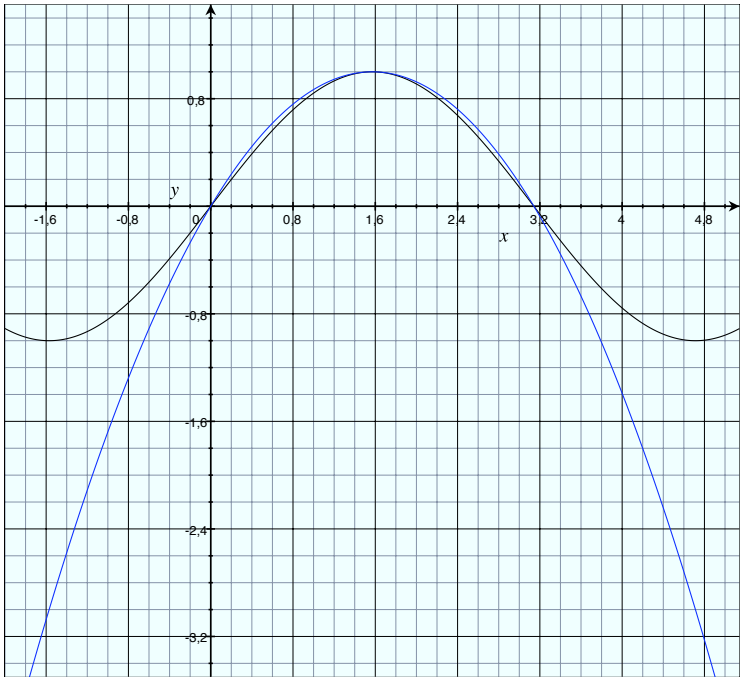

FIG. 1.1 – Interpolation de Lagrange de $x \longrightarrow \sin x$ en 0, $\pi/2$ et π.

THÉORÈME II.2.1
Il existe un unique polynôme p de degré au plus $2n+1$ tel que $p(x_i) = f_i$ et $p'(x_i) = f'_i$ pour $i = 0 \cdots n$ Le polynôme p s'appelle le polynôme d'Hermite (associé aux points (x_i, f_i, f'_i), $i = 0 \cdots n$). On dira que p interpole f en x_i, $i = 0 \cdots n$ si $f(x_i) = f_i$ et $f'(x_i) = f'_i$ pour $i = 0 \cdots n$.

2.2 Estimation d'erreur d'interpolation

Comme pour l'interpolation de Lagrange, on quantifie l'erreur entre une fonction f régulière et son polynôme d'interpolation d'Hermite :

THÉORÈME II.2.2
On suppose que $x_i \in [a, b]$ pour $i = 0 \cdots n$.
Si f est $C^{2n+2}([a, b])$, alors il existe $\xi \in [\min(x, x_0), max(x, x_n)]$ tel que

$$f(x) - p(x) = \prod_{i=0\cdots n} \frac{(x - x_i)^2}{(2n+2)!} f^{(2n+2)}(\xi).$$

Un exemple d'interpolation d'Hermite est donné sur la figure 2.1. On peut comparer le résultat avec l'interpolation de Lagrange en figure 1.1.

REMARQUE II.2.1 *On peut construire des interpolations hybrides entre Lagrange et Hermite, en se servant de points du graphe et des dérivées en quelques-un de ces points. On peut également construire des interpolations en interpolant le graphe en quelques points et les dérivées à différents ordres en ces*

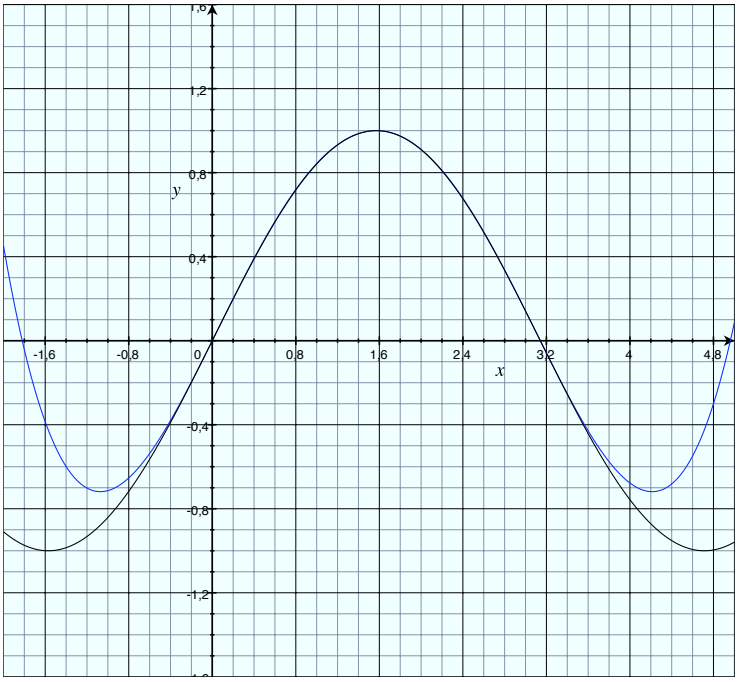

FIG. 2.1 – Interpolation d'Hermite de $x \to \sin x$ en 0, $\pi/2$ et π.

points. En interpolant plusieurs dérivées en un point, on collera encore mieux à la courbe autour de ce point.

3 Choix des points d'interpolation

Dans le cas de l'interpolation polynomiale de Lagrange ou d'Hermitte, on cherche une répartition des points d'interpolation permettant de diminuer l'erreur d'interpolation.
D'après la formule d'erreur d'interpolation (II.1.4), on cherche à minimiser $\sup_{[a,b]} |v|$ avec $v(x) = \prod_{i=0}^{n}(x - x_i)$, par le choix des x_i sur le segment $[a, b]$. On remarque que pour l'interpolation d'Hermitte, c'est le carré de la même expression qu'on cherche à minimiser.
Quitte à translater et à dilater l'intervalle, on peut, sans perte de généralité supposer que $[a, b] = [-1, 1]$.
Optimiser le choix des points revient à optimiser le choix du polynôme v et

$$v \in F_{n+1} = \{p \in P_{n+1} \text{ tel que } p(x) = x^{n+1} + q(x) \text{ avec } q \in P_n\}. \qquad \text{(II.3.1)}$$

On va chercher $v \in F_{n+1}$, s'il existe, tel que

$$\forall p \in F_{n+1}, \quad \max_{x\in[-1,1]} v(x) \leq \max_{x\in[-1,1]} p(x).$$

On vérifiera que v possède $n + 1$ racines dans $[-1, 1]$ et ainsi ces $n + 1$ racines seront les points d'interpolation optimaux.

3.0.1 Polynôme de Tchebycheff

DÉFINITION II.3.1
On appelle polynôme de Tchebycheff de degré n, le polynôme de degré n T_n

$$\begin{aligned} T_n :\ [-1,1] &\longrightarrow \mathbb{R} \\ x \quad &\longrightarrow \cos(n\text{Arccos } x) = \sum_{k=0}^{|n/2|} (-1)^k C_n^k x^{n-2k}(1-x^2)^k. \end{aligned}$$

PROPOSITION II.3.2
Les polynômes de Tchebycheff sont définis pare récurrence

$$\begin{aligned} &T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x), \\ &T_0(x) = 1, \quad T_1(x) = x. \end{aligned}$$

PROPOSITION II.3.3
Les zéros du polynôme de Tchebycheff T_n sont définis par

$$a_k = \cos(\frac{2k-1}{2n}\pi), \quad k = 1 \cdots n.$$

3.0.2 Choix optimal

THÉORÈME II.3.4
Le plus petit polynôme de F_n défini par (II.3.1), au sens de la norme $L^\infty([-1,1])$ est $\overline{T}_n = \frac{1}{2^{n-1}} T_n$:

$$\forall v \in F_n, \quad \frac{1}{2^{n-1}} = \max_{x\in[-1,1]} |\overline{T}_n(x)| \leq \max_{x\in[-1,1]} |v(x)|.$$

PROPOSITION II.3.5
On note $(a_k)_{k=1\cdots n}$ les zéros du polynôme de Tchebycheff T_n et on appelle p le polynôme d'interpolation de Lagrange d'une fonction f régulière aux points $(a_k)_{k=1\cdots n}$, on a l'estimation d'erreur :

$$\max_{x\in[-1,1]} |f(x) - p(x)| \leq \frac{1}{n!} \frac{1}{2^{n-1}} \max_{x\in[-1,1]} |f^{n+1}(x)|. \tag{II.3.2}$$

REMARQUE II.3.1 *Même avec une repartition optimale des points, il n'est pas forcément intéressant d'augmenter le nombre de points d'interpolation, en effet le membre de droite de (II.3.2) ne décroît pas forcément avec n. De plus se pose la question de propagation des erreurs dans l'évaluation d'un polynôme d'interpolation dont les coefficients des monômes peuvent devenir grand quand n augmente. En pratique, on ne dépassera pas $n = 10$. On préférera interpoler une fonction en un grand nombre de points par paquets de n points ($n < 10$).*

4 Autres Interpolations

Dans cette section, on étend les techniques d'interpolation à des interpolations non nécessairement polynômiales, on optimise le choix des points d'interpolation et on introduit les splines qui surpassent largement l'interpolation polynomiale à trop grand nombre de points.

4.1 Interpolation sur un espace vectoriel

On a jusqu'ici construit des polynômes interpolant une fonction f. On a donc construit une interpolation sur un espace vectoriel P_k (les polynômes de degré inférieur ou égal à k). On se donne désormais un espace vectoriel quelconque E de dimension n et on note $(e_1, \cdots, e_n)$ une base de E. On prendra E un sous-espace vectoriel de $C^0(\mathbb{R}, \mathbb{R})$. Ainsi, e_i est une fonction continue réelle de la variable réelle. On peut par exemple construire une base de fonctions trigonométriques.
On se donne n couples (x_i, f_i), $i = 1 \cdots n$, avec $x_i \neq x_j$ si $i \neq j$. On cherche f, une fonction de E tel que $f(x_i) = f_i$ pour $i = 1 \cdots n$. On note A la matrice de $\mathscr{M}_{n,n}(\mathbb{R})$ définie par

$$A = \begin{pmatrix} e_1(x_1) & \cdots & e_n(x_1) \\ \cdots & \cdots & \cdots \\ e_n(x_1) & \cdots & e_n(x_n) \end{pmatrix} \tag{II.4.1}$$

PROPOSITION II.4.1
Si $\det(A) \neq 0$*, alors il existe une unique fonction de* E *qui interpole les points* (x_i, f_i), $i = 1 \cdots n$. *Si* $\det(A) = 0$*, les points d'interpolation sont mal choisi et on a l'alternative : soit il existe une infinité de fonctions de* E *qui interpole les points* (x_i, f_i), $i = 1 \cdots n$*, soit il n'existe aucune fonction de* E *interpolant les points* (x_i, f_i), $i = 1 \cdots n$.

