Analyse et analyse numérique

LAVOISIER
11, rue Lavoisier
75008 Paris

Serveur web : www.hermes-science.com

2-7462-0993-4 ISBN Général

2-7462-0995-0 ISBN Volume 2

APPLICATIONS MATHÉMATIQUES
AVEC MATLAB®

Analyse et analyse numérique

rappel de cours et exercices corrigés

Luc Jolivet
Rabah Labbas

Hermes Science
publications

Table des matières

Avant-propos

Le but de cette collection "Applications Mathématiques avec *Matlab*" est de comprendre et d'utiliser les outils mathématiques fondamentaux de premier cycle à l'aide d'un logiciel de calcul. Elle correspond à l'esprit des formations en IUT, BTS, Ecoles d'ingénieurs, mais aussi en premiers semestres du cycle L du nouveau schéma LMD.

Nous nous sommes basés sur l'expérience de nos cours, travaux dirigés et séances de travaux pratiques de mathématiques avec des étudiants de 1ère et 2ème année du département d'Informatique d'IUT de l'Université du Havre. Pour cet enseignement, nous disposons du logiciel *Matlab*[1](la version actuellement installée est 6.5.0) et de son extension *Symbolic Math Toolbox* (version 2.1.3).

Ces outils nous ont permis d'accompagner les notions de base présentées, par des illustrations numériques et graphiques, et par des vérifications utilisant le calcul formel.

L'utilisation d'un logiciel de calcul permet de se concentrer davantage sur la compréhension du problème posé, sur une stratégie de résolution et sur l'interprétation des résultats. L'étudiant devra aussi porter un regard critique sur les réponses fournies, en prenant garde aux erreurs d'arrondi dans les calculs numériques, et aux simplifications abusives dans certaines expressions symboliques.

Ce second tome comprend deux parties.

En première partie, on présente les notions de base et les principaux théorèmes de l'analyse (suites, fonctions numériques d'une variable réelle, intégration), le plus souvent sans démonstrations, pour les utiliser principalement dans des applications et calculs concrets.

---

1. *Matlab* est une marque déposée de *The MathWorks Inc.* Tous les autres produits cités sont des marques déposées de leur société respective.

La deuxième partie est consacrée à l'arithmétique des ordinateurs, à quelques outils de base en analyse numérique et à la gestion d'erreurs. Le but étant d'initier les étudiants à résoudre numériquement quelques problèmes.

Chaque partie est composée de chapitres. Ils sont accompagnés d'illustrations et d'exemples traités avec *Matlab*. Des exercices sont ensuite proposés. Certains sont originaux, d'autres sont repris ou inspirés de divers manuels dont la liste est donnée en bibliographie. La correction de ces exercices se trouve en fin de chapitre. Nous avons choisi de la présenter en utilisant systématiquement *Matlab*. Le lecteur pourra cependant traiter la plupart de ces exercices "à la main".

Lorsqu'une commande *Matlab* est utilisée pour la première fois, elle est expliquée et apparaît en gras. Les programmes et séquences de calcul sous *Matlab* sont mis en évidence dans des tableaux. Le lecteur pourra trouver une initiation à la pratique de ce logiciel dans les premiers chapitres du tome 1.

En fin d'ouvrage, se trouve un index regroupant les mots-clés mathématiques et les commandes *Matlab* utilisées. Ces dernières apparaissent en italique.

Nous tenons à remercier vivement tous nos collègues qui ont consacré un temps précieux à la lecture de cet ouvrage, notamment Serge Derible, Thierry Dumont, Khaled Sadallah et Francis Wirth.

Nous remercions particulièrement François Coquet, Professeur à l'Université du Havre, pour sa lecture attentive, ses remarques et conseils judicieux .

Nous accueillerons avec reconnaissance les éventuelles remarques que le lecteur voudra bien nous faire parvenir.

Note au lecteur

Ce recueil de rappels de cours et d'exercices corrigés fait partie d'un ensemble comportant trois tomes.

Tome 1

– première partie : présentation de *Matlab,*
– deuxième partie : algèbre linéaire,
– troisième partie : géométrie.

Tome 2

– première partie : analyse,
– deuxième partie : analyse numérique élémentaire.

Tome 3

– théorie élémentaire du signal.

PREMIÈRE PARTIE

Analyse

Chapitre 1

Suites réelles

Dans ce chapitre, on rappelle l'essentiel concernant les suites réelles et on termine par une étude complète d'une suite récurrente avec *Matlab*.

1.1. Généralités sur les suites

1.1.1. *Définitions*

On appelle suite numérique une application définie par

$$\begin{array}{rcl} u : \mathbb{N} & \longrightarrow & \mathbb{R} \\ n & \longmapsto & u(n). \end{array}$$

$u(n)$ est le **terme général** de la suite et est souvent noté u_n. Par abus de langage, la suite u qui est déterminée par ses valeurs $u(n) = u_n$, se note

$$(u_n)_{n\geqslant 0}\,.$$

Une suite peut être définie à partir d'un certain rang p fixé. On notera

$$(u_n)_{n\geqslant p}\,.$$

Une suite $(u_n)_{n\geqslant 0}$ est **majorée** s'il existe une constante M telle que

$$\forall n \geqslant 0 \quad u_n \leqslant M,$$

elle est **minorée** s'il existe une constante m telle que

$$\forall n \geqslant 0 \quad u_n \geqslant m.$$

Elle est **bornée** s'il existe une constante $C > 0$ telle que

$$\forall n \geqslant 0 \quad |u_n| \leqslant C.$$

Il est facile de vérifier qu'une suite est bornée si, et seulement si, elle est majorée et minorée.

1.1.2. *Exemple*

Considérons la suite définie pour $n \geqslant 2$, par

$$u_n = \frac{\sin(n)}{n + (-1)^n}.$$

Sachant que pour tout n, on a

$$-1 \leqslant \sin(n) \leqslant 1,$$

et que pour $n \geqslant 2$

$$n + (-1)^n \geqslant 2 - 1 = 1,$$

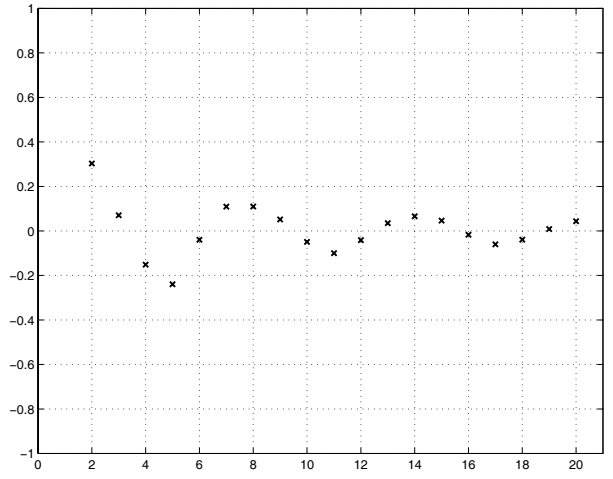
on déduit que, pour tout $n \geqslant 2$,

$$|u_n| \leqslant 1.$$

On peut avec *Matlab* observer numériquement et graphiquement cette majoration, pour les 19 premiers termes de la suite.

```
» for n=2 :20,
   u(n)=sin(n)/(n+(-1)^n);
end
u(2 :20)
ans =
0.3031 0.0706 -0.1514 -0.2397 -0.0399 0.1095 0.1099
0.0515 -0.0495 -0.1000 -0.0413 0.0350 0.0660 0.0464
-0.0169 -0.0601 -0.0395 0.0083 0.0435
```

```
» plot(2 :20,u(2 :20),'x')
» grid on ; axis([0 21 -1 1]) ;
```



1.2. Limite d'une suite

1.2.1. *Approche intuitive*

Lorsque n devient de plus en plus grand ($n \longrightarrow +\infty$), il existe quatre possibilités pour les termes d'une suite numérique donnée $(u_n)_{n\geqslant 0}$:

1) u_n "s'approche" d'un nombre l, on notera :

$$\lim u_n = l \quad \text{ou} \quad u_n \longrightarrow l,$$

2) u_n devient "aussi grand qu'on veut", on écrira

$$\lim u_n = +\infty,$$

3) u_n devient "aussi petit qu'on veut", on écrira

$$\lim u_n = -\infty,$$

4) u_n n'a aucun des comportements ci-dessus. C'est par exemple le cas de la suite

$$u_n = (-1)^n$$

qui prend alternativement les valeurs $+1$ et -1.

1.2.2. *Cas de limite finie*

1.2.2.1. *Définitions*

La description mathématique de la première possibilité doit **traduire** le fait qu'à partir d'un rang n_0, les termes successifs de la suite

$$u_{n_0},\ u_{n_0+1},\ u_{n_0+2}, \cdots$$

sont "aussi proches qu'on veut" de l. On écrira alors :

> *étant donné un nombre* $\varepsilon > 0$, *(aussi petit que l'on veut)*
> *il existe un rang* n_0 *tel que pour tout* $n \geqslant n_0$, *on a :*
> $l - \varepsilon \leqslant u_n \leqslant l + \varepsilon$.

On dira que la suite est **convergente** et converge vers l. On notera

$$\lim u_n = l.$$

On montre, grâce à cette définition, qu'une telle limite, si elle existe, est unique.

Dans les autres cas on dira que la suite est **divergente**.

1.2.2.2. *Suites de référence*

Les suites définies pour $n > 0$, de la forme

$$a_n = \frac{1}{n^\alpha},$$

où α est un réel positif, sont appelées suites de référence. Elles vérifient :

$$\lim a_n = 0.$$

Il est souvent commode de les utiliser pour montrer qu'une suite $(u_n)_{n\geqslant 0}$ converge vers une limite finie l. Lorsque la suite $(u_n)_{n\geqslant 0}$ vérifie : il existe une constante $C > 0$ et un entier $n_0 \in \mathbb{N}$ tels que pour tout $n \geqslant n_0$ on ait

$$|u_n - l| \leqslant \frac{C}{n^\alpha},$$

alors $\lim u_n = l$.