REMARQUE II.4.1 *Le choix de l'espace* E *à travers le choix des fonctions de base n'est pas anodin, selon les fonctions à interpoler, il convient de choisir des fonctions de base adaptées. De plus, le choix des points d'interpolation est important, il permet d'assurer* $\det(A) \neq 0$ *mais aussi de réduire l'erreur d'interpolation.*

REMARQUE II.4.2 *L'interpolation présentée ici est de type Lagrange, on peut également étendre les techniques d'interpolation de type Hermite sur un espace vectoriel quelconque de fonctions, de dimension adaptée (dimension* $2n$ *pour* n *points d'interpolations* (x_i, f_i, f_i')*).*

Preuve : On cherche $f \in E$ sous la forme

$$f = \sum_{j=1}^{n} u_j e_j, \quad \text{avec } u_j \in \mathbb{R}.$$

Ainsi,

$$f(x_i) = \sum_{j=1}^{n} u_j e_j(x_i), \quad \text{pour } i = 1 \cdots n.$$

Déterminer f revient à déterminer le vecteur $u \in \mathbb{R}^n$, $u = (u_1, \cdots, u_n)^t$, tel que

$$\begin{pmatrix} f(x_1) \\ \cdots \\ f(x_n) \end{pmatrix} = \begin{pmatrix} e_1(x_1) & \cdots & e_n(x_1) \\ \cdots & \cdots & \cdots \\ e_n(x_1) & \cdots & e_n(x_n) \end{pmatrix} \begin{pmatrix} u_1 \\ \cdots \\ u_n \end{pmatrix} = Au$$

Le vecteur u est défini de façon unique si et seulement si $\det(A) \neq 0$.

4.2 Les splines

L'idée est de raccorder des interpolations polynomiales de degré faible ($2 \leq n \leq 5$) en imposant une régularité au niveau des raccords des polynômes.
On dispose de m points, $x_1 < x_2 < \cdots < x_m$, on cherche une interpolation de f en ces points et on dispose seulement de l'évaluation de f en ces points.

Exemple1 : On définit les splines affines de la façon suivante : soit s_i la fonction affine définie sur $[x_i, x_{i+1}]$ par $s_i(x_i) = f(x_i)$ et $s_i(x_{i+1}) = f(x_{i+1})$. Ainsi, $s_i(x) = f(x_i) + \frac{x-x_i}{x_{i+1}-x_i} f(x_{i+1})$, pour $i = 1 \cdots m-1$.
Les splines ainsi construites sont continues sur $[x_1, x_m]$, mais elles n'ont aucune raison d'être dérivables en x_i, pour $i = 2 \cdots m-1$.

Exemple2 : On définit les splines paraboliques de la façon suivante : soit s_i la parabole définie sur $[x_i, x_{i+1}]$ par $s_i(x_i) = f(x_i)$, $s_i(x_{i+1}) = f(x_{i+1})$ et $s_i'(x_i) = s_{i-1}'(x_i)$. On vérifie (exercice) que la parabole s_i est bien définie dès lors que s_{i-1} est connue.
Reste à définir la spline s_1 afin d'initier la récurrence de définition des splines. On choisira par exemple $s_1'(x_1) = \frac{f(x_2)-f(x_1)}{x_2-x_1}$.
Les splines ainsi construites sont C^1 sur $[x_1, x_m]$.

Exemple3 : On définit les splines cubiques de la façon suivante : soit s_i la cubique définie sur $[x_i, x_{i+1}]$ par $s_i(x_i) = f(x_i)$, $s_i(x_{i+1}) = f(x_{i+1})$, $s_i'(x_i) = s_{i-1}'(x_i)$ et $s_i''(x_i) = s_{i-1}''(x_i)$. On vérifie (exercice) que la cubique s_i est bien définie dès lors que s_{i-1} est connue.
Reste à définir la spline s_1 afin d'initier la récurrence de définition des splines. On choisira par exemple $s_1'(x_1) = \frac{f(x_2)-f(x_1)}{x_2-x_1}$ et $s_1''(x_1) = 0$.
Les splines ainsi construites sont C^2 sur $[x_1, x_m]$.

Les splines cubiques interpolent la courbe rouge sur la figure 4.1, aux points indiqués par des ronds. On notera la précision accrue en augmentant le nombre de points d'interpolation. Pour un nombre de points impaire, on a la chance d'avoir un point d'interpolation qui capte le maximum de la fonction, la précision de l'interpolation est alors, par chance, améliorée.

5 Exercices

Exercice 1.7. Soit f la fonction définie sur $\mathbb{R}$ par $f(x) = \dfrac{1}{1+x^2}$

1. Construire le polynôme d'interpolation P_3 de Lagrange sur les points 0, 1, 2 et 4.
2. Calculer $P_3(3)$ que l'on comparera à $f(3)$.
3. Evaluer f''' et estimer l'erreur commise entre f et P_3 sur $[0, 4]$.
4. Construire le polynôme d'interpolation P_4 de Lagrange sur les points 0, 1, 3 et 5.
5. Calculer $P_4(4)$ que l'on comparera à $f(4)$.
6. Evaluer $f^{(4)}$ et estimer l'erreur commise entre f et P_4 sur $[0, 5]$.
7. Construire le polynôme d'interpolation Q_3 de Hermite sur les points 0 et 5.

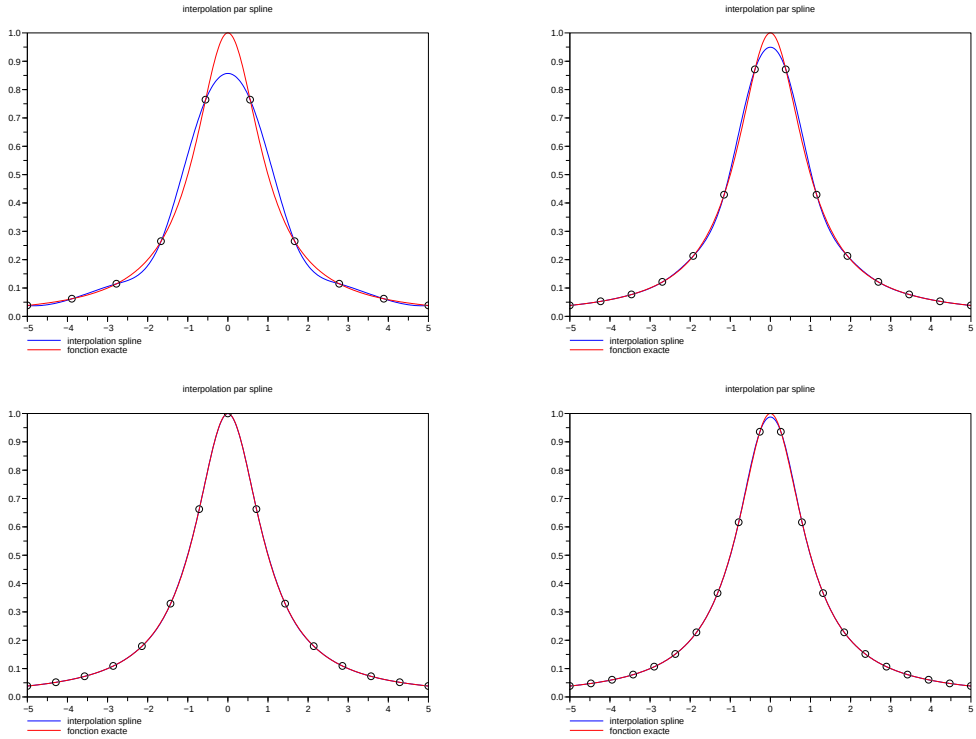

FIG. 4.1 – Splines cubiques à 10, 14, 15 et 20 points.

8. Calculer $Q_3(4)$ que l'on comparera à $f(4)$. Commentez par rapport à la question 5.
9. Estimer l'erreur commise entre f et Q_3 sur $[0, 5]$.

Exercice 1.8. Soit $h = \frac{b-a}{N}$ le pas des N subdivisions de l'intervalle $[a, b]$. Pour i entier, $0 \leq i \leq N$, on définit $x_i = a + ih$ et pour $0 \leq i \leq N-1$, on pose $x_{i+\frac{1}{2}} = a + ih + \frac{1}{2}$. On connaît f évalué en x_i : $f(x_i) = b_i$ pour $0 \leq i \leq N-1$. Sur chaque intervalle $[x_i, x_{i+1}]$, pour approcher f, on cherche un polynôme s_i de degré le plus bas possible tel que

$$s_i(x_{i+\frac{1}{2}}) = b_i, \quad 0 \leq i \leq N-1,$$
$$s_{i-1}(x_i) = s_i(x_i), \quad 0 \leq i \leq N-1.$$

1. Quel est le degré de chaque polynôme s_i.
2. Déterminer les relations liant les coefficients des polynômes s_i.
3. Calculer ces polynômes dans le cas où $s_0(a) = 0$.
4. Ecrire ces polynômes sur l'intervalle $[0, 1]$ lorsque $b_i = i + \frac{1}{2}$.

Exercice 1.9. Soit E l'espace vectoriel des fonctions de $\mathbb{R}$ dans $\mathbb{R}$ engendré par les fonctions e_1 et e_2, $e_1(x) = \sin(x)$ et $e_2(x) = \sin(2x)$.

1. Quelle est la dimension de E.

2. Interpoler sur l'espace E, aux points $\pi/4$ et $\pi/3$, une fonction f quelconque.
3. Interpoler sur l'espace E, aux points $\pi/3$ et $\pi/2$, une fonction f quelconque.
4. Interpoler sur l'espace E, aux points 0 et $\pi/3$, une fonction f quelconque. Que se passe-t-il ?

Exercice 1.10. Soit f une fonction passant par les points $(-1, 1)$, $(-1/2, 9/16)$, $(0, 1)$, $(1/2, 31/16)$, $(1, 1)$.

1. Déterminer le polynôme d'interpolation de Lagrange, P, qui passe par les points précités. Calculer $P(2)$, $P(-2)$.
2. Déterminer la droite d'interpolation, Q, qui passe au plus prés des points précités au sens des moindres carrés. Calculer $Q(2)$, $Q(-2)$.

Chapitre III

Intégration numérique.