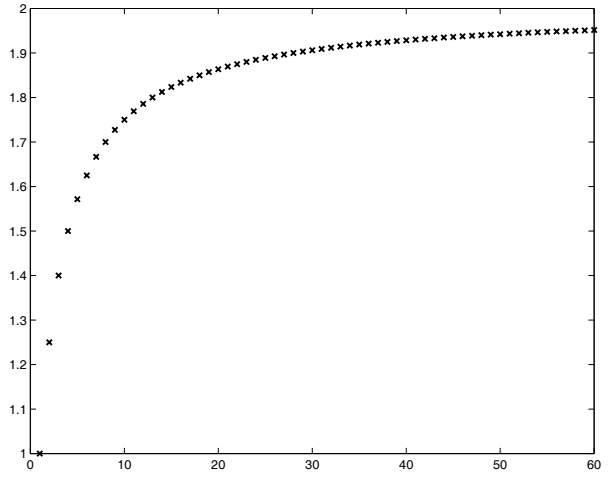
1.2.2.3. *Exemple*

Examinons l'exemple de la suite définie pour $n \geqslant 3$ par :

$$u_n = \frac{2n+1}{n-2}.$$

On a, sous *Matlab,*

```
» for n=1 :60,
      u(n)=(2*n+1)/(n+2);
end
» u(50 :60)
ans =
Columns 1 through 7
1.9423 1.9434 1.9444 1.9455 1.9464 1.9474 1.9483
Columns 8 through 11
1.9492 1.9500 1.9508 1.9516
» plot(u,'x')
```



Cela laisse présager que la suite converge vers 2. Pour montrer ce résultat , on peut utiliser la suite de **référence**

$$a_n = 1/n$$

(ou toute autre suite de la forme $a_n = 1/(n^\alpha)$, $\alpha > 0$) et montrer qu'il existe une constante positive C telle que, à partir d'un certain rang n_0, on ait

$$n \geqslant n_0 \Longrightarrow \left| \frac{2n+1}{n+2} - 2 \right| \leqslant C\frac{1}{n}.$$

Or

$$\left| \frac{2n+1}{n+2} - 2 \right| = \frac{3}{n+2},$$

Il suffit de prendre $C = 3$ et $n_0 = 0$

1.2.3. *Cas de limite infinie*

La deuxième possibilité s'écrit mathématiquement sous la forme :

étant donné un nombre $A > 0$, *(aussi grand que l'on veut)*
il existe un rang n_0 *tel que pour tout* $n \geqslant n_0$, *on a :*
$u_n \geqslant A$.

On écrit

$$\lim u_n = +\infty.$$

De manière similaire, la troisième possibilité s'énonce par

étant donné un nombre $A < 0$, *(aussi petit que l'on veut)*
il existe un rang n_0 *tel que pour tout* $n \geqslant n_0$, *on a :*
$u_n \leqslant A$.

On écrit

$$\lim u_n = -\infty.$$

1.3. Propriétés des limites de suites

1.3.1. *Cas de limites finies*

On se donne deux suites $(u_n)_{n\geqslant 0}$ et $(v_n)_{n\geqslant 0}$ convergentes respectivement vers des limites finies l et l'.

Alors on a :

- $u_n + v_n \longrightarrow l + l'$,
- $\lambda u_n \longrightarrow \lambda l$ *(pour tout nombre* λ *)*
- $u_n.v_n \longrightarrow l.l'$,
- $\dfrac{u_n}{v_n} \longrightarrow \dfrac{l}{l'}$, *(si* $l' \neq 0$*),*

On a aussi le résultat suivant :

- *Si la suite* $(u_n)_{n\geqslant 0}$ *est bornée et si* $v_n \longrightarrow 0$, *alors* $u_n.v_n \longrightarrow 0$.

Sa démonstration utilise l'encadrement dit des "gendarmes" :

il existe $C > 0$ tel que pour tout $n \in \mathbb{N}$, on a

$$-C \leqslant u_n \leqslant C,$$

d'autre part, pour ε petit donné strictement positif, en notant

$$\varepsilon' = \frac{C}{\varepsilon},$$

il existe n_0 tel que pour $n \geqslant n_0$, on a

$$-\varepsilon' \leqslant v_n \leqslant +\varepsilon',$$

d'où l'encadrement

$$-C\varepsilon' \leqslant u_n v_n \leqslant C\varepsilon',$$

ou bien

$$-\varepsilon \leqslant u_n v_n \leqslant \varepsilon,$$

pour $n \geqslant n_0$, ce qui montre que $u_n v_n \to 0$.

1.3.2. *Cas de limites infinies*

Les résultats précédents s'étendent aux cas de limites infinies :

- *si* $\lim u_n = +\infty$ *et* $\lim v_n = +\infty$ *alors* $\lim(u_n + v_n) = +\infty$,
- *si* $\lim u_n = +\infty$ *et* $\lim v_n = l$, $l \in \mathbb{R}$ *alors* $\lim(u_n + v_n) = +\infty$,
- *si* $\lim u_n = -\infty$ *et* $\lim v_n = -\infty$ *alors* $\lim(u_n + v_n) = -\infty$,
- *si* $\lim u_n = -\infty$ *et* $\lim v_n = l$, $l \in \mathbb{R}$ *alors* $\lim(u_n + v_n) = -\infty$,
- *si* $\lim u_n = +\infty$ *et* $\lim v_n = +\infty$ *alors* $\lim(u_n.v_n) = +\infty$,
- *si* $\lim u_n = +\infty$ *et* $\lim v_n = -\infty$ *alors* $\lim(u_n.v_n) = -\infty$,
- *si* $\lim u_n = -\infty$ *et* $\lim v_n = -\infty$ *alors* $\lim(u_n.v_n) = +\infty$,
- *si* $\lim u_n = +\infty$ *et* $\lim v_n = l$, $l \in \mathbb{R}^*$ *alors* $\lim(u_n.v_n) = +\infty$ *(si* $l > 0$*)* *et* $-\infty$ *(si* $l < 0$*)*.

Par contre, toute recherche de limite qui se présente sous l'une des formes suivantes est une indétermination :

$$+\infty - \infty, \quad 0 \times \infty, \quad \infty/\infty, \quad 0/0, \quad 1^\infty, \quad 0^0, \quad \infty^0.$$

L'indétermination est à lever en appronfondissant les calculs sur l'expression donnée.

1.3.3. *Calculs de limites avec Matlab*

On utilise la commande **limit** pour obtenir directement la limite d'une suite (u_n). Par exemple, si

$$u_n = \frac{2n+1}{n+6}, \qquad v_n = \frac{2\left|n^2 - 20\right| + 1}{|n - 10| + 7}$$

on a

```
» syms n ; Un=(2*n+1)/(n+6) ;
» limit(Un,n,inf)
ans = 2
```

```
» Vn=(2*abs(n^2-20)+1)/(abs(n-10)+7);
» limit(Vn,n,inf)
ans = inf
```

Matlab permet aussi de retrouver les calculs sur les limites infinies :

```
» inf+inf
ans=Inf
» inf*inf
ans=Inf
» inf*(-inf)
ans=-Inf
» 0*inf
ans=NaN
```

la dernière réponse exprime l'indétermination : Not a Number.

1.4. Suites monotones

Une suite réelle $(u_n)_{n\geqslant 0}$ est dite croissante si

$$\forall n \geqslant 0 \quad u_{n+1} - u_n \geqslant 0,$$

elle est décroissante si

$$\forall n \geqslant 0 \quad u_{n+1} - u_n \leqslant 0.$$

La suite est strictement croissante si

$$\forall n \geqslant 0 \quad u_{n+1} - u_n > 0,$$

strictement décroissante si

$$\forall n \geqslant 0 \quad u_{n+1} - u_n < 0.$$

Dans chacun de ces cas on parlera de suite monotone (resp. strictement monotone).

Ces notions restent valables si elles sont vraies à partir d'un rang donné $n_0 > 0$. Le résultat essentiel pour les suites monotones est :

Théorème.
- *Toute suite réelle croissante et majorée est convergente.*
- *Toute suite réelle décroissante et minorée est convergente.*

1.5. Suites récurrentes

1.5.1. *Définition*

On donne une fonction $f : \mathbb{R} \to \mathbb{R}$ et on définit la suite $(u_n)_{n\geqslant 0}$ par la relation de récurrence :

$$\begin{cases} u_0 = a \\ u_{n+1} = f(u_n), \quad \text{pour } n \geqslant 0, \end{cases}$$

où a est réel donné.

De telles suites sont dites **récurrentes**. Elles sont bien définies lorsque pour tout $n \geqslant 0$, les termes u_n appartiennent au domaine de définition de f.

Le calcul de la limite l de telles suites récurrentes, lorsqu'elles convergent, se fait par passage à la limite dans la relation de récurrence :

1) on peut vérifier que si $\lim u_n = l$, alors $\lim u_{n+1} = l$, (en utilisant la définition),

2) lorsque la fonction f est **continue**, on a : $\lim f(u_n) = f(\lim u_n) = f(l)$,

3) on en déduit l'équation

$$l = f(l).$$

Si elle admet une solution, ou plusieurs, une étude de la suite (monotonie, minoration ou majoration,...) permet de préciser la limite éventuelle.

1.5.2. *Etude complète d'un exemple modèle*

On donne la suite définie pour $n \in \mathbb{N}^*$ par

$$\begin{cases} u_1 = 1 \\ u_{n+1} = \sqrt{1 + u_n}. \end{cases}$$

En réitérant on voit que le terme général s'écrit

$$u_n = \sqrt{1 + \sqrt{1 + \ldots \sqrt{1 + 1}}}.$$

1.5.2.1. *Calcul des premiers termes de la suite*

La suite est évidemment minorée par 1.

On calcule, sous *Matlab,* les dix premiers termes de cette suite, qu'on range dans un tableau u :

```
» u(1)=1 ;
» for n=1 :9,u(n+1)=sqrt(1+u(n));end
» u(1 :5)
1.0000 1.4142 1.5538 1.5981 1.6118
» u(6 :10)
1.6161 1.6174  1.6179  1.6180  1.6180
```

Cela suggère que la suite est majorée par 2. Montrons-le par récurrence. On a

$$u_1 = 1 \leqslant 2.$$

D'autre part, la résolution de l'inéquation

$$\sqrt{1+u_n} \leqslant 2$$

donne

```
» maple('solve(sqrt(1+Un)-2<=0)')
ans=RealRange(-1,3)
```

D'où l'implication

$$u_n \in [-1, 3] \Longrightarrow u_{n+1} - 2 \leqslant 0.$$

Ainsi, l'hypothèse de récurrence $u_n \leqslant 2$ et le fait que u_n est minoré par 1, donnent $u_{n+1} \leqslant 2$.

1.5.2.2. *Calcul de la limite éventuelle*

La fonction définie par $f(x) = \sqrt{1+x}$ étant continue sur $[0, +\infty[$, la limite l, si elle existe, vérifie nécessairement l'équation $f(l) = l$. On la calcule sous *Matlab :*

```
» syms x
» l= solve('x=sqrt(1+x)')
l =1/2+1/2*5^(1/2)
```

Si la suite (u_n) converge, alors sa limite est égale au nombre (dit d'or)

$$l = \frac{1}{2} + \frac{\sqrt{5}}{2}.$$

On va maintenant prouver la convergence, en montrant que la suite (u_n) est croissante et majorée. On sait déjà que la suite est majorée par 2. Mais dans l'étude de la monotonie, on aura besoin de la majoration

$$u_n \leqslant l.$$

Etablissons cette propriété :

1.5.2.3. *Majoration de la suite par* 1

On raisonne par récurrence. Il est clair que la suite est positive et que

$$u_1 \leqslant l.$$

On étudie le signe de $u_{n+1} - l$:

```
» maple('solve(sqrt(1+Un)-1/2-sqrt(5)/2<=0)')
ans = RealRange(-1,1/2+1/2*5^(1/2))
```

Ainsi, si on suppose $0 \leqslant u_n \leqslant l$, alors $u_{n+1} - l \leqslant 0$, d'où

$$\forall n \in \mathbb{N}^* \quad 0 \leqslant u_n \leqslant l.$$

1.5.2.4. *Sens de variation*

Pour étudier le signe de $u_{n+1} - u_n$, on étudie celui de l'expression

$$(u_{n+1} - u_n)\,(u_{n+1} + u_n)\,.$$

Pour cela, montrons que

$$(u_{n+1} - u_n)\,(u_{n+1} + u_n) = -(u_n - l)(u_n - l'),$$

où

$$l' = \frac{1}{2} - \frac{\sqrt{5}}{2}.$$

On a :

```
» syms Un
» UnPlus1 = sqrt(1+Un);
» P = expand((UnPlus1-Un)*(UnPlus1+Un))
P = 1+Un-Un^2
» % On cherche les racines de P pour factoriser :
» S = solve(P)
S = [ 1/2+1/2*5^(1/2)]
    [ 1/2-1/2*5^(1/2)]
» Q=(Un-S(1))*(Un-S(2)); expand(Q)
   ans = -1-Un+Un^2
```

On en déduit que pour tout $n \geqslant 1$,

$$u_{n+1} - u_n = \frac{-(u_n - l)(u_n - l')}{(u_{n+1} + u_n)} \geqslant 0,$$

car

$$u_n - l \leqslant 0$$

et

$$u_n - l' \geqslant 0,$$

puisque $u_n \geqslant 0$ et $l\prime < 0$.

La suite étant croissante et majorée, elle est donc convergente et sa limite est bien l.

1.6. Exercices

1.6.1. *Limite d'une suite et majorations*

On se propose de calculer la limite de la suite définie par

$$u_n = \frac{\sqrt{n}}{\sqrt{n}-5}, \qquad n \geqslant 26$$

en utilisant la définition.

1) Peut-on se servir de la lecture du graphe de la suite pour en donner l'éventuelle limite ?

2) En calculant

$$\left| \frac{\sqrt{n}}{\sqrt{n}-5} - 1 \right|,$$

résoudre mathématiquement l'inéquation

$$\left| \frac{\sqrt{n}}{\sqrt{n}-5} - 1 \right| \leqslant \frac{10}{\sqrt{n}}.$$

3) Conclure.

4) Retrouver par $Matlab$ tous les résultats de la question 2.

(solution p. 29)

1.6.2. *Etude d'une suite récurrente (1)*

La direction d'un journal constate pour chaque année un taux de réabonnement voisin de 60%, ainsi que l'apparition de 5000 nouveaux abonnés.

On note a_n le nombre d'abonnés l'année numérotée n. L'année numéro 1, le nombre d'abonnés est $a_1 = 10000$.

1) Donner la relation de récurrence entre a_{n+1} et a_n.

2) Calculer les 20 premiers termes de cette suite et les représenter graphiquement.

3) Vers quelle limite l cette suite semble-t-elle converger ?

4) On considère la suite définie par $u_n = l - a_n$. Utiliser le calcul symbolique de *Matlab* pour montrer que (u_n) est une suite géométrique

5) En déduire l'expression de u_n, puis celle de a_n.

6) Confirmer le résultat expérimental de la question 3.

(solution p. 31)

1.6.3. *Etude d'une suite récurrente (2)*

On définit la suite $(u_n)_{n\geqslant 1}$ par la relation de récurrence

$$\begin{cases} u_0 = 0 \\ u_{n+1} = \dfrac{1}{5}u_n + 1 \quad \text{pour } n \geqslant 0, \end{cases}$$

1) Calculer u_n pour $n = 1, 2, ..., 20$ et en donner une représentation graphique.

2) Quelle est l'éventuelle limite l de cette suite ?

3) Montrer avec *Matlab* que la suite est majorée par $5/4$.

4) En déduire qu'elle est croissante.

5) Vérifier que $u_{n+1} - l = \frac{1}{5}(u_n - l)$.

6) En déduire que $u_n - l = \frac{-5}{4}\left(\frac{1}{5}\right)^n$.

7) Trouver un rang n_0 tel que : $\quad n \geqslant n_0 \Longrightarrow |u_n - l| \leqslant 10^{-7}$.

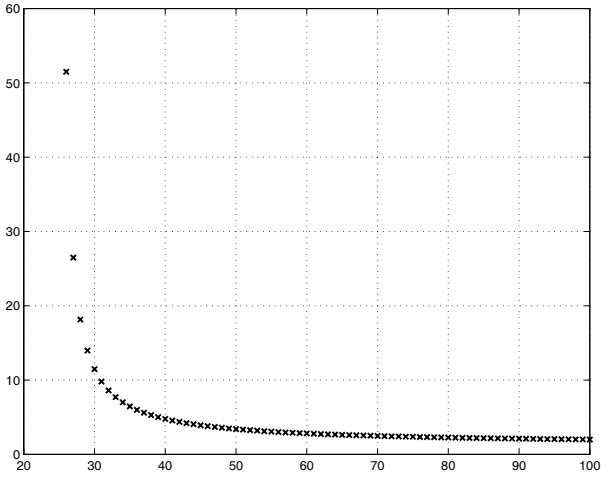
(solution p. 33)

1.7. Solutions

Exercice 1.6.1

1) On représente graphiquement les nombres u_n, pour $n \in [26, 100]$.

```
» clf
» n=26 :1 :100 ;
» u=sqrt(n)./(sqrt(n)-5) ;
» plot(n,u,'x')
» grid on
```



Le dessin laisse à penser que la suite est décroissante, minorée, mais n'indique pas clairement la limite. Le calcul des termes u_{100}, u_{1000}, u_{10000} laisse présager que la limite est 1 :

```
» n=[100 1000 10000];
» u= sqrt(n)./(sqrt(n)-5)
u = 2.0000 1.1878 1.0526
```

2) On a, pour $n \geqslant 26$

$$\left| \frac{\sqrt{n}}{\sqrt{n}-5} - 1 \right| = \frac{5}{\sqrt{n}-5}.$$

Et l'inéquation

$$\left| \frac{\sqrt{n}}{\sqrt{n}-5} - 1 \right| \leqslant \frac{10}{\sqrt{n}}$$

est équivalente à

$$\frac{5}{\sqrt{n}-5} \leqslant \frac{10}{\sqrt{n}},$$

soit à

$$5\sqrt{n} \leqslant 10(\sqrt{n}-5),$$

ou encore à

$$-5\sqrt{n} \leqslant -50,$$

d'où

$$\sqrt{n} \geqslant 10,$$

et finalement $n \geqslant 100$.

3) Pour $n \geqslant 100$, on peut majorer $|u_n - 1|$ par $10/\sqrt{n}$, qui est une suite de référence (voir § 1.2.2.3). Cette majoration montre que la suite (u_n) converge vers 1.

4) On utilise le calcul symbolique de *Matlab* pour calculer $|u_n - 1|$, puis pour résoudre l'inéquation

$$|u_n - 1| \leqslant \frac{10}{\sqrt{n}}.$$

```
» syms n
» un=sqrt(n)/(sqrt(n)-5);
» simplify(abs(un-1))
ans = 5/abs(n^(1/2)-5)
» maple('solve(5/abs(n^(1/2)-5)<=10/n^(1/2))')
ans = RealRange(100,inf),RealRange(Open(0),100/9)
```

L'ensemble solution de cette inéquation $]0, 100/9] \cup [100, +\infty[$ montre que l'inégalité

$$|u_n - 1| \leqslant \frac{10}{\sqrt{n}}$$

est vérifiée pour $n \geqslant 100$.

En utilisant la commande *limit* de *Matlab*, on obtient directement la limite de la suite (u_n) :

```
» limit(un,n,inf)
ans = 1
```

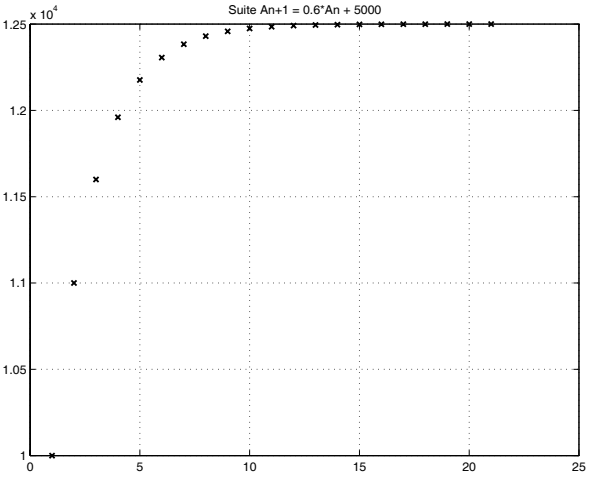
Exercice 1.6.2

1) On a la formule de récurrence

$$a_{n+1} = 0.6a_n + 5000.$$

2) On calcule les 20 premiers termes de cette suite et on les représente graphiquement

```
» a(1)=10000;
» for n=1 :20,
   a(n+1)=0.6*a(n)+5000;
end
» plot(a, 'x')
» title('Suite An+1 = 0.6*An + 5000')
» grid on
» a(1 :5)  % 5 premiers termes
ans = 10000 11000 11600 11960 12176
» a(16 :20)  % 5 derniers
ans = 12499 12499 12500 12500 12500
```



3) Cette suite semble converger vers la valeur $l = 12500$.

4) On définit en fonction de a_n les expressions de u_n, a_{n+1}, u_{n+1} :

```
» syms An
» AnPlus1 = 0.6*An+5000;
» Un= 12500-An;
» UnPlus1= 12500-AnPlus1
UnPlus1 = 7500-3/5*An
» simplify(UnPlus1/Un)
ans = 3/5
```

De la dernière égalité, on déduit que la suite (u_n) est une suite géométrique de raison $3/5$.

5) A partir du premier terme $u_1 = 12500 - a_1$, on déduit l'expression générale

$$u_n = u_1 \times \left(\frac{3}{5}\right)^{n-1},$$

puis

$$a_n = 12500 - u_n.$$

```
» U1 = subs(Un,An,10000)
U1 =2500
» syms n ; Un = U1*(3/5)^(n-1);
» An = 12500-Un
An =12500-2500*(3/5)^(n-1)
```

On a, pour tout $n \geqslant 1$,

$$a_n = 12500 - 2500 * (0.6)^{n-1}.$$

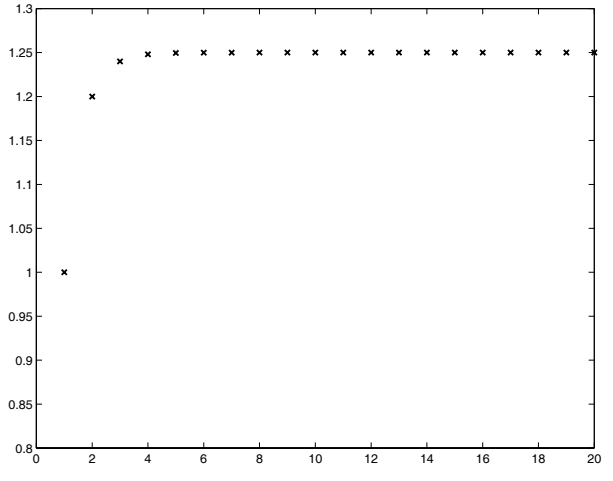
6) Comme $\lim (0.6)^{n-1} = 0$, on déduit que

$$\lim a_n = 12500 = l.$$

Exercice 1.6.3

1) On place dans le tableau u les 20 premiers termes de la suite, et on utilise *plot* pour représenter graphiquement cette suite.

```
» clear
» u(1)=1 ;
» for n=1 :19, u(n+1)= 1/5*u(n)+1 ;end
» u
u =
Columns 1 through 7
1.0000 1.2000 1.2400 1.2480 1.2496 1.2499 1.2500
Columns 8 through 14
1.2500 1.2500 1.2500 1.2500 1.2500 1.2500 1.2500
Columns 15 through 20
1.2500 1.2500 1.2500 1.2500 1.2500 1.2500
» plot(u,'x')
» hold on ; axis([0 20 0.8 1.3])
```



Au vu de ce dessin, il semble que la suite soit croissante, majorée, convergente vers $1,25$. Les questions suivantes vont prouver que ces observations sont fondées.