A l'aide d'outils mathématiques tels que l'intégration par partie et les changements de variable, on arrive parfois à se ramener au calcul d'intégrale de fonction dont on connaît une primitive. On peut dans ce cas faire du calcul exacte d'intégrale. Des outils de calcul formel permettent également de déterminer une primitive lorsque celle-ci possède une expression analytique. Ce n'est pas toujours le cas. Il peut également arriver que la fonction que l'on cherche à intégrer ne soit pas définie par une expression analytique. On a alors recours au calcul approché d'intégrale. On se limitera au calcul intégral sur $\mathbb{R}$. Le calcul approché d'intégrale est largement inspiré de la construction de l'intégrale de Riemann.

1 Formule de quadrature

Soit $[a,b] \subset \mathbb{R}$ et f une fonction de $\mathscr{C}^0([a,b];\mathbb{R})$.
L'objectif est de calculer

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

On note $\{x_0, x_1, \cdots, x_n\}$ une subdivision de $[a,b]$: $a = x_0 < x_1 < \cdots < x_n = b$.

DÉFINITION III.1.1
On appelle formule de quadrature à $n+1$ points, l'expression

$$I_n = \sum_{k=0}^{n} A_k^n f(x_k),$$

destinée à approcher I.

Exemple : approchons I par une formule dite des rectangles à droites :

$$I_n = \sum_{k=1}^{n} (x_k - x_{k-1}) f(x_k),$$

Cette formule correspond à l'intégration exacte d'une fonction g_n constante par marceaux définie par

$$\forall k \in [1,n]_{\mathbb{N}},\ \forall x \in [x_{k-1}, x_k[, \quad g_n(x) = f(x_k).$$

On montre alors que

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

Ainsi, $A_0^n = 0$ et $A_k^n = x_k - x_{k-1}$ pour $1 \leq k \leq n$ définissent les coefficients de la formule de quadrature pour la méthode des rectangles à droite.
L'objectif visé dans les prochaines sections est d'établir des propriétés de convergence de la formule de quadrature vers I lorsque n tend vers l'infini et également de quantifier l'erreur commise en fonction de n. Le but est ainsi d'obtenir une grande précision dans l'approximation de I pour un indice n le plus faible possible.

2 Approximation polynômiale

On a vu au chapitre précédent qu'on pouvait approcher une fonction par un polynôme. On sait aisément calculer la primitive d'un polynôme. L'idée consiste alors à proposer des formules de quadrature basées sur l'intégration d'interpolations polynômiales.

2.1 Interpolation P_k

On se limitera ici à l'interpolation de Lagrange. On se donne $k+1$ points $\xi_0 < \xi_1 < \cdots < \xi_k$, on note p_k le polynôme de Lagrange interpolant les points $(\xi_j, f(\xi_j))_{0 \leq j \leq k}$. On commet alors l'approximation :

$$\int_{\xi_0}^{\xi_k} f(x)dx \sim \int_{\xi_0}^{\xi_k} p_k(x)dx = \sum_{j=0}^{k} A_j^k f(\xi_j).$$

Comme on l'a vu au chapitre précédent, il n'est pas raisonnable de choisir $k > 10$. On peut néanmoins choisir une formule de quadrature à grand nombre de points en groupant les points par paquet de $k+1$ points. On note $\{x_0, x_1, \cdots, x_n\}$ une subdivision de $[a,b]$: $a = x_0 < x_1 < \cdots < x_n = b$. On interpole alors la fonction f sur $[x_{j-1}, x_j[$ par p_k^j le polynôme de Lagrange de f aux points $(x_{j-1} + \xi_l, f(x_{j-1} + \xi_l))_{0 \leq l \leq k}$ avec $0 \leq \xi_0 < \xi_1 < \cdots < \xi_k \leq x_j - x_{j-1}$. On obtient alors une formule de quadrature du type

$$I = \int_a^b f(x)dx \sim I_n = \sum_{j=1}^{n} \sum_{l=0}^{k} A_l^k f(x_{j-1} + \xi_l). \tag{III.2.1}$$

Exemple : Formule des trapèzes.
Ecrivons la formule de quadrature basée sur une approximation P_1 sur tout ségment $[x_{j-1}, x_j[$. On obtient :

$$I_n = \sum_{j=1}^{n} (x_j - x_{j-1})(f(x_{j-1}) + f(x_j))/2 = \frac{x_1 - x_0}{2} f(x_1) + \frac{x_n - x_{n-1}}{2} f(x_n) + \sum_{j=1}^{n-1} \frac{x_{j+1} - x_{j-1}}{2} f(x_j).$$

2.2 Newton-Cotes

Les formules de quadrature de Newton-Cotes sont obtenues par l'intégration d'interpolations de Lagrange à l'aide de points equirépartis. Soient $\{\xi_0, \xi_1, \cdots, \xi_k\}$ tels que $\xi_i = \xi_0 + i(\xi_k - \xi_0)/(k+1)$, on

appelle p_k le polynôme de Lagrange interpolant f en $(\xi_i)_{i=0\cdots k}$, la formule de quadrature est obtenue par

$$\int_{\xi_0}^{\xi_k} p_k(x)dx = \sum_{j=0}^{k} A_j^k f(\xi_j).$$

On note $h = (\xi_k - \xi_0)/(k+1)$ la distance entre deux points consécutifs de la subdivision, que l'on appelle le pas de subdivision. Ce pas est ici uniforme.
Avec trois points, on a la formule dite de Simpson :

$$\int_{\xi_0}^{\xi_2} p_2(x)dx = 2h(\frac{1}{6}f(\xi_0) + \frac{2}{3}f(\xi_1) + \frac{1}{6}f(\xi_2)).$$

Avec quatre points, on a la formule de Newton-Cotes suivante :

$$\int_{\xi_0}^{\xi_3} p_3(x)dx = 3h(\frac{1}{8}f(\xi_0) + \frac{3}{8}f(\xi_1) + \frac{3}{8}f(\xi_2) + \frac{1}{8}f(\xi_3)).$$

Avec cinq points, on a la formule de Newton-Cotes suivante :

$$\int_{\xi_0}^{\xi_4} p_4(x)dx = 4h(\frac{7}{90}f(\xi_0) + \frac{32}{90}f(\xi_1) + \frac{12}{90}f(\xi_2) + \frac{32}{90}f(\xi_3)\frac{7}{90}f(\xi_4)).$$

REMARQUE III.2.1 *Afin de réduire les calculs inutiles, il convient d'exploiter :*
- l'invariance par translation de l'intégrale (pour se ramener à l'origine),
- le changement de variable $y = (x - \xi_0)/h$ (pour se ramener aux points d'interpolation $0, 1, \ldots k$),
- la symétrie dans les coefficients de la formule de quadrature (résultant de la symétrie des points d'interpolation par rapport au point milieu de l'intervalle d'intégration).

3 Erreur d'approximation et convergence

3.1 Erreur locale

DÉFINITION III.3.1
Soient $\{\xi_0, \xi_1, \cdots, \xi_k\}$ une subdivision, et A_l^k les coefficients d'une formule de quadrature en ces points. On appelle erreur locale d'intégration pour une fonction f, la quantité

$$E(f) = \int_{\xi_0}^{\xi_k} f(x)dx - \sum_{l=0}^{k} A_l^k f(\xi_l).$$

Si la formule de quadrature est basée sur une interpolation P_k et en notant p_k le polynôme de Lagrange interpolant la fonction f en $(\xi_i)_{i=0\cdots k}$, l'erreur locale d'intégration devient

$$E(f) = \int_{\xi_0}^{\xi_k} f(x) - p_k(x)dx.$$

THÉORÈME III.3.2
Soit f une fonction $\mathscr{C}^{k+1}[a,b]$. L'erreur locale d'intégration pour une interpolation P_k est au plus un $O(h^{k+1})$.

Il existe un résultat plus fin pour une subdivision equi-répartie :

THÉORÈME III.3.3
Soit f une fonction $\mathscr{C}^{k+2}[a,b]$. L'erreur locale d'intégration pour la méthode de Newton-Cotes à $k+1$ points (interpolation P_k) est au plus un $O(h^{k+2})$ si k est paire.

3.2 Stabilité

Par une interpolation P_k, l'erreur locale est décroissante avec k (pour $h << 1$). Cela suggère de choisir k grand. La suite va montrer que ce choix n'est pas forcément judicieux.

DÉFINITION III.3.4
Une formule de quadrature I_n est dite stable s'il existe une constante M indépendante de n tel que

$$\sum_{k=0}^{n} |A_k^n| \leq M,$$

avec

$$I_n = \sum_{k=0}^{n} A_k^n.$$

Cette notion de stabilité assure que si les calculs de la formule de quadrature sont entachés d'erreur, le cumul des erreurs sera contrôlé lorsque n tend vers l'infini.

PROPOSITION III.3.5
Les formules de Newton-Cotes sont instables pour une interpolation P_n lorsque n tend vers l'infini.

PROPOSITION III.3.6
Les formules de quadrature basées sur une juxtaposition d'interpolation P_k (k fixé) (III.2.1) sont stables.

3.3 Convergence

DÉFINITION III.3.7
Soit f une fonction définie sur $[a,b]$. Soit $\{x_0, x_1, \cdots, x_n\}$ une subdivision de $[a,b]$ respectant la propriété de répartition quasi-uniforme, ie, il existe $C > 0$ tel que

$$a = x_0 < x_1 < \cdots < x_n = b, \quad \forall 0 < i \leq n,\ |x_{i+1} - x_i| \leq C\frac{b-a}{n}.$$

Soient $(A_n^k)_k$ les coefficients d'une formule de quadrature en ces points. La formule de quadrature $I_n = \sum_{k=0}^{n} A_k^n f(x_k)$ est convergente lorsque n tend vers l'infini si

$$\lim_{n\to\infty} I_n = I = \int_a^b f(x)dx.$$

REMARQUE III.3.1 *Cette notion de convergence est théorique, elle ne tient pas compte des erreurs lors du calcul de I_n. Pour s'assurer de la validité d'une méthode, on s'assurera de la convergence* **et** *de la stabilité de la méthode.*

THÉORÈME III.3.8
Soit f une fonction $\mathscr{C}^{k+1}[a,b]$. Soit $\{x_0, x_1, \cdots, x_n\}$ une subdivision de $[a,b]$ respectant la propriété de répartition quasi-uniforme. Alors, les formules de quadrature basées sur une juxtaposition d'interpolation P_l ($l \leq k$, k fixé) (III.2.1) sont convergentes.

4 Formule de Gauss

Soit $\{x_0, x_1, \cdots, x_n\}$ une subdivision de $[a,b]$. On considère une formule de quadrature à $n+1$ points qui s'écrit

$$\sum_{k=0}^{n} A_k^n f(x_k). \tag{III.4.1}$$

L'objectif est de choisir les points de la subdivision ainsi que les coefficients de la formule de quadrature afin que l'erreur soit la plus petite possible pour aprocher

$$\int_a^b f(x)dx.$$

On dispose ainsi de $2n+2$ degrés de libertés (les $n+1$ coefficients A_k^n et les $n+1$ points x_k). On va faire en sorte que la formule de quadrature soit exacte sur un espace vectoriel de fonctions de dimension $2n+2$. On choisit l'espace vectoriel P_{2n+1}.