2) On sait que, pour une suite (u_n) vérifiant la relation de récurrence

$$u_{n+1} = f\left(u_n\right),$$

avec f continue sur $\mathbb{R}$, sa limite l, si elle existe, vérifie nécessairement l'équation

$$l = f(l),$$

soit ici

$$l = (1/5)l + 1,$$

d'où

$$l = 5/4.$$

Si la suite converge, sa limite est nécessairement $5/4$.

3) On utilise la relation de récurrence

$$u_{n+1} = \frac{1}{5}u_n + 1,$$

et on résout l'inéquation, d'inconnue u_n,

$$\frac{1}{5}u_n + 1 \leqslant \frac{5}{4}.$$

```
» syms Un
» maple('solve(1/5*Un+1<=5/4)')
ans =RealRange(-inf,5/4)
```

Ainsi, si $u_n \leqslant 5/4$, alors $u_{n+1} \leqslant 5/4$. Comme par hypothèse $u_0 \leqslant 5/4$, la majoration demandée est démontrée par récurrence.

4) On résout l'inéquation $u_{n+1} - u_n \geqslant 0$

```
» UnPlus1=1/5*Un+1 ;
» UnPlus1-Un
ans =-4/5*Un+1
» maple('solve(-4/5*Un+1>=0)')
ans =RealRange(-inf,5/4)
```

D'après la question précédente, on a, pour tout n,

$$u_n \leqslant 5/4$$

d'où $u_{n+1} - u_n \geqslant 0$.

5) On compare $d = u_{n+1} - l$, avec $p = (1/5)(u_n - l)$:

```
» l=5/4 ;
» d=UnPlus1-l
d =1/5*Un-1/4
» p=1/5*(Un-l)
p =1/5*Un-1/4
```

cela montre l'égalité demandée.

6) On en déduit par récurrence

$$u_n - l = \left(\frac{1}{5}\right)^n (u_0 - l) = \left(\frac{1}{5}\right)^n \left(\frac{-5}{4}\right).$$

7) Si on utilise l'expression ci-dessus pour résoudre l'inéquation

$$|u_n - l| \leqslant 10^{-7},$$

on obtient

```
» maple('solve(5/4*(1/5)^n<=10^(-7))')
ans =' '
```

ce qui signifie que l'ensemble solution n'a pu être obtenu par *Maple*.

On transforme l'inéquation

$$\left(\frac{1}{5}\right)^n \left(\frac{5}{4}\right) \leqslant 10^{-7}$$

en l'inéquation équivalente

$$\left(\frac{1}{5}\right)^n \leqslant \frac{4}{5}10^{-7}$$

puis en

$$5^n \geqslant \frac{5}{4}10^7.$$

On obtient alors

```
» maple('solve(5^n >=5/4*10^7)')
ans = RealRange(log(12500000)/log(5),inf)
» double(log(12500000)/log(5))
ans=10.1534
```

Ainsi, pour $n \geqslant 11$, $|u_n - l| \leqslant 10^{-7}$.

Chapitre 2

Fonctions numériques d'une variable réelle

Ce chapitre est consacré à l'étude des fonctions d'une variable réelle définies sur un intervalle I de $\mathbb{R}$ fini ou infini.

2.1. Rappels généraux sur les fonctions

2.1.1. *Majoration d'une fonction et extrema*

Comme pour les suites, on dira que :

1) f est **majorée** sur I s'il existe une constante M telle que

$$\forall x \in I \quad f(x) \leqslant M.$$

2) Elle est **minorée** s'il existe une constante m telle que

$$\forall x \in I \quad f(x) \geqslant m.$$

On dira qu'elle est **bornée** s'il existe une constante $C > 0$ telle que

$$\forall x \in I \quad |f(x)| \leqslant C.$$

On peut vérifier qu'une fonction est bornée si, et seulement si, elle est majorée et minorée.

On rappelle aussi qu'en un point $a \in I$, la fonction admet un **minimum** si

$$\forall x \in I \quad f(x) \geqslant f(a),$$

et qu'en un point $b \in I$, elle a un **maximum** si

$$\forall x \in I \quad f(x) \leqslant f(b);$$

en chacun de ces points, on dira qu'il y a un extremum.

2.1.2. *Exemple*

Soit f la fonction définie sur l'intervalle $I = [0, +\infty[$ par

$$f(x) = \frac{1}{1+x^2}.$$

On a

$$\forall x \in I \quad f(x) \leqslant 1.$$

et

$$\forall x \in I \quad f(x) \geqslant 0.$$

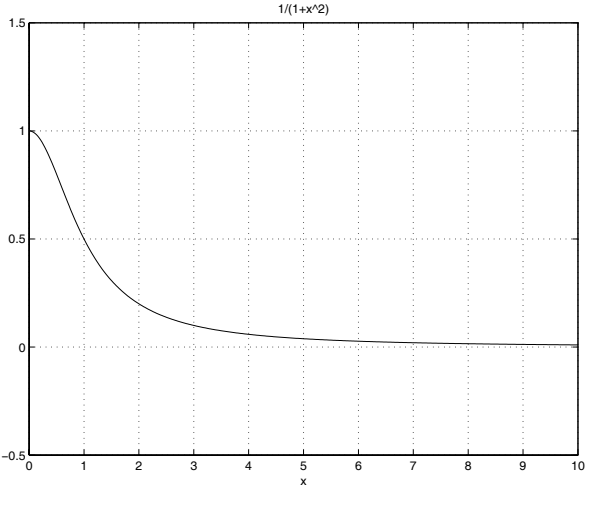
On remarque que $1 = f(0)$, donc f admet un maximum au point

$$a = 0.$$

Mais il n'existe pas de point b tel que $f(b) = 0$.

Graphiquement, on a :

```
» syms x real ; f=1/(1+x^2) ;
» ezplot(f,0,10) ; grid on
» axis([0 10 -0.5 1.5])
```



2.1.3. *Périodicité, parité et imparité d'une fonction*

Une fonction définie sur $\mathbb{R}$ est dite **périodique** de période T si on a

$$\forall x \in \mathbb{R} \quad f(x+T) = f(x).$$

Pour de telles fonctions, il suffira de les étudier sur un intervalle de longueur T.

Lorsqu'une fonction f est définie sur un intervalle centré $I =]-a, +a[$ (avec éventuellement $a = +\infty$), on dira qu'elle est **paire** si

$$\forall x \in I \quad f(-x) = f(x).$$

Si f est paire, alors son graphe admet une symétrie par rapport à l'axe des ordonnées et donc il suffira de l'étudier sur $[0, a[$.

De même, une fonction f définie sur $]-a, +a[$ (avec éventuellement $a = +\infty$) est dite **impaire** si

$$\forall x \in I \quad f(-x) = -f(x),$$

son graphe admet alors une symétrie par rapport à l'origine du repère. Il suffira d'étudier la fonction sur $[0, a[$.

2.1.4. *Exemple*

La fonction cosinus, définie sur $I = \mathbb{R}$ est paire et périodique, de période 2π. Sous *Matlab* :

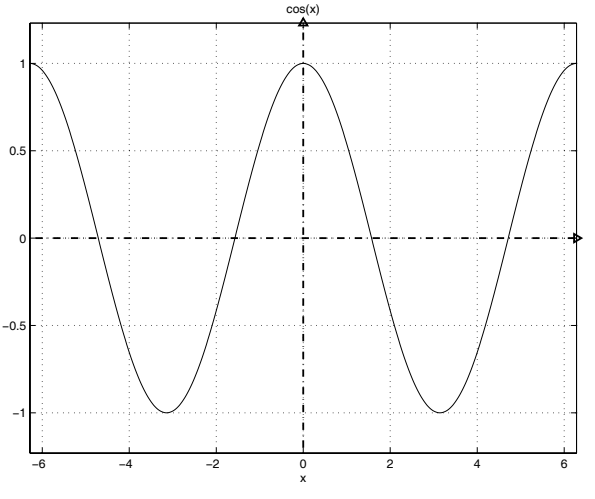
```
» syms x real
» f=cos(x) ;
» simplify(cos(-x))
ans = cos(x)
»simplify(cos(x+2*pi))
ans = cos(x)
```

Pour la représentation graphique, on utilise la fonction *dessineRepere* permettant de faire apparaître les axes du repère en traits mixtes.

```
function dessineRepere
hold on
V=axis ;
% V contient les abscisses et ordonnées minimum et maximum
Xmin=V(1) ;Xmax=V(2) ;Ymin=V(3) ;Ymax=V(4) ;
plot([Xmin Xmax],[0 0],'-.','LineWidth',1.5)
% LineWidth permet de choisir la largeur du trait (1 par défaut)
plot(Xmax,0,'>','LineWidth',1.5) %dessine la flèche horizontale
plot([0 0],[Ymin Ymax],'-.','LineWidth',1.5)
plot(0,Ymax,'^','LineWidth',1.5)
```

On utilise la commande *dessineRepere* à la suite de *ezplot*, pour compléter le tracé

```
» ezplot(f,-2*pi,2*pi) ; grid on
» dessineRepere
```



2.1.5. *Fonctions monotones*

Une fonction $f : I \to \mathbb{R}$ est dite **croissante** sur I, si pour tous x_1, x_2 de I on a

$$x_1 > x_2 \Longrightarrow f(x_1) \geqslant f(x_2),$$

De même f est **décroissante** sur I, si pour tous x_1, x_2 de I on a

$$x_1 > x_2 \Longrightarrow f(x_1) \leqslant f(x_2).$$

Dans ces deux cas, f est dite monotone.

1) Lorsque l'inégalité est stricte pour les valeurs de f, on dit que f est strictement monotone (strictement croissante ou strictement décroissante).

2.1.6. *Fonctions injectives, surjectives, bijectives*

Soit $f : A \to B$ où A et B sont deux sous-ensembles quelconques de $\mathbb{R}$. On rappelle que :

1) f est **injective** de A dans B si deux éléments quelconques et différents de A admettent deux images différentes :

$$x_1 \neq x_2 \Longrightarrow f(x_1) \neq f(x_2).$$

On écrira, par contraposée, que

$$f(x_1) = f(x_2) \Longrightarrow x_1 = x_2.$$

2) f est **surjective** de A dans B si tout élément de l'ensemble d'arrivée B est l'image d'au moins un élément de A. On écrira

$$\forall y \in B, \exists x \in A : \ f(x) = y.$$

3) f est **bijective** de A dans B si elle est injective et surjective. Dans ce cas, à chaque élément x de A correspond une et une seule image y de B et réciproquement, pour tout élément y de B, il existe un et un seul élément x de A tel que $f(x) = y$.

4) Lorsqu'une fonction

$$f : A \to B, \quad x \longmapsto f(x) = y,$$

est bijective, on définit la fonction dite **réciproque** (ou inverse) de f par

$$g : B \to A, \quad y \longmapsto g(y) = x,$$

caractérisée par

$$\forall x \in A, \forall y \in B \qquad f(x) = y \Longleftrightarrow x = g(y).$$

On la note

$$g = f^{-1}$$

et on vérifie que

$$\begin{cases} \forall y \in B & (f \circ f^{-1})(y) = y \\ \forall x \in A & (f^{-1} \circ f)(x) = x. \end{cases}$$

2.2. Limite d'une fonction

2.2.1. *Définitions*

2.2.1.1. *Limite finie en un point*

Soit x_0 un point de $\mathbb{R}$ et $I =]a, b[$ un intervalle ouvert contenant x_0. On se donne une fonction f définie en tout point de I, sauf peut-être en x_0. On dit que $f(x)$ tend vers une limite finie l pour $x \longrightarrow x_0$ si et seulement si pour tout $\varepsilon > 0$, il existe $\eta > 0$ tel que

$$|x - x_0| \leqslant \eta \Longrightarrow |f(x) - l| \leqslant \varepsilon.$$

On écrira

$$\lim_{x \to x_0} f(x) = l.$$

De même, on dira que

$$\lim_{x \to x_0} f(x) = +\infty$$

si et seulement si pour tout $A > 0$, il existe $\eta > 0$ tel que

$$|x - x_0| \leqslant \eta \Longrightarrow f(x) \geqslant A.$$

D'une manière similaire,

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

si et seulement si, pour tout $A > 0$, il existe $\eta > 0$ tel que

$$|x - x_0| \leqslant \eta \Longrightarrow f(x) \leqslant -A.$$

2.2.1.2. *Limite finie en* $+\infty$

Lorsque la fonction f est définie sur un intervalle $]a, +\infty[$, on dit que

$$\lim_{x \to \infty} f(x) = l$$

si pour tout $\varepsilon > 0$, il existe $B > 0$ tel que

$$x \geqslant B \Longrightarrow |f(x) - l| \leqslant \varepsilon.$$

2.2.1.3. *Autres cas*

D'une manière similaire, on définit les autres cas de limites.

2.2.1.4. *Exemple*

Soit f définie sur $]1, +\infty[$ par

$$f(x) = \frac{2x+3}{x-1}.$$

On cherche la limite de $f(x)$ en $+\infty$. Le tableau de valeurs :

```
» X=[10 100 1000 10000];
» Y=(2.*X+3)./(X-1)
Y =
2.5556 2.0505 2.0050 2.0005
```

laisse présager que

$$\lim_{x\to\infty} f(x) = 2.$$

Montrons-le. Fixons $\varepsilon > 0$. L'inégalité

$$\left|\frac{2x+3}{x-1} - 2\right| \leqslant \varepsilon$$

est équivalente à

$$\left|\frac{5}{x-1}\right| \leqslant \varepsilon,$$

ou encore à

$$\left|\frac{x-1}{5}\right| \geqslant \frac{1}{\varepsilon},$$

qui est vérifiée dès que

$$x \geqslant A = \frac{5}{\varepsilon} + 1.$$

Sous *Matlab*, on obtient la limite par

```
» syms x real;
» limit((2*x+3)/(x-1),x,inf)
ans = 2
```

2.2.1.5. *Limite à gauche, limite à droite*

On dira que f admet l pour limite à gauche en x_0 si pout tout $\varepsilon > 0$, il existe $\eta > 0$ tel que

$$x_0 - \eta \leqslant x < x_0 \Longrightarrow |f(x) - l| \leqslant \varepsilon.$$

On notera

$$\lim_{x\overset{<}{\to} x_0} f(x) = l.$$

On définit de même la limite à droite de x_0.

f admet l pour limite en x_0 si et seulement si f admet l pour limite à gauche et à droite de x_0.

Par exemple, pour la fonction f définie par

$$f(x) = \begin{cases} e^x & \text{si } x \in [0,1[\\ x^2 & \text{si } x \in [1,2], \end{cases}$$

on a

```
» syms x real ;
» limit(exp(x),x,1,'left')
ans = exp(1)
» limit(x^2,x,1,'right')
ans = 1
```

D'où

$$\lim_{x \overset{<}{\to} 1} f(x) = e,$$

et

$$\lim_{x \overset{>}{\to} 1} f(x) = 1.$$

2.2.2. *Résultat fondamental*

Les règles sur les limites de fonctions sont similaires à celles sur les suites numériques.

Un résultat important sur les limites de fonctions (en lien avec les suites) est le suivant :

Théorème.
Une fonction f admet une limite l pour $x \longrightarrow x_0$ si et seulement si, pour toute suite $(x_n)_{n \geqslant 0}$ convergente vers x_0, la suite $(f(x_n))_{n \geqslant 0}$ converge vers l.

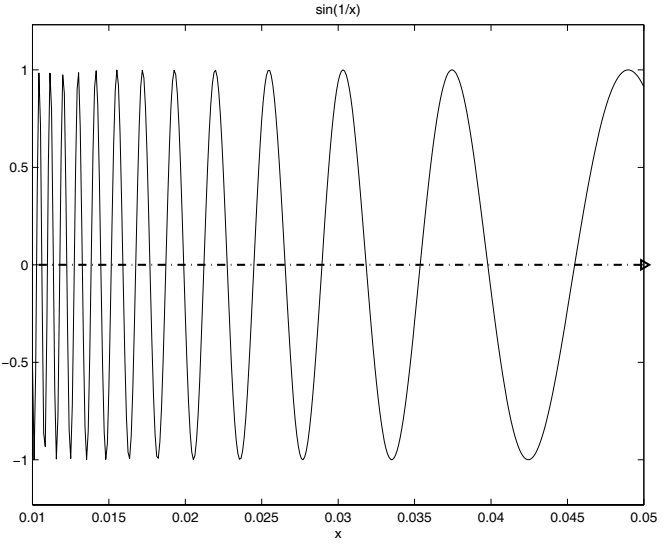
Dans la pratique, on se servira de ce théorème pour montrer que certaines fonctions oscillantes (de type trigonométriques par exemple), n'admettent pas de limite finie en certains points particuliers (ou en $\pm\infty$ éventuellement).

2.2.3. *Exemple*

Considérons la fonction $x \longmapsto f(x) = \sin(1/x)$ et examinons sa limite (éventuelle) pour $x \longrightarrow 0$.

On trace le graphe de cette fonction sur le "petit" intervalle $[0.01,\ 0.05]$

```
» syms x real
» fDeX=sin(1/x);
» ezplot(fDeX,[0.01,0.05])
» dessineRepere
```



Cette courbe montre une allure oscillante au voisinage de zéro. On construit une suite particulière $(x_n)_{n\geqslant 0}$ tendant vers zéro telle que la suite $(f(x_n))_{n\geqslant 0}$ n'admette pas de limite.

```
» syms n real
» Xn=1/(n*pi+pi/2);
» limit(Xn,n,inf)
ans=0
» fDeXn=subs(fDeX,x,Xn)
fDeXn=cos(n*pi)
```

Ainsi $f(x_n) = \cos(n\pi) = (-1)^n$ n'admet pas de limite.

2.3. Continuité

2.3.1. *Définitions*

2.3.1.1. *Cas d'un intervalle ouvert de* $\mathbb{R}$

Soit f une fonction définie sur un intervalle ouvert I de $\mathbb{R}$ et $x_0 \in I$ un point donné. La fonction f est dite **continue au point** x_0 si et seulement si

$$\lim_{x \to x_0} f(x) = f(x_0).$$

Cela traduit le fait que lorsque la variable x est proche de x_0, la valeur $f(x)$ est proche de $f(x_0)$. Autrement dit, pour tout $\varepsilon > 0$, il existe $\eta > 0$ tel que :

$$|x - x_0| \leqslant \eta \Longrightarrow |f(x) - f(x_0)| \leqslant \varepsilon.$$

En posant $x = x_0 + h$, on a la définition équivalente

$$\lim_{h \to 0} f(x_0 + h) = f(x_0).$$

On définit aussi la **continuité à gauche** en x_0 (resp. à droite en x_0). Par exemple, f est continue à gauche en x_0 si pour tout $\varepsilon > 0$, il existe $\eta > 0$ tel que

$$x_0 - \eta \leqslant x < x_0 \Longrightarrow |f(x) - f(x_0)| \leqslant \varepsilon.$$

On notera

$$\lim_{x \overset{<}{\to} x_0} f(x) = f(x_0).$$

f est continue en x_0 si et seulement si elle est continue à gauche et à droite de x_0.

La fonction f est **continue sur tout l'intervalle** I lorsqu'elle est continue en tout point de I.

2.3.1.2. *Cas d'un intervalle fermé* $[a,\ b]$

Soit f une fonction définie sur **un intervalle fermé** $[a,\ b]$ (avec $a < b$). On peut définir comme ci-dessus la continuité en x_0, pour $x_0 \in]a,\ b[$, ainsi que la continuité à droite en a, et la continuité à gauche en b.

La fonction f est **continue sur l'intervalle fermé** $[a,\ b]$ si

$$\begin{cases} f \text{ est continue sur l'intervalle ouvert }]a, b[\,, \\ f \text{ est continue à droite en } a, \\ f \text{ est continue à gauche en } b. \end{cases}$$

2.3.2. *Exemple*

On donne la fonction

$$f : \mathbb{R} \to \mathbb{R}$$
$$x \longmapsto f(x) = \begin{cases} \dfrac{3-x^2}{2} & \text{si } x \in]-\infty, 1[\\ \dfrac{1}{x} & \text{si } x \in [1, +\infty[. \end{cases}$$

Etudions sous *Matlab* sa continuité en 1.

On calcule les limites à gauche et à droite, au point 1, de $f(x)$.

```
» syms x
» f1Moins =(3-x^2)/2 ;
» f1Plus=1/x ;
» limit(f1Moins,x,1,'left')
ans = 1
» limit(f1Plus,x,1,'right')
ans = 1
```

Les deux limites étant égales à $f(1)$, la fonction est continue en 1.

2.3.3. *Résultats généraux sur la continuité*

On donne deux fonctions $f, g : I \to \mathbb{R}$ continues en x_0 et un réel quelconque λ. Alors on a les résultats :

- $f + g$ et $f - g$ sont continues en x_0,
- λf et $f.g$ sont continues en x_0,
- $\dfrac{f}{g}$ est continue en x_0, (si $g(x_0)$ est non nul).