PROPOSITION III.4.1
La formule de quadrature (III.4.1) est exacte sur l'espace vectoriel P_{2n+1} si et seulement si
– la formule de quadrature (III.4.1) est exacte sur l'espace vectoriel P_n
– $\forall\, q \in \mathbb{N},\ q \leq n$,

$$\int_a^b x^q v(x)dx = 0, \quad v(x) = \prod_{i=0}^{n}(x - x_i).$$

PROPOSITION III.4.2
Il existe une unique subdivision de $[a,b]$, $\{x_0, x_1, \cdots, x_n\}$, telle que si $v(x) = \prod_{i=0}^{n}(x - x_i)$, alors, $\forall\, q \in \mathbb{N},\ q \leq n$,

$$\int_a^b x^q v(x)dx = 0.$$

THÉORÈME III.4.3
Pour la subdivision obtenue à la proposition précédente, les coefficients de la formule de quadrature (III.4.1) s'écrivent

$$A_k^n = \int_a^b \frac{\displaystyle\prod_{i=0, i\neq k}^{n}(x - x_i)^2}{\displaystyle\prod_{i=0, i\neq k}^{n}(x_k - x_i)^2} dx.$$

De plus, la formule de quadrature est stable.

5 Méthode de Romberg

Voir TD.

Chapitre IV

Equations différentielles ordinaires et approximation numérique.

1 EDO et modélisation

On s'intéresse aux équations différentielles ordinaire (EDO) du premier ordre. Ce sont des équations dont l'inconnue est une fonction (d'une variable réelle à valeurs dans R^n) dont la dérivée s'exprime en fonction d'elle même : Soit $t \to u(t)$ une solution de

$$u'(t) = f(t, u(t)).$$

Ces problèmes n'ont pas toujours de solutions, ou les solutions existent éventuellement que sur des intervalles courts $t \in]t_0, t_1[$. L'étude fine de l'existence relève du programme de L3-Math et ne sera que brièvement abordée ici.

Les équations issues de la Physique, mécanique, chimie,... s'écrivent souvent sous forme d'équations différentielles. En effet, on est souvent en mesure de modéliser l'accroissement d'une quantité inconnue en fonction d'elle même. Cela concerne souvent des problèmes où la variable de l'inconnue est le temps, c'est pourquoi on choisira d'appeler t la variable de la fonction inconnue.

Exemple1 : la mécanique du point. On décrit la vitesse (vecteur de R^3) d'une particule par la dérivée de sa position (vecteur de R^3) :

$$v(t) = x'(t).$$

La vitesse répond à la loi fondamentale de la dynamique (Newton) :

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

où f est la force par unité de masse agissant sur le point. Par exemple, f est une force de frottement proportionnelle à la vitesse du point :

$$v'(t) = -\frac{1}{m} v(t).$$

Autre exemple, la force dépend de la position x, on est ramené au système d'EDO :

$$x'(t) = v(t),$$
$$v'(t) = f(x(t)).$$

Exemple2 : la Chimie. Les réactions chimiques sont décrites par des EDO. On quantifie l'accroissement d'une concentration chimique en lisant la réaction chimique et l'échelle de temps associé à la réaction (k_i). Soit une réaction chimique affectant les espèces chimiques A, B C et D :

$$A + B \longrightarrow^{k_1} C$$
$$C + D \longrightarrow^{k_2} A$$

On en déduit le système décrivant l'accroissement des différentes concentrations chimiques a, b, c et d des espèces A, B, C et D en construit le système ainsi :

$$a'(t) = -k_1 a(t)b(t) + k_2 c(t)d(t),$$
$$b'(t) = -k_1 a(t)b(t),$$
$$c'(t) = +k_1 a(t)b(t) - k_2 c(t)d(t),$$
$$d'(t) = -k_2 c(t)d(t).$$

Pour plus de détail et en particulier pour intégrer la notion d'ordre de réaction chimique, se référer à [1]. Les deux réactions chimiques sont supposées d'ordre 1 ici.

Exemple3 : Dynamique des populations. On s'intéresse à l'évolution d'une ou plusieurs densité de populations.
Un modèle à croissance Maltusienne est décrit par l'acroissement d'une population qui est proportionelle à sa population (aucun facteur ne limite la reproduction, ni n'augmente la mortalité, qui sont uniformes dans le temps) :

$$P'(t) = nP(t).$$

La population est alors à croissance exponentielle ou décroissance exponentielle selon le signe de n correspondant au fait que la natalité l'emporte ou non sur la mortalité.
Un modèle prédateur-proie couple l'évolution de deux populations :

$$L'(t) = n_l L(t) - c_p L(t)R(t),$$
$$R'(t) = -m_r R(t) + c_v L(t)R(t).$$

Pour un modèle à une espèce, un terme logistique exprime que la population est un prédateur pour elle-même.

$$P'(t) = nP(t) - cP^2(t) = nP(t)(1 - \frac{c}{n}P(t)). \qquad \text{(IV.1.1)}$$

Le facteur n/c s'appelle la capacité d'accueil du milieu. La population se régule vers la capacité d'accueil du milieu, comme le montre la solution sur la figure 1 où la capacité d'accueil est de 10.

La donnée de l'EDO ne suffit pas à définir une solution unique, on précise alors une donnée initiale : la connaissance de la solution à un instant donné. L'EDO et une donnée initiale définissent un problème de Cauchy.

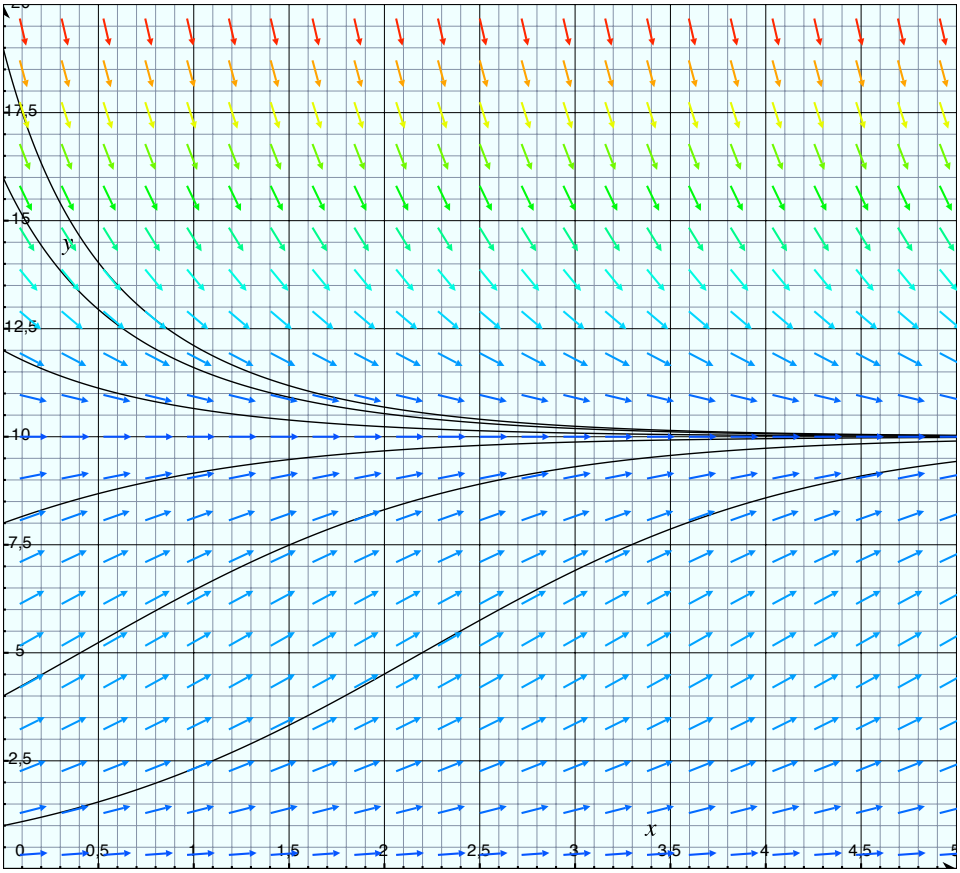

FIG. 1.1 – Solutions de (IV.1.1), graphe de P en fonction de t.

2 Problèmes de Cauchy

DÉFINITION IV.2.1
On appelle problème de Cauchy du premier orde, la donnée d'une EDO du premier ordre associé à une condition initiale (ou donnée initiale) :

$$\begin{aligned} u'(t) &= f(t, u(t)), \\ u(t_0) &= u_0, \end{aligned} \tag{IV.2.1}$$

où la fonction inconnue u vérifie $u(t) \in \mathbb{R}^n$.

2.1 Existence de solutions

THÉORÈME IV.2.2

(Cauchy-Lipschitz.) On suppose que f est continue de $I \times \mathbb{R}^n$ dans $\mathbb{R}^n$ avec I un intervalle contenant t_0. On suppose de plus que f est localement Lischitzienne par rapport au deuxième argument, ie

$$\forall\, t \in I,\ \forall\, B \subset \mathbb{R}^n, \quad \exists C \in \mathbb{R} \text{ et } \eta > 0 \text{ tels que}$$
$$\forall\, s \in I \cap [t-\eta, t+\eta],\ \forall\, (u,v) \in B^2, \quad |f(s,u) - f(s,v)| \leq C|u-v|.$$

($|.|$ désigne la norme de $\mathbb{R}^n$.) Alors, il existe $(t_1, t_2) \in I^2$, $t_1 < t_0$, $t_2 > t_0$, et une unique solution u de (IV.2.1) défini sur $[t_1, t_2]$ noté $([t_1, t_2], u)$.

On attache à la notion de solution, l'intervalle sur lequel on définit une solution. On cherche en général à définir la solution sur le plus grand intervalle possible.

DÉFINITION IV.2.3

Soit (I_1, u_1) et (I_2, u_2) deux solutions de (IV.2.1) tels que $I_1 \subset I_2$, on a $u_1 = u_{2|I_1}$ par unicité de la solution.
On dit que (I_2, u_2) prolonge (I_1, u_1).