Lorsque on a

$$f : I \to J, \quad g : J \to \mathbb{R}$$

où J est un autre intervalle de $\mathbb{R}$ et si f est continue en x_0 et g est continue en $f(x_0)$ alors la composée $g \circ f$ définie sur I par

$$(g \circ f)(x) = g(f(x)),$$

est continue en x_0.

Les fonctions usuelles :

1) polynomiales,

2) trigonométriques directes et inverses,

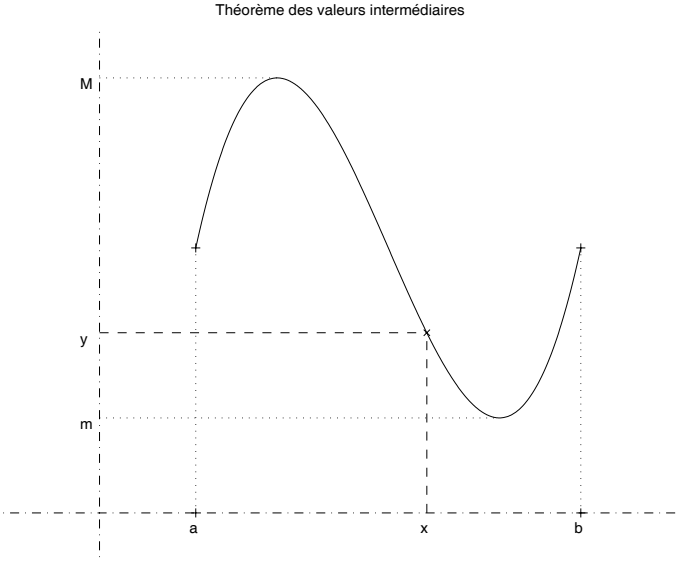
3) logarithme, exponentielle,...

sont continues sur leur domaine de définition.

On retiendra le théorème suivant :

Théorème des valeurs intermédiaires.
Pour toute fonction f réelle continue sur $[a, b]$, *l'image par f de cet intervalle est l'intervalle $[m, M]$ où m est le minimum de f et M est le maximum de f sur* $[a, b]$.

Autrement dit, pour toute valeur **intermédiaire** y comprise entre m et M il existe au moins une valeur x dans $[a, b]$ telle que $f(x) = y$.



2.4. Dérivation

2.4.1. *Définitions*

La notion de dérivée (comme le mot l'indique bien) est liée à la "courbure" du graphe représentatif d'une fonction. Cette notion sert aussi dans de nombreuses applications telles que la cinématique, les systèmes dynamiques, etc...

Une fonction $f : I \to \mathbb{R}$ (I intervalle ouvert) est **dérivable en un point** $x_0 \in I$ si le rapport

$$\frac{f(x) - f(x_0)}{x - x_0},$$

défini sur $I \setminus \{x_0\}$, admet une limite finie lorsque $x \to x_0$.

Noter que ce rapport est exactement le coefficient directeur de la droite D_x joignant les points M_x et M_{x_0} de coordonnées $(x, f(x))$ et $(x_0, f(x_0))$.

La limite (lorsqu'elle existe) est notée

$$f'(x_0) \text{ ou } \frac{df}{dx}(x_0),$$

et est appelée la **dérivée** de f en x_0.

De même ici, on peut définir une dérivée à gauche et une dérivée à droite au point x_0 : par exemple la dérivée à gauche est, quand elle existe,

$$f'_g(x_0) = \lim_{x \overset{<}{\to} x_0} \frac{f(x) - f(x_0)}{x - x_0}.$$

Si f est dérivable en x_0 alors

$$f'(x_0) = f'_g(x_0) = f'_d(x_0).$$

Si la fonction est dérivable en chaque point x de I, on définit la **fonction dérivée**

$$f' : I \longrightarrow \mathbb{R}, \qquad x \longmapsto f'(x).$$

Une fonction f définie sur un intervalle $[a,\ b]$ (avec $a < b$), est dite **dérivable sur l'intervalle fermé** $[a,\ b]$ si

$$\begin{cases} f \text{ est dérivable sur l'intervalle ouvert }]a, b[\,, \\ f \text{ est dérivable à droite en } a, \\ f \text{ est dérivable à gauche en } b. \end{cases}$$

2.4.2. *Exemple*

La fonction $x \longmapsto f(x) = x^2$ est dérivable en tout $x_0 \in \mathbb{R}$:

en effet

$$\lim_{x \to x_0} \frac{f(x) - f(x_0)}{x - x_0} = \lim_{x \to x_0} \frac{x^2 - x_0^2}{x - x_0} = \lim_{x \to x_0} (x + x_0) = 2x_0 = f'(x_0).$$

Ce calcul de limite s'effectue avec *Matlab* par :

```
» syms x x0 real
» fDeX=x^2 ;
» fDeX0=x0^2 ;
» fPrimeDeX0=limit((fDeX- fDeX0)/(x-x0),x,x0)
fPrimeDeX0 = 2*x0
```

Grâce à la commande **diff**, on obtient directement la fonction dérivée :

```
» fPrimeDeX=diff(fDeX,x)
 fPrimeDeX=2*x
» fPrimeDeX0=subs(fPrimeDeX,x,x0)
fPrimeDeX0 = 2*x0
```

2.4.3. *Interprétation géométrique*

Comme nous l'avons dit précédemment, si $f'(x_0)$ existe, la droite D_x admet donc une droite limite D_{x_0} qui est tangente à la courbe représentative de f au point M_0. Cette tangente a pour équation cartésienne

$$y = f'(x_0)(x - x_0) + f(x_0).$$

En notant

$$\varepsilon(x) = \frac{f(x) - f(x_0)}{x - x_0} - f'(x_0),$$

on remarque que pour tout $x \in I$

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \varepsilon(x)(x - x_0),$$

avec

$$\varepsilon(x) \to 0 \text{ pour } x \to x_0.$$

En posant $x = x_0 + h$, on a

$$f(x_0 + h) = f(x_0) + f'(x_0)h + h.\varepsilon_1(h)$$

où $\varepsilon_1(h) = \varepsilon(x_0 + h)$.

Cette dernière égalité donne une première **approximation affine** de f au voisinage de x_0 dès que $f'(x_0)$ existe et montre aussi qu'une fonction dérivable en x_0 est nécessairement continue en ce point.

2.4.4. *Propriétés générales*

Soient $f, g : I \to \mathbb{R}$ deux fonctions dérivables sur I et $\lambda \in \mathbb{R}, n \in \mathbb{N}$. Alors on a

- $(f+g)' = f' + g'$,
- $(\lambda f)' = \lambda f'$,
- $(fg)' = f'g + fg'$,
- $\left(\frac{f}{g}\right)' = \frac{f'g - fg'}{g^2}$, *(si* $g \neq 0$*)*,
- $(f^n)' = nf'f^{n-1}$,
- $(\ln f)' = \frac{f'}{f}$, *(dérivée logarithmique de* $\ln f$*, si* $f > 0$*)*,

• Lorsqu'on a

$$f : I \to J, \quad g : J \to \mathbb{R}$$

où J est un autre intervalle de $\mathbb{R}$, alors, si f est dérivable sur I et g est dérivable sur J, on a

$$(g \circ f)'(x) = g'(f(x)).f'(x).$$

• Si f est dérivable et est une bijection de I dans J, alors f^{-1} est dérivable en tout point où $f'(x_0) \neq 0$, et on a

$$\left(f^{-1}\right)'(y_0) = \frac{1}{f'(x_0)} = \frac{1}{f'(f^{-1}(y_0))},$$

avec $y_0 = f(x_0)$.

2.4.5. *Dérivées successives*

Soit $f : I \to \mathbb{R}$ dérivable sur I. Si la fonction f' est dérivable, on dira que f est deux fois dérivable et on note

$$f''(x) = (f')'(x).$$

En réitérant, on définit de même la dérivée à l'ordre n de f notée $f^{(n)}$ par

$$f^{(n)}(x) = \left(f^{(n-1)}\right)'(x).$$

On peut montrer par récurrence la formule dite de Leibnitz :

$$(f.g)^{(n)} = \sum_{k=0}^{k=n} C_n^k f^{(k)} g^{(n-k)},$$

vraie pour deux fonctions f et g, n fois dérivables. On rappelle que

$$C_n^k = \frac{n!}{(n-k)!k!}.$$

On dira qu'une fonction f est de classe C^k ($k \in \mathbb{N}$) sur I si $f, f', ..., f^{(k)}$ existent et sont continues sur I. Une fonction de classe C^0 est une fonction continue sur I.

Sous *Matlab*, le calcul de la dérivée $nième$ s'effectue en utilisant *diff*. Par exemple, pour

$$f(x) = \left(x^3 + 2x - 5\right) e^x$$

```
» syms x real ;
» fDeX=(x^3+2*x-5)*exp(x);
» fOrdre4DeX=diff(fDeX,x,4);
» factor(fOrdre4DeX)
ans=exp(x)*(x+1)*(x^2+11*x+27)
```

on obtient

$$f^{(4)}(x) = (x+1)\left(x^2 + 11x + 27\right) e^x.$$

2.4.6. *Conséquences de la dérivation*

Les premières propriétés de la dérivation sont :

> *Si f, définie sur $I =]a, b[$, est dérivable en $x_0 \in I$ et admet un extremum en x_0 alors $f'(x_0) = 0$.*

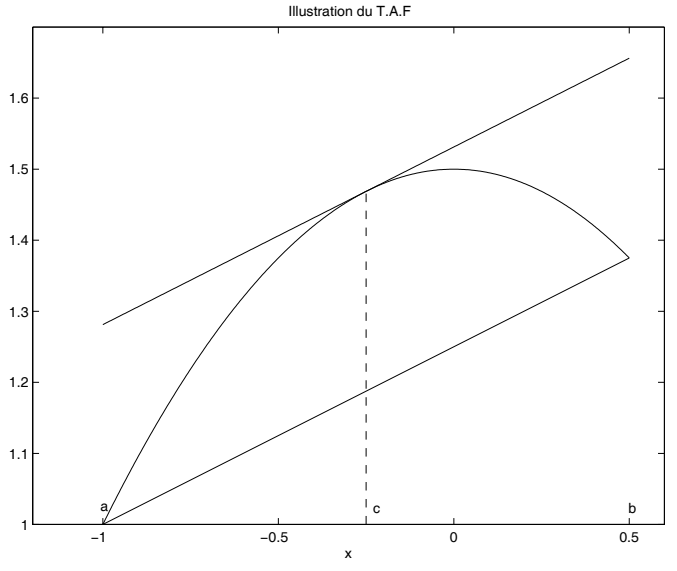
> *Si f est continue sur $[a, b]$ avec $f(a) = f(b)$ et est dérivable sur $]a, b[$ alors il existe un point c de $]a, b[$ tel que $f'(c) = 0$.*

Ce dernier résultat est connu sous le nom du **théorème de Rolle** et exprime le fait qu'il y a au moins un point où la courbe représentative de f admet une tangente horizontale (c'est-à-dire parallèle à l'axe des abcisses).

> *Si f est continue sur $[a, b]$ et est dérivable sur $]a, b[$ alors il existe un point c de $]a, b[$ tel que $f(b) - f(a) = (b - a)f'(c)$.*

C'est le théorème des **accroissements finis** (T.A.F). Il exprime le fait qu'il existe au moins une tangente à la courbe de f parallèle à la sécante joignant les points de

coordonnées $(a, f(a))$ et $(b, f(b))$. (Voir figure ci-dessous).



En écrivant $b = a + h$ et sachant qu'un point de l'intervalle ouvert $]a, b[$ s'écrit sous forme

$$c = a + \theta(b - a) = a + \theta h$$

où $\theta \in]0, 1[$, on obtient la formulation courante

$$f(a + h) - f(a) = hf'(a + \theta h),$$

qui exprime bien l'idée d'accroissements finis. De ce théorème on déduit aussi (sous les mêmes hypothèses) :

- f est croissante sur I si $\forall x \in I \quad f'(x) \geqslant 0$,
- f est décroissante sur I si $\forall x \in I \quad f'(x) \leqslant 0$,
- f est constante sur I si $\forall x \in I \quad f'(x) = 0$.

2.4.7. *Etude d'une fonction avec Matlab*

On se propose d'étudier les variations de la fonction f définie sur $\mathbb{R}$ par

$$f(x) = 1 + xe^{-x},$$

et de tracer sa courbe représentative.

– Cette fonction est de classe C^1 sur $\mathbb{R}$.

– On déclare l'expression symbolique correspondant à la fonction, on calcule la dérivée et on la factorise :

```
» syms x real
» f=1+x*exp(-x);
» fprime=diff(f)
=exp(-x)-x*exp(-x)
» factor(fprime)
=-exp(-x)*(-1+x)
```

Ce résultat est suffisant pour donner le signe de $f'(x)$, mais on peut aussi résoudre l'équation

$$f'(x) = 0,$$

ou l'inéquation

$$f'(x) > 0.$$

```
» solve(fprime)
ans=1
»maple('solve(-exp(-x)*(-1+x)>0)')
ans = realRange(-inf,open(1))
```

Ainsi, la dérivée est positive sur $]-\infty, 1[$. On calcule alors, symboliquement et numériquement, $f(1)$, puis les limites de f en $+\infty$ et $-\infty$.

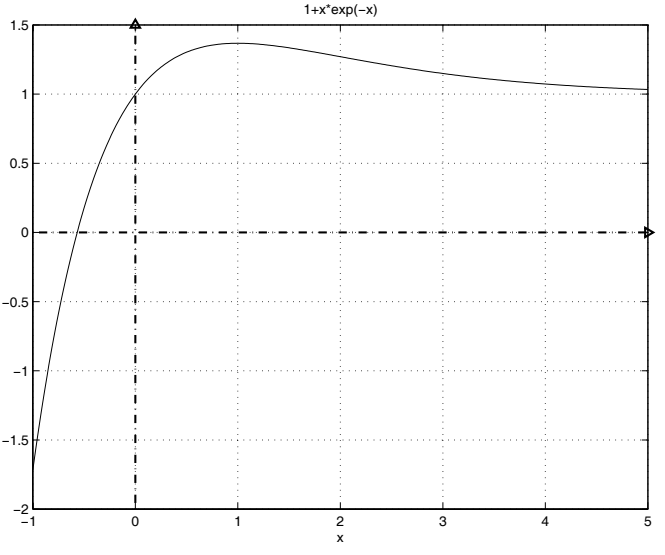
```
»fDe1=subs(f,x,sym(1))
fDe1=1+exp(-1)
» double(fDe1)
ans=1.3679
» limit(f,x,-inf,'right')
ans=-inf
» limit(f,x,inf,'left')
ans=1
```

D'où le tableau de variations

x	$-\infty$		1		$+\infty$
$f'(x)$		$+$	0	$-$	
$f(x)$	$-\infty$	$\nearrow$	$1+1/e$	$\searrow$	1

On construit le graphe sur un intervalle contenant 1, par exemple $[-1,\ 5]$:

```
» ezplot(f,-1,5)
» grid on
» axis auto % ajuste le cadre
» dessineRepere
```



2.4.8. *Retour à l'exemple modèle*

Etudions la dérivabilité et la continuité de la fonction dérivée en 1 de la fonction :

$$f : \mathbb{R} \to \mathbb{R}$$
$$x \longmapsto f(x) = \begin{cases} \dfrac{3-x^2}{2} & \text{si } x \in]-\infty, 1[\\ \dfrac{1}{x} & \text{sinon.} \end{cases}$$

La fonction f est indéfiniment dérivable sur $\mathbb{R} - \{1\}$. On déclare les deux expressions de f

```
» syms x real
» f1Moins =(3-x^2)/2 ;
» f1Plus=1/x ;
```

On calcule les limites à gauche et à droite suivantes :

$$\lim_{x \overset{<}{\to} 1} \frac{f(x) - f(1)}{x - 1}, \qquad \lim_{x \overset{>}{\to} 1} \frac{f(x) - f(1)}{x - 1},$$

```
» taux1Moins = (f1Moins - 1)/(x-1)
taux1Moins = (1/2-1/2*x^2)/(x-1)
» factor(taux1Moins)
ans = -1/2*x-1/2
» taux1Plus = (f1Plus - 1)/(x-1)
taux1Plus = (1/x-1)/(x-1)
» factor(taux1Plus)
ans = -1/x
» limit(taux1Moins,x,1,'left')
ans = -1
» limit(taux1Plus,x,1,'right')
ans = -1
```

Ces deux limites étant égales, la fonction est dérivable en 1 et $f'(1) = -1$.

Pour la continuité de la fonction dérivée en 1, on calcule

$$\lim_{x \overset{<}{\to} 1} f'(x) \quad \text{et} \quad \lim_{x \overset{>}{\to} 1} f'(x).$$

```
» f1MoinsPrime = diff(f1Moins)
f1MoinsPrime = -x
» f1PlusPrime = diff(f1Plus)
f1PlusPrime = -1/x^2
» limit(f1MoinsPrime,x,1,'left')
ans = -1
» limit(f1PlusPrime,x,1,'right')
ans = -1
```

Donc f est de classe C^1 (continûment dérivable) et

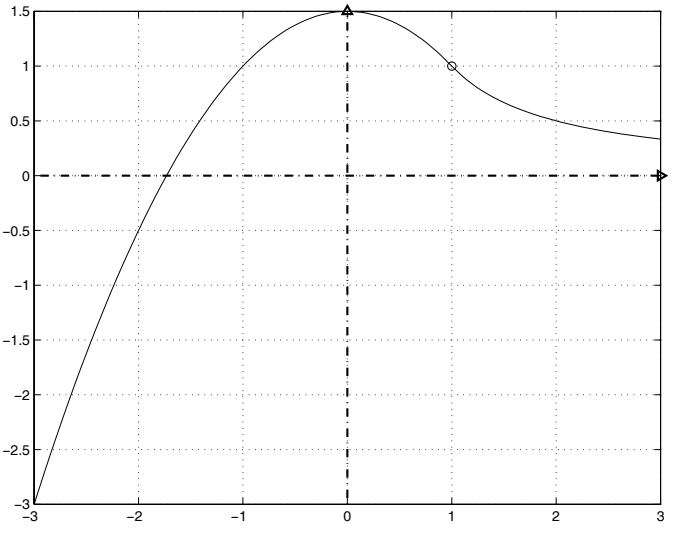
$$f' : \mathbb{R} \to \mathbb{R}$$
$$x \longmapsto f'(x) = \begin{cases} -x & \text{si } x \in]-\infty, 1[\\ -1 & \text{si } x = 1 \\ -1/x^2 & \text{sinon.} \end{cases}$$

Pour le graphe de cette fonction définie par morceaux, on crée un fichier appelé $f1.m$ dans lequel est définie la fonction f

```
function y = f1(x)
if(x<1) y = (3-x.^2)./2 ;
else y =1./x ;
end
```

puis on utilise *fplot* pour le tracé de la courbe :

```
» fplot('f1',[-3 3])
» grid on ; hold on
» plot(1,f1(1),'o') % pour marquer le point d'abscisse 1
» dessineRepere
```



Observons que le raccord au point 1 se fait d'une manière "lisse", au sens que la fonction en ce point admet une dérivée et cette dernière y est continue.

2.5. Fonctions trigonométriques inverses

2.5.1. *Rappel*

Dans cette section on utilisera en particulier le résultat suivant :

Théorème.
Si f est une fonction continue strictement croissante sur $[a, b]$ alors f est une bijection de $[a, b]$ dans $[f(a), f(b)]$. Il en est de même si f est strictement décroissante de $[a, b]$ dans $[f(b), f(a)]$.

De plus sa fonction réciproque $g = f^{-1}$ est de même nature (c'est-à-dire strictement décroissante ou strictement croissante selon le sens de variations de f).

Supposons que f est continue strictement croissante et notons

$$C(f) = \{(x, f(x)) : x \in [a, b]\},$$

la courbe représentative de f dans le plan rapporté à un repère orthonormé. Alors celle de f^{-1} s'écrit :

$$\begin{aligned} C(f^{-1}) &= \{(y, g(y)) : y \in [f(a), f(b)]\} \\ &= \{(f(x), x) : x \in [a, b]\}, \end{aligned}$$

et est clairement obtenue à partir de $C(f)$ par symétrie par rapport à la première bissectrice d'équation $y = x$.

Comme applications, on va définir les trois fonctions circulaires inverses suivantes.

2.5.2. *Fonction arcsin*

On part de la fonction sinus, restreinte à l'intervalle $[-\pi/2, +\pi/2]$

$$\begin{aligned} \sin : [-\pi/2, +\pi/2] &\to [-1, +1] \\ x &\longmapsto \sin(x), \end{aligned}$$

qui est continue et strictement croissante et donc elle admet une fonction réciproque $\sin^{-1}$ notée $\arcsin$

$$\begin{aligned} \arcsin : [-1, +1] &\to [-\pi/2, +\pi/2] \\ y &\longmapsto \arcsin(y), \end{aligned}$$

avec la caractérisation

$$\sin(x) = y, \quad x \in [-\pi/2, +\pi/2]$$

si et seulement si

$$\arcsin(y) = x, \quad y \in [-1, +1].$$

Cette fonction est notée sous *matlab* **asin**.