REMARQUE IV.2.1 *Le théorème IV.2.2 prouve l'existence d'une solution dite locale. Si on sait prolonger la solution sur I tout entier, on parlera de solution globale sur I. Si la solution est prolongée sur $]t_1, t_2[\subset I$ de sorte que la solution ne puisse plus être prolongée : $\lim_{t \to t_1^-} u(t) \notin \mathbb{R}^n$ ou $t_1 = \inf I$, et $\lim_{t \to t_1^-} u(t) \notin \mathbb{R}^n$ ou $t_2 = \sup I$, la solution $(]t_1, t_2[, u)$ est dite maximale. Ainsi, sous les hypothèses du théorème IV.2.2 de Cauchy-Lipschitz, il existe une unique solution maximale (I, u).*

REMARQUE IV.2.2 *Si f est $C^1(I \times \mathbb{R}^n)$ alors f est continue et localement Lipschitz, elle vérifie donc les hypothèses du théorème IV.2.2.*

Exemple1 : le problème suivant ne possède pas de solutions globales sur $\mathbb{R}$ autre que 0 :

$$\begin{aligned} u'(t) &= u^2(t), \\ u(t_0) &= u_0 \in \mathbb{R}. \end{aligned} \tag{IV.2.2}$$

On peut appliquer le théorème IV.2.2 en vérifiant le caractère localement Lipschitz de $x \to x^2$. Il existe donc une unique solution locale à (IV.2.2). On peut vérifier que cette solution s'écrit, si $u_0 \neq 0$:

$$u(t) = \frac{1}{\frac{1}{u_0} - t}.$$

Si $u_0 > 0$ la solution maximale est $(]-\infty, \frac{1}{u_0}[, u)$.
Si $u_0 < 0$ la solution maximale est $(]\frac{1}{u_0}, +\infty[, u)$.
Les solutions sont représentées sur le graphe 2.1.
Exemple2 : on propose le problème de Cauchy suivant où l'inconnue est une fonction de $\mathbb{R}^2$ dans $\mathbb{R}^2$:

$$\begin{aligned} u'(t) &= -v(t) - 0,05u(t) + \cos(2u(t)), \\ v'(t) &= u(t) - 0,05v(t) + sin(2v(t)), \\ u(t_0) &= u_0 \in \mathbb{R}, \\ v(t_0) &= v_0 \in \mathbb{R}. \end{aligned} \tag{IV.2.3}$$

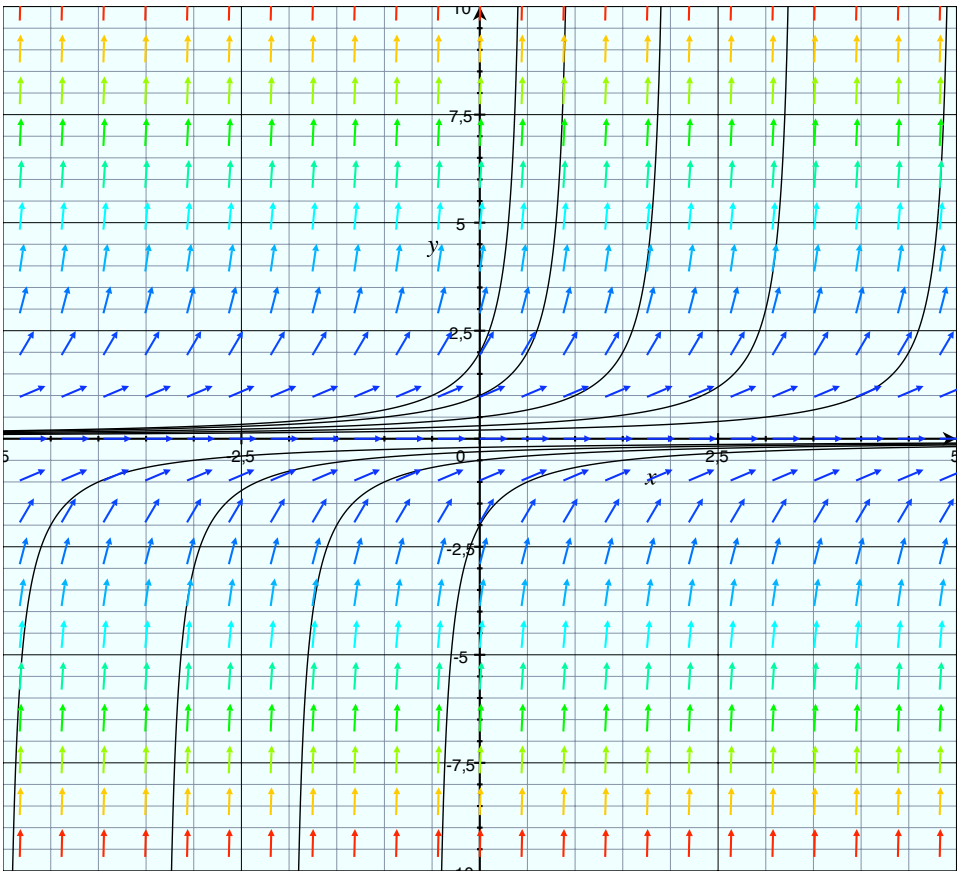

FIG. 2.1 – Solutions de (IV.2.2), graphe de u en fonction de t.

On peut appliquer le théorème IV.2.2 en vérifiant la régularité $\mathscr{C}^1$ de l'application de $\mathbb{R}^2$ dans $\mathbb{R}^2$ définie par

$$\begin{pmatrix} x \\ y \end{pmatrix} \longrightarrow \begin{pmatrix} -y - 0,05x + \cos(2x) \\ x - 0,05y + sin(2y) \end{pmatrix}.$$

Il existe donc une unique solution locale à (IV.2.3). On pourra vérifier par la suite que la solution globale existe.
On choisit de représenter graphiquement les solutions de système 2×2 dans le plan de phase, *ie* en portant u en abscisse et v en ordonnée et en représentant la solution paramétrée par le temps t, voir figure 2.2.

Exemple3 : Voici un exemple de non unicité de la solution lorsque l'hypothèse Lipschitz n'est pas réalisée pour f :

$$\begin{aligned} u'(t) &= \sqrt{u(t)}, \\ u(t_0) &= 0 \in \mathbb{R}. \end{aligned} \tag{IV.2.4}$$

On note que la fonction racine carré n'est pas Lipschitz au voisinage de 0 (tangente verticale en zéro sur

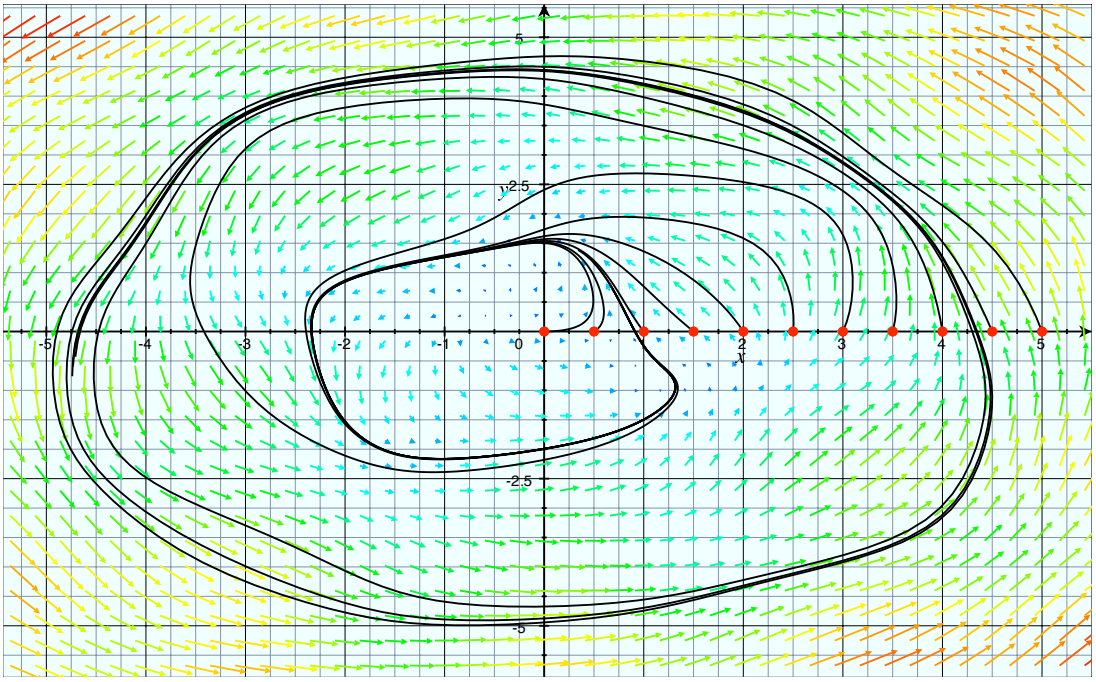

FIG. 2.2 – Solutions de (IV.2.3) dans le plan de phase, $t \in [0, 10]$.

le graphe de la fonction racine carré). On remarque que 0 est solution pour tout temps, mais n'est pas la seule solution. En effet $u(t) = (1/2t)^2$ est aussi solution sur $\mathbb{R}^+$.

2.2 Propriétés qualitatives

Pour montrer qu'une solution est globale, on montre que la solution est bornée sur tout intervalle borné inclus dans I. Pour cela, on peut utiliser des propriétés de comparaison de solution ou le lemme de Gronwall.

PROPOSITION IV.2.4
Soient a et b deux fonctions rélles continues sur $[t_0, t^*[$ et u dérivable sur $[t_0, t^*[$ tel que

$$\forall t \in [t_0, t^*[, \quad u'(t) \leq a(t)u(t) + b(t).$$

Alors,

$$\forall t \in [t_0, t^*[, \quad u(t) \leq u(t_0) + \exp(\int_{t_0}^{t} a(\tau)d\tau) + \int_{t_0}^{t} \exp(\int_{s}^{t} a(\tau)d\tau)b(s)ds.$$

PROPOSITION IV.2.5
(lemme de Gronwall.) Si a est une fonction $C^1(\mathbb{R}, \mathbb{R})$, si b et u sont deux fonctions $C^0(\mathbb{R}, \mathbb{R})$ telles que

$$u(t) \leq u_0 + \int_{t_0}^{t} a(s)u(s)ds + \int_{t_0}^{t} b(s)ds,$$

alors,

$$u(t) \leq u_0 \exp\left(\int_{t_0}^{t} a(s)ds\right) + \int_{t_0}^{t} \exp\left(\int_{s}^{t} a(\tau)d\tau b(s)ds\right).$$

REMARQUE IV.2.3 *Ces deux propostions ne s'applique pas que pour des EDO d'inconnue $u(t) \in \mathbb{R}$. On peut les appliquer aux système d'EDO d'inconnue $U(t) \in \mathbb{R}^n$ en travaillant sur une inégalité vérifiée par $u(t) = \|U(t)\|^2_{\mathbb{R}^n}$ où tout autre norme ou quantité scalaire qui est une fonction de $U(t)$.*

Exemple : Soit le système d'EDO défini par

$$\begin{aligned} u'(t) &= -u(t)v(t), \\ v'(t) &= u^2(t), \\ u(t_0) &= u_0 \in \mathbb{R}, \quad v(t_0) = v_0 \in \mathbb{R}. \end{aligned} \tag{IV.2.5}$$

Le théorème IV.2.2 assure l'existence d'une unique solution locale. On montre maintenant que la solution est globale sur $I = \mathbb{R}$. En effet la solution est bornée pour tout temps :

$$(u^2(t) + v^2(t))' = 2u(t)u'(t) + 2v(t)v'(t) = -2u^2(t)v(t) + 2u^2(t)v(t) = 0.$$

Sur le graphe 2.3, on voit en effet que le graphe de la solution dans le plan de phase est inclus dans un cercle centré à l'origine. La solution converge asymptotiquement vers le point $(0, \sqrt{u_0^2 + v_0^2})$.

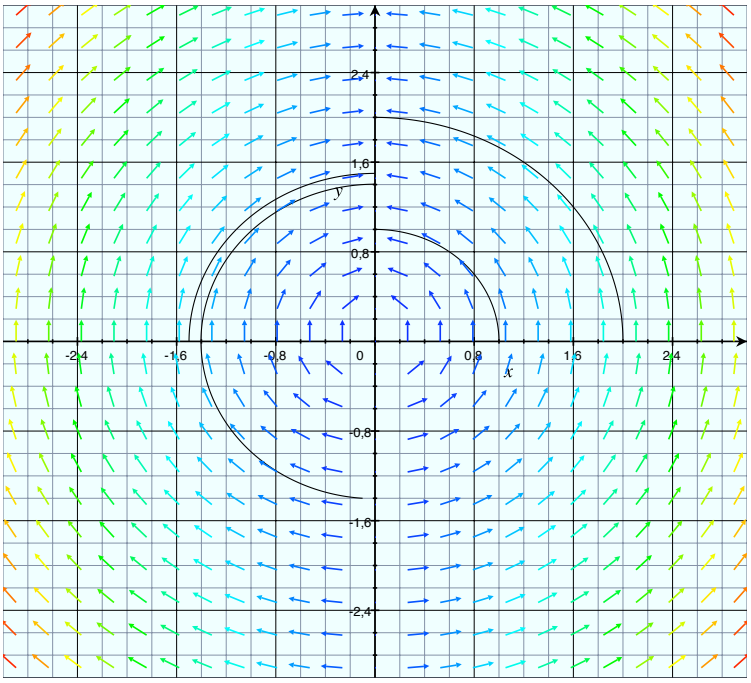

FIG. 2.3 – Solutions de (IV.2.5) dans le plan de phase.

Exemple : Le système d'EDO suivant, défini par

$$\begin{aligned} u'(t) &= -v(t), \\ v'(t) &= u(t), \\ u(t_0) &= u_0 \in \mathbb{R}, \quad v(t_0) = v_0 \in \mathbb{R}, \end{aligned} \tag{IV.2.6}$$

réalise la même propriété de conservation. En effet,

$$(u^2(t) + v^2(t))' = 2u(t)u'(t) + 2v(t)v'(t) = -2u(t)v(t) + 2u(t)v(t) = 0.$$

Sur le graphe 2.4, on voit en effet que le graphe de la solution dans le plan de phase décrit un cercle centrée en l'origine.

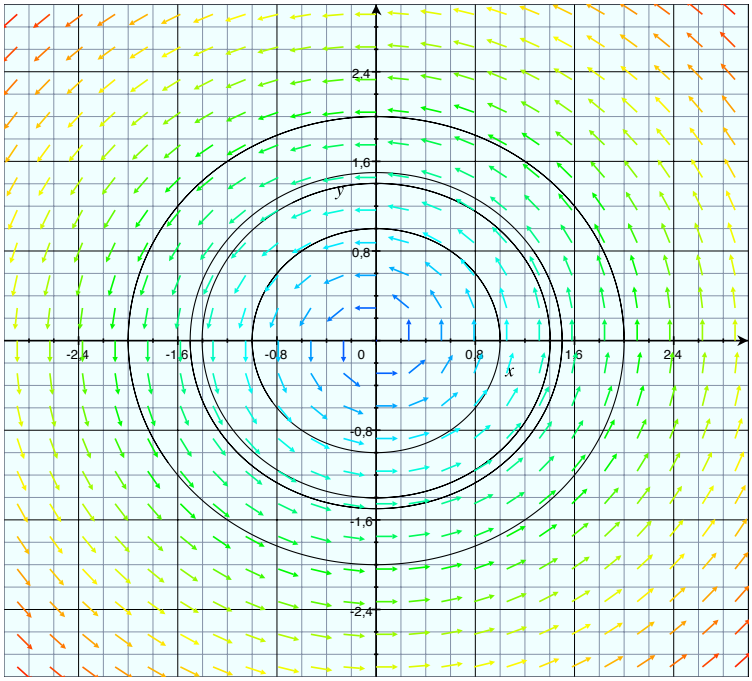

FIG. 2.4 – Solutions de (IV.2.6) dans le plan de phase.

PROPOSITION IV.2.6

(Positivité.) On suppose qu'un système d'EDO peut s'écrire sous la forme,

$$\begin{aligned} u'(t) &= u(t)g(t, u(t), v(t)), \\ v'(t) &= h(t, u(t), v(t)), \\ u(t_0) &= u_0 \in \mathbb{R}, \quad v(t_0) = v_0 \in \mathbb{R}^n, \end{aligned} \tag{IV.2.7}$$

avec g et h continues et localement lipschitz par rapport à l'argument (u, v).
Alors, il existe une unique solution maximale $(I, (u, v))$ telle que

$$\forall\, t \in I, \quad u(t)u_0 > 0 \text{ si } u_0 \neq 0.$$
$$\forall\, t \in I, \quad u(t) = 0 \text{ si } u_0 = 0.$$

Exemple : Le système d'EDO suivant, l'oregonateur, décrit un modèle simplifié de réaction chimique.

$$\begin{aligned} u'(t) &=, \\ v'(t) &=, \\ u(t_0) = u_0 \in \mathbb{R}, &\quad v(t_0) = v_0 \in \mathbb{R}, \end{aligned} \tag{IV.2.8}$$

où.... Les concentrations intiales sont positives. D'après la proposition IV.2.6, on peut montrer que les concentrations restent positives au cours du temps. Le graphe des solutions de (IV.2.8) est donnée sur la figure 2.5. On renvoie à [2] pour plus de détails.

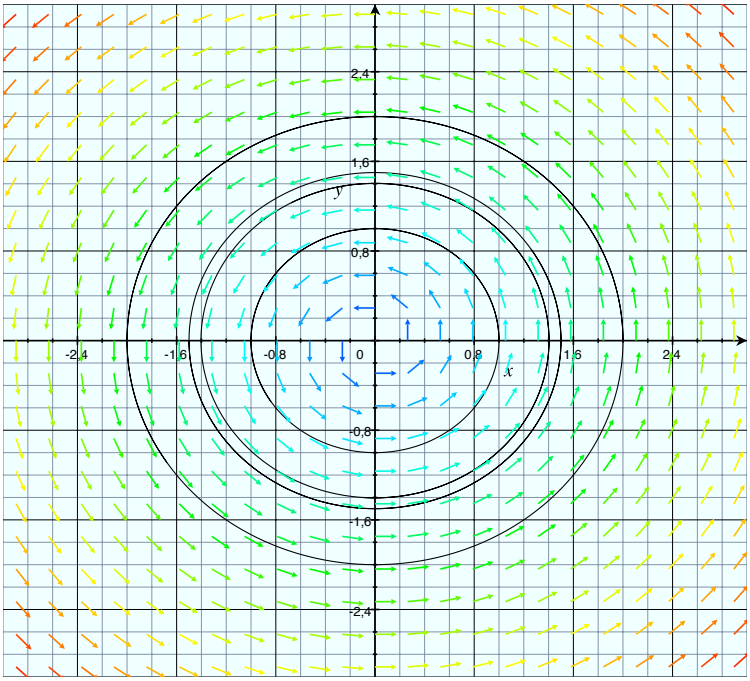

FIG. 2.5 – Solutions de (IV.2.8) dans le plan de phase.

Exemple : Le système d'EDO suivant, le système de Lorentz, est représenté par le graphe 2.6. Il met en évidence le caractère chaotique (sensibilité aux perturbations) des trajectoires qu'un système non-linéaire d'EDO de $\mathbb{R}^3$ peut générer.

$$\begin{aligned} x'(t) &= 10(y - x), \\ y'(t) &= 28x - y - xz, \\ z'(t) &= -\frac{8}{3}z + xy, \\ x(0) = -10, &\quad y(0) = 10, \quad z(0) = 25. \end{aligned} \tag{IV.2.9}$$

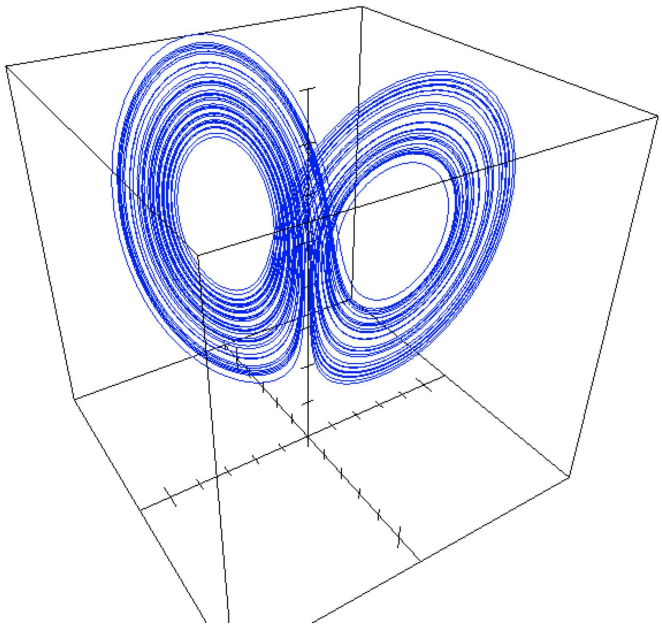

FIG. 2.6 – Solution de (IV.2.9) dans le plan de phase, $0 \leq t \leq 60$.

3 Dicrétisation des EDO

Les solutions des systèmes rencontrés précédemment possèdent rarement des solutions explicites. On propose alors une méthode approchée pour représenter les solutions comme on peut les voir sur les différentes figures présentées ci-dessus.

3.1 Introduction d'un schéma

3.1.1 Approximation de la dérivée

Soit u une fonction dérivable de $\mathbb{R}$ dans $\mathbb{R}^n$. Par définition de la dérivée, on a

$$u'(t) = \lim_{\Delta t \to 0} \frac{u(t+\Delta t) - u(t)}{\Delta t}.$$

Pour construire des solutions approchées des EDO, on va remplacer la dérivée de u par la quantité

$$\frac{u(t+\Delta t) - u(t)}{\Delta t},$$

pour $\Delta t > 0$ fixé à une "petite" valeur. On appellera Δt le pas de temps et on cherchera la fonction inconnue u à travers une suite discrète de valeurs approchées $(u(k\Delta t))_k$. On va définir une telle suite par récurrence à l'aide d'une approximation de l'EDO.
On appelera discrétisation de l'EDO cette méthode qui consiste à transformer un problème continue

(l'inconnue est une fonction de la variable réelle) en un problème discret (l'inconnue est une suite approchant la solution en un ensemble discret de points $k\Delta t$). L'équation discrète remplaçant l'EDO s'appelle le schéma numérique.

3.1.2 Schéma d'Euler explicite

Le schéma d'Euler est le schéma numérique le plus simple à construire. On approche l'équation définie pour $t \in \mathbb{R}^+$,

$$\begin{aligned} &u'(t) = f(t, u(t)), \\ &u(0) = u, \end{aligned}$$

par le schéma

$$\begin{aligned} &\forall\, n \in \mathbb{N}, \quad \frac{u_{n+1} - u_n}{\Delta t} = f(t_n, u_n), \\ &u_0 = u, \end{aligned} \tag{IV.3.1}$$

où $t_n = n\Delta t$ et u_n est censé approcher $u(t_n)$.
Ce schéma est explicite au sens où la suite $(u_n)_n$ est définie explicitement par la formule de récurrence,

$$u_{n+1} = u_n + \Delta t f(t_n, u_n).$$

3.1.3 Schéma d'Euler implicite

Le schéma d'Euler implicite est construit selon le même principe mais avec la fonction f évaluée au temps t_{n+1} :

$$\begin{aligned} &\forall\, n \in \mathbb{N}, \quad \frac{u_{n+1} - u_n}{\Delta t} = f(t_{n+1}, u_{n+1}), \\ &u_0 = u. \end{aligned} \tag{IV.3.2}$$

Le schéma est ici implicite car, pour u_n connu, on doit trouver u_{n+1} comme solution de (IV.3.2). C'est un problème qui revient à rechercher le zéro d'une fonction comme étudié au premier chapitre.

3.2 Convergence d'un schéma

L'objectif est de montrer que lorsque le pas de temps tend vers zéro, la solution approchée construite par le schéma numérique converge vers la solution de l'EDO.
On s'intéresse aux schémas dits à un pas définis de la façon suivante

$$\begin{aligned} &\forall\, n \in \mathbb{N}, \quad u_{n+1} = F(t_n, u_n, \Delta t), \\ &u_0 \text{ donné dans } \mathbb{R}^n. \end{aligned} \tag{IV.3.3}$$

Pour le schéma d'Euler explicite, F s'exprime comme,

$$F(t_n, u_n, \Delta t) = u_n + \Delta t f(t_n, u_n).$$

3.2.1 Convergence

DÉFINITION IV.3.1
Soit une fonction u solution d'une EDO définie sur $[0,T]$ au moins telle que $u(0)=u_*$. Un schéma est dit convergent si

$$\lim_{u_0\to u_*}\lim_{\Delta t\to 0}\max_{n\leq\frac{T}{\Delta t}}|u_n-u(t_n)|=0.$$

REMARQUE IV.3.1 *Dans cette définition de convergence, on s'assure que l'erreur sur la donnée initiale (erreur machine par exemple) se contrôle au cours des itérations du schéma. Pour approcher le temps $T>0$, le nombre d'itérations tend vers l'infini quand Δt tend vers zéro. Ainsi la définition de convergence réclame que le cumul des erreurs faites à chaque itération tend vers zéro alors que le nombre d'itérations tend vers l'infini.*

3.2.2 Consistance

La consistance d'un schéma consiste à s'assurer que l'erreur commise sur une itération du schéma est suffisamment faible.

DÉFINITION IV.3.2
Un schéma défini par (IV.3.3) est dit consistant avec la solution u si

$$\lim_{u_0\to u_*}\sum_{n=0}^{|T/\Delta t|}|u(t_{n+1})-F(t_n,u(t_n),\Delta t)|=0.$$

REMARQUE IV.3.2 *Dans cette définition, on aura remarquer que $F(t_n,u(t_n),\Delta t)|$ correspond à la prédiction de la solution au temps t_{n+1} par le schéma (IV.3.3) à partir de la solution exacte au temps t_n : $u(t_n)$.*
Une condition suffisante de consistance est

$$\forall\, n\leq\frac{T}{\Delta t},\quad |u(t_{n+1})-F(t_n,u(t_n)|\leq\Delta t g(\Delta t),$$

avec g une fonction continue telle que $g(0)=0$.
La consistance n'assure pas la convergence du schéma car la notion de consistance ne tient pas compte du cumul des erreurs.

3.2.3 Stabilité

La stabilité d'un schéma assure que le cumul des erreurs commises au cours des itérations par le schéma est contrôlé.

DÉFINITION IV.3.3
Un schéma défini par (IV.3.3) est dit stable s'il existe $\Delta t^*>0$ et $M>0$ tel que pour tout $u_0\in\mathbb{R}^n$, $v_0\in\mathbb{R}^n$,

$$\forall\,\Delta t\leq\Delta t^*,\ \forall(\varepsilon_n)_n,\ \varepsilon_n\in\mathbb{R},$$

la suite $(u_n)_n$ définie par (IV.3.3) et la suite $(v_n)_n$ définie par

$$\forall\, n \in \mathbb{N}, \quad v_{n+1} = F(t_n, v_n, \Delta t) + \varepsilon_n,$$
$$v_0 \text{ donné dans } \mathbb{R}^n,$$

vérifient

$$\forall\, n \leq \frac{T}{\Delta t}, \quad |u_n - v_n| \leq M(|u_0 - v_0| + \sum_{k=0}^{n-1} |\varepsilon_k|.$$

THÉORÈME IV.3.4
Si le schéma (IV.3.3) est tel que F est lipschitz par rapport à l'argument u_n, et est consistant et stable, alors le schéma est convergent.

REMARQUE IV.3.3 *La consistance du schéma découle d'un choix raisonable du schéma. La stabilité est toujours acquise dès lors que Δt^* est suffisamment petit et que la propriété lipschitz de F est vérifiée. Néanmoins, cette propriété "automatique" n'est pas vrai pour des équations plus générales que les EDO étudiées ici.*

3.3 Schéma numérique et propriétés qualitatives

3.3.1 Schéma implicite et A-stabilité

Un schéma nunérique normalement construit est toujours stable pour des pas de temps petits. Pour éviter qu'un programme plante à cause d'un pas de temps trop grand n'assurant pas la propriété de stabilité, il est intéressant de construire des schémas stables inconditionnellement sur le pas de temps.
Nous allons mettre en évidence le phénomène d'instabilité sur l'EDO suivante

$$u'(t) = -\lambda u(t), \quad \text{avec } \lambda > 0.$$

On connaît explicitement la solution : $u(t) = u(0)\exp(-\lambda t)$ qui est une solution bornée pour $t \geq 0$ et tendant vers 0 quand t tend vers l'infini.
Le schéma d'Euler explicite (IV.3.1) appliqué à cette équation donne

$$u_{n+1} = (1 - \lambda\Delta t)u_n.$$

C'est une suite géométrique de raison $1 - \lambda\Delta t$. Si $\Delta t > 2/\lambda$, la suite est alternée et divergente. Le schéma ne respecte alors ni la conservation du signe de la solution ($u(t)$ du même signe que la donnée initiale), ni la convergence en $+\infty$. Le programme va planter sur la machine ! En revanche, si Δt est suffisament petit, ce phénomène d'instabilité n'a pas lieu.
Pour ce type de problème, le recours aux schémas implicite élimine ces problèmes de stabilité conditionelle sur le pas de temps. Le schema d'Euler implicite (IV.3.2) appliqué à la même équation donne

$$u_{n+1} = \frac{1}{1 + \lambda\Delta t} u_n.$$

On a ainsi construit une suite géométrique de raison positive et strictement plus petite que 1. La suite est donc du signe de la donnée initiale et décroissante vers zéro.

DÉFINITION IV.3.5
Lorsque le schéma génère une suite bornée, on dit que le schéma est A-stable. Lorsque cette propriété est vrai pour tout Δt, on dit que le schéma est inconditionnellement A-stable.

3.3.2 Schéma conservatif pour équation conservative

On s'intéresse à nouveau aux EDO qui possède une propriété de conservation. C'est le cas de (IV.2.5) et (IV.2.6) vus précédemment. On cherche alors un schéma qui vérifie la propriété de conservation. Reprenons un exemple d'EDO sur $\mathbb{R}^2$ qui vérifie la conservation de la norme euclidienne de la solution : Soit,

$$u'(t) = f(t, u(t), v(t))v(t) \tag{IV.3.4}$$
$$v'(t) = -f(t, u(t), v(t))u(t), \tag{IV.3.5}$$
$$u(0) = u_0, \tag{IV.3.6}$$
$$v(0) = v_0. \tag{IV.3.7}$$

On a la conservation $u^2(t) + v^2(t) = u_0^2 + v_0^2$.
Le schéma d'Euler explicite appliqué à cet équation donne

$$\frac{u_{n+1} - u_n}{\Delta t} = f(t_n, u_n, v_n)v_n,$$
$$\frac{v_{n+1} - v_n}{\Delta t} = -f(t_n, u_n, v_n)u_n.$$

Par multiplication par u_n la première équation et par v_n la deuxième équation, on a

$$(u_{n+1} - u_n)u_n + (v_{n+1} - v_n)v_n = 0.$$

D'après l'inégalité,

$$ab \leq \frac{1}{2}(a^2 + b^2),$$

on montre que

$$u_n^2 + v_n^2 \leq u_{n+1}^2 + v_{n+1}^2,$$

et l'égalité n'a lieu que si $u_{n+1} = u_n$ et $v_{n+1} = v_n$. Le schéma ne conserve donc pas la norme, il l'amplifie inexorablement.
Le même calcul montre que le schéma d'Euler implicite ne conserve pas la norme, il l'écrase.
Un schéma adapté à l'EDO (IV.3.7) est le suivant :

$$\frac{u_{n+1} - u_n}{\Delta t} = f(t_n, u_n, v_n)\frac{v_n + v_{n+1}}{2},$$
$$\frac{v_{n+1} - v_n}{\Delta t} = -f(t_n, u_n, v_n)\frac{u_n + u_{n+1}}{2}.$$

En effet, en multipliant par $u_n + u_{n+1}$ la première équation et par $v_n + v_{n+1}$ la seconde, on obtient après sommation,

$$u_{n+1}^2 - u_n^2 + v_{n+1}^2 - v_n^2 = 0.$$

Le schéma ainsi construit est conservatif. De ce fait, il est également inconditionnellement A-stable.

3.4 Ordre d'un schéma numérique

Naturellement, on souhaite une erreur d'approximation la plus faible possible pour un pas de temps donné. On s'intéresse alors à la taille de l'erreur locale (*ie* l'erreur de consistance) en fonction du pas de temps. L'odre d'un schéma va caractériser la précision du schéma.

3.4.1 Définition

DÉFINITION IV.3.6
On dit qu'un schéma défini par (IV.3.3) est d'ordre p pour $p \geq 1$ si, pour $T > 0$ donné, il existe une constante réelle C telle que

$$\sum_{n=0}^{|T/\Delta t|} |u(t_{n+1}) - F(t_n, u(t_n), \Delta t)| \leq C\Delta t^p.$$

Pour qu'un schéma soit d'ordre p, il suffit que l'erreur locale soit contrôlée de la façon suivante :

$$\forall\, n \leq \frac{T}{\Delta t}, \quad |u(t_{n+1}) - F(t_n, u(t_n), \Delta t)| \leq C\Delta t^{p+1}.$$

3.4.2 Schéma d'Euler et Cranck-Nickolson

PROPOSITION IV.3.7
Les schéma d'Euler explicite (IV.3.1) et implicite (IV.3.2) sont d'ordre 1.

PROPOSITION IV.3.8
Le schéma de Crank-Nickolson défini par (IV.3.8) est d'ordre 2.

$$\frac{u_{n+1} - u_n}{\Delta t} = f(\frac{t_n + t_{n+1}}{2}, \frac{u_n + u_{n+1}}{2}). \tag{IV.3.8}$$

3.4.3 Méthode de Runge et Kutta

On construit facilement des schémas d'ordre élevé à l'aide d'une technique basée sur les formules de quadrature vues au chapitre 3. Ces ont les schémas de Runge-Kutta.
L'idée est la suivante. Pour approcher l'EDO $u'(t) = f(t, u(t))$ sur un pas de temps, on intègre l'EDO sur un pas de temps en vue d'approcher l'intégrale par une formule de quadrature :

$$u(t_{n+1}) = u(t_n) + \int_{t_n}^{t_{n+1}} f(s, u(s))ds.$$

On note que cette formule, bien qu'exacte, est totalement implicite. On fait comme si l'on connaissait u sur $[t_n, t_{n+1}]$ et on applique une formule de quadrature pour approcher l'intégrale. On obtient alors le schéma numérique :

$$u_{n+1} = u_n + \Delta t \sum_{j=1}^{q} b_j f(t_{n_j}, u_{n_j}).$$

Les coefficients $\Delta t b_j$ sont les coefficients de la formule de quadrature. Les temps intermédiaires sont définis par

$$t_{n_j} = t_n + c_j \Delta t$$

et les approximations u_{n_j} de u au temps t_{n_j} sont également donnés par une formule de quadrature :

$$u_{n_j} = u_n + \Delta t \sum_{k=1}^{q} a_{j,k} f(t_{n_k}, u_{n_k}).$$

On pourra choisir les $a_{j,k}$ de cette dernière formule de quadrature tels que $a_{j,k} = 0$ pour $k < j$ afin que le calcul de u_{n_j} soit explicite dès lors qu'il est effectué dans l'ordre croissant des j. De même, on pourra choisir les $a_{j,k}$ de cette dernière formule de quadrature tels que $a_{j,k} = 0$ pour $k > j$ afin que le calcul de u_{n_j} soit explicite dès lors qu'il est effectué dans l'ordre décroissant des j.
On représente l'ensemble des coefficients à l'aide du tableau suivant

$$\begin{array}{c|ccc} c_1 & a_{11} & \cdots & a_{1q} \\ \vdots & \cdots & \cdots & \cdots \\ c_q & a_{q1} & \cdots & a_{qq} \\ \hline & b_1 & \cdots & b_q \end{array}$$

Le schéma est explicite pour des coefficients correspondant aux tableaux

$$\begin{array}{c|ccc} c_1 & 0 & \cdots & a_{1q} \\ \vdots & 0 & \ddots & \vdots \\ c_q & 0 & 0 & 0 \\ \hline & b_1 & \cdots & b_q \end{array} \qquad \begin{array}{c|ccc} c_1 & 0 & 0 & 0 \\ \vdots & \cdots & \ddots & \vdots \\ c_q & a_{q1} & \cdots & 0 \\ \hline & b_1 & \cdots & b_q \end{array}$$

Exemple : Le schéma d'Euler explicite s'obtient par la formule de quadrature des rectangles à gauche :

$$\begin{array}{c|c} 0 & 0 \\ \hline & 1 \end{array}.$$

Le schéma d'Euler implicite s'obtient par la formule de quadrature des rectangles à droite :

$$\begin{array}{c|c} 1 & 1 \\ \hline & 1 \end{array}.$$

Le schéma connu sous le nom de Runge Kutta d'ordre 2 (RK2) s'écrit :

$$\begin{array}{c|cc} 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 \\ \hline & 0 & 1 \end{array}$$

Le schéma connu sous le nom de Runge Kutta d'ordre 4 (RK4) s'écrit :

$$\begin{array}{c|cccc} 0 & 0 & 0 & 0 & 0 \\ \frac{1}{2} & \frac{1}{2} & 0 & 0 & 0 \\ \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ \hline & \frac{1}{6} & \frac{1}{3} & \frac{1}{3} & \frac{1}{6} \end{array}$$

4 Exercices

Exercice 1.11. Soit l'équation différentielle ordinaire suivante :

$$\begin{aligned} u'(t) &= -u(t)(u(t)-1)(u(t)+1) \\ u(0) &= u_0 \end{aligned}$$

1. Montrer qu'il existe $T > 0$ et une unique solution $C^\infty(0, T)$.
2. Montrer que si $-1 \leq u_0 \leq 1$ alors $-1 \leq u(t) \leq 1$, pour $t > 0$.
3. Montrer que si $u_0 > 1$ alors u est décroissante sur $\mathbb{R}^+$.
4. Montrer que si $u_0 < -1$ alors u est croissante sur $\mathbb{R}^+$.
5. Si $u_0 > 1$, on considère le schéma semi-implicite

$$\frac{u_{n+1} - u_n}{\Delta t} = -u_n(u_{n+1} - 1)(u_n + 1).$$

   Montrer que ce schéma est A-stable si $u_0 > 0$, sans condition sur le pas de temps.
6. Construire un schéma semi-implicite pour $u_0 < 0$, afin qu'il soit inconditionnellement A-stable.

Exercice 1.12. Soit le système d'équations différentielles ordinaires suivant :

$$\begin{aligned} u'(t) &= -v(t), \\ v'(t) &= u(t), \\ u(0) &= u_0, \\ v(0) &= v_0. \end{aligned}$$

1. Montrer qu'il existe une unique solution globale à ce système. (Indication : on ajoutera les deux équations multipliées par u et v respectivement).
2. Ecrire le schéma d'Euler explicite et comparer $u_n^2 + v_n^2$ à $u_{n+1}^2 + v_{n+1}^2$.
3. Ecrire le schéma d'Euler implicite et comparer $u_n^2 + v_n^2$ à $u_{n+1}^2 + v_{n+1}^2$.
4. Soit le schéma de Cranck-Nicolson :

$$\begin{aligned} \frac{u_{n+1} - u_n}{\Delta t} &= -\frac{1}{2}(v_n + v_{n+1}), \\ \frac{v_{n+1} - v_n}{\Delta t} &= \frac{1}{2}(u_n + u_{n+1}). \end{aligned}$$

   Montrer que ce schéma vérifie la propriété du problème continue :

$$u_n^2 + v_n^2 = u_{n+1}^2 + v_{n+1}^2.$$

Exercice 1.13. Soit le système d'équations différentielles ordinaires suivant :

$$\begin{aligned} u'(t) &= -(u^2(t) + v^2(t))v(t), \\ v'(t) &= (u^2(t) + v^2(t))u(t), \\ u(0) &= u_0, \\ v(0) &= v_0. \end{aligned}$$

1. Montrer qu'il existe une unique solution globale à ce système en montrant que $u^2(t) + v^2(t)$ est constant au court du temps.
2. Ecrire un schéma qui possède la même propriété de conservation que le problème continu.

Exercice 1.14. On suppose que f est une fonction C^2 de $\mathbb{R}^+ \times \mathbb{R}^d$ à valeur $\mathbb{R}^d$. Soit l'équation différentielle ordinaire

$$\begin{aligned} u'(t) &= f(t, u(t)) \\ u(0) &= u0. \end{aligned}$$

1. Montrer qu'il existe une unique solution locale u, avec $u \in C^3(0, T, \mathbb{R}^d)$, où T est pris suffisament petit.
2. Soit le schéma d'Euler explicite

$$\begin{aligned} \frac{u_{n+1} - u_n}{\Delta t} &= f(t_n, u_n), \\ u_0 &= u0. \end{aligned}$$

   Montrer que ce schéma est d'ordre 1.
3. Soit le schéma d'Euler implicite

$$\begin{aligned} \frac{u_{n+1} - u_n}{\Delta t} &= f(t_{n+1}, u_{n+1}), \\ u_0 &= u0. \end{aligned}$$

   Montrer que ce schéma est d'ordre 1.
4. Soit le schéma de Cranck-Nicolson

$$\begin{aligned} \frac{u_{n+1} - u_n}{\Delta t} &= f(t_{n+1/2}, \frac{u_{n+1} + u_n}{2}), \\ u_0 &= u0. \end{aligned}$$

   Montrer que ce schéma est d'ordre 2.

Chapitre V

Résolution de systèmes linéaires.

1 Méthodes directes

1.1 Remontée

1.2 Méthode de Gauss et factorisation LU

1.3 Pivot de Gauss

1.4 Factorisation de Cholesky

2 Méthodes itératives

2.1 Jacobi

2.2 Gauss-Seidel

Bibliographie

[1] http://www.physique-eea.unicaen.fr/enseignement/deug-st/sm/dsm263/web/cinetique/Cinetique.html

[2] http://www.scholarpedia.org/article/Oregonator

[3] R. Théodor, Initiation à l'analyse numérique, Masson.