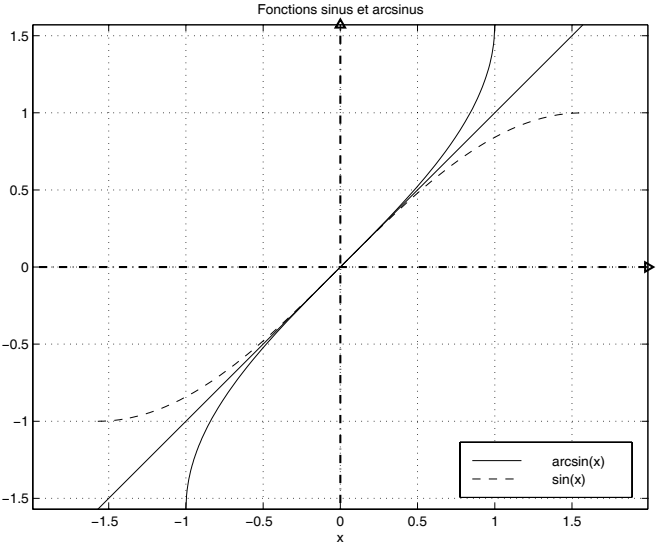
On représente sur une même figure les graphes des fonctions $\arcsin$ et sin, ainsi que la droite d'équation

$$y = x.$$

```
» clf
» fplot('asin',[-1,1])
» hold on
» fplot('sin',[-pi/2,pi/2],'–')
» ezplot('x',[-pi/2,pi/2])
» axis equal ; axis auto
```

Pour placer une légende, associée aux tracés successifs :

```
» legend ('arcsin(x)','sin(x)',4)
» grid on ; dessineRepere
```



Par lecture inverse des valeurs du sinus on a par exemple

$$\begin{cases} \arcsin(0) = 0, \\ \arcsin(1/2) = \pi/6, \\ \arcsin(\sqrt{3}/2) = \pi/3, \\ \arcsin(\sqrt{2}/2) = \pi/4, \\ \arcsin(1) = \pi/2. \end{cases}$$

En appliquant la règle de dérivation d'une fonction inverse (voir 2.4.4), la dérivée de arcsin en $y = \sin(x) \in]-1, 1[$ est

$$(\arcsin)'(y) = \frac{1}{\cos(x)} = \frac{1}{\sqrt{1-y^2}}.$$

2.5.3. *Fonction arccos*

On part cette fois de la restriction de la fonction cosinus à l'intervalle $[0,\,\pi]$

$$\begin{array}{rcl}\cos:\,[0,\pi] & \to & [-1,+1]\\ x & \longmapsto & \cos(x),\end{array}$$

qui est continue et strictement décroissante et donc on définit son inverse par

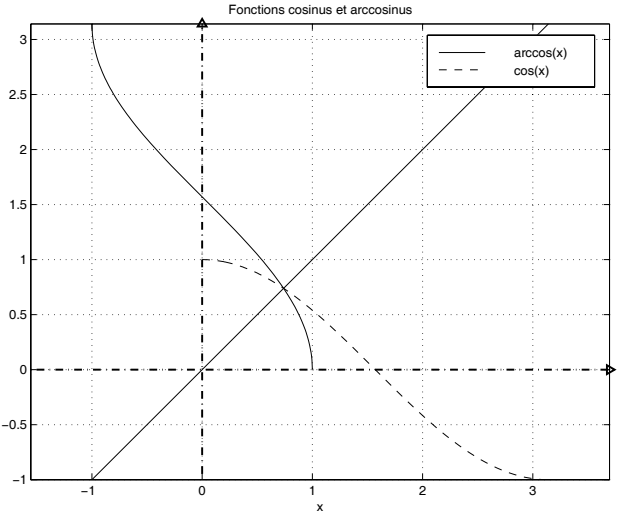
$$\begin{array}{rcl}\arccos:\,[-1,+1] & \to & [0,\pi]\\ y & \longmapsto & \arccos(y).\end{array}$$

La dérivée en $y=\cos(x)\in]-1,1[$ s'obtient par

$$(\arccos)'(y)=\frac{1}{\cos'(x)}=\frac{1}{-\sin(x)}=-\frac{1}{\sqrt{1-y^2}}.$$

Cette fonction est notée sous *matlab* **acos**.

On obtient les graphes de ces deux fonctions de la même façon que précédemment :



2.5.4. *Fonction arctan*

La restriction de la fonction tangente à l'intervalle $]-\pi/2,+\pi/2[$

$$\begin{array}{rcl}\tan:\,]-\pi/2,+\pi/2[& \to & \mathbb{R}\\ x & \longmapsto & \tan(x),\end{array}$$

est continue et strictement croissante donc elle admet une fonction réciproque notée

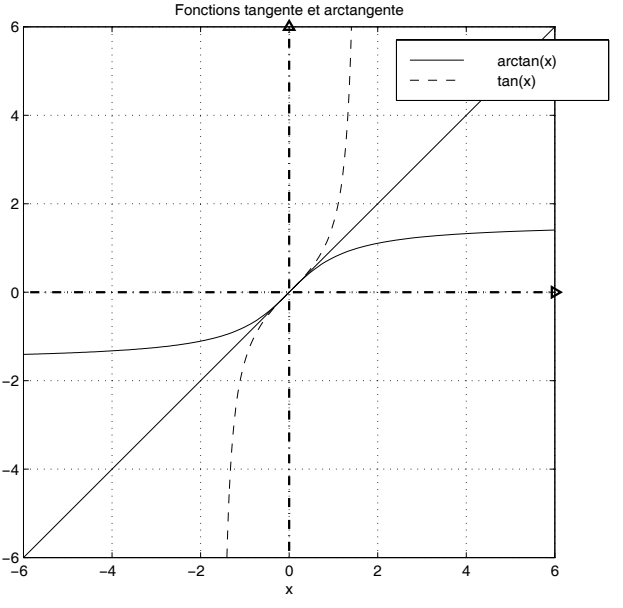
$$\begin{array}{rcl} \arctan : \mathbb{R} & \to &]-\pi/2, +\pi/2[\\ y & \longmapsto & \arctan(y), \end{array}$$

qui est de même nature et, si $y = \tan(x)$,

$$(\arctan)'(y) = \frac{1}{1+\tan^2(x)} = \frac{1}{1+y^2}.$$

Cette fonction est notée sous *matlab* **atan**.

Graphiquement, on a



2.5.5. *Exemple modèle*

On donne la fonction suivante :

$$\begin{array}{rcl} f : \mathbb{R}_+ & \to & \mathbb{R} \\ x & \longmapsto & f(x) = \arcsin\left(\dfrac{2\sqrt{x}}{1+x}\right), \end{array}$$

On note, pour $x \geqslant 0$,

$$h(x) = \frac{2\sqrt{x}}{1+x}.$$

1) Vérifier que pour tout $x \geqslant 0$, on a

$$h(x) = 1 - \frac{(1-\sqrt{x})^2}{1+x}.$$

En déduire le domaine de définition D de f.

2) Montrer que, sur

$$D' =]0,1[\cup]1,+\infty[,$$

f est dérivable, et vérifier avec *Matlab* la relation

$$f'(x) = h'(x).\arcsin'(h(x)),$$

pour tout $x \in D'$.

3) Etudier la dérivabilité de f en 0 et 1.

4) Etudier les variations de la fonction f et tracer son graphe dans un repère orthonormal.

5) On donne la fonction

$$\begin{array}{rccl} g: & \mathbb{R}_+ & \to & \mathbb{R} \\ & x & \longmapsto & g(x) = 2\arctan\sqrt{x}. \end{array}$$

Etudier ses variations et tracer son graphe sur la même figure que f.

6) En comparant $f'(x)$ et $g'(x)$, sur $]0,1[$, puis sur $]1,+\infty[$, montrer que

$$f(x) = \begin{cases} g(x) \text{ si } x \in [0,1] \\ \pi - g(x) \text{ si } x \in]1,+\infty[. \end{cases}$$

Solution

1) On définit avec *Matlab*

$$d(x) = 1 - \frac{(1-\sqrt{x})^2}{1+x},$$

et on vérifie que $d(x) = h(x)$:

```
» syms x positive
» dDeX=1-(1-sqrt(x))^2/(1+x);
» simplify(dDeX)
ans = 2*x^(1/2)/(1+x)
```

On déduit de cette égalité que, pour $x \in \mathbb{R}_+$, on a

$$h(x) \leqslant 1.$$

De plus

$$h(x) = \frac{2\sqrt{x}}{1+x} \geqslant 0.$$

Ainsi, pour tout $x \in \mathbb{R}_+$, $h(x) \in [0,\,1]$ et $\arcsin(h\,(x))$ est défini.
Le domaine de définition de f est $D = \mathbb{R}_+$.

2) On sait que $u \longmapsto \arcsin u$ est dérivable sur $]-1,\ 1[$.
$x \longmapsto \arcsin(h\,(x))$ est donc dérivable si $x \longmapsto h(x)$ l'est, et si de plus

$$h(x) \in]-1, 1[.$$

La fonction h est dérivable sur $]0,\,+\infty[$, et $h(x) = 1$ si

$$\frac{(1-\sqrt{x})^2}{1+x} = 0,$$

soit $x = 1$.
La fonction f est donc dérivable sur

$$D' =]0, 1[\cup]1, +\infty[.$$

Pour $x \in D'$, on calcule avec *Matlab* $f'(x)$ et

$$d_1\,(x) = h'(x).\,\arcsin'(h(x)) = h'(x) \times \frac{1}{\sqrt{1-(h(x))^2}}.$$

```
» hDeX=2*sqrt(x)/(1+x);
» fDeX=asin(hDeX);
» hPrimeDeX=diff(hDeX)
hPrimeDeX = 1/x^(1/2)/(1+x)-2*x^(1/2)/(1+x)^2
» d1DeX=simplify(hPrimeDeX*1/sqrt(1-hDeX^2))
d1DeX = -1/x^(1/2)/(x+1)*signum(x-1)
» fPrimeDeX=simplify(diff(fDeX))
fPrimeDeX = -1/x^(1/2)/(x+1)*signum(x-1)
```

On a bien vérifié l'égalité

$$f'(x) = h'(x).\,\arcsin'(h(x)).$$

3) Pour étudier la dérivabilité à droite en 0, on calcule avec *Matlab*

$$\lim_{x \xrightarrow{>} 0} \frac{f(x) - f(0)}{x - 0}$$

```
» fDe0=simplify(subs(fDeX,x,sym('0')))
fDe0 = 0
» limit((fDeX-fDe0)/x,x,0,'right')
ans = inf
```

Comme

$$\lim_{x \xrightarrow{>} 0} \frac{f(x) - f(0)}{x - 0} = +\infty,$$

f n'est pas dérivable à droite en 0, mais sa courbe représentative admet au point d'abscisse 0 une demi-tangente verticale.
On étudie de même la dérivabilité en 1 :

```
» fDe1=simplify(subs(fDeX,x,sym('1')))
fDe1 = 1/2*pi
» limit((fDeX-fDe1)/(x-1),x,1,'right')
ans = -1/2
» limit((fDeX-fDe1)/(x-1),x,1,'left')
ans = 1/2
```

Ainsi

$$\lim_{x \xrightarrow{>} 1} \frac{f(x) - f(1)}{x - 1} = -\frac{1}{2} \quad \text{et} \quad \lim_{x \xrightarrow{<} 1} \frac{f(x) - f(1)}{x - 1} = \frac{1}{2}.$$

f admet en 1 une dérivée à droite et une dérivée à gauche, qui sont distinctes.

4) On a montré à la question 2, qu'en tenant compte du signe de $(x-1)$, on avait

$$f'(x) = \begin{cases} \dfrac{1}{\sqrt{x}\,(1+x)} & \text{si } x \in]0,\ 1[\\ \dfrac{-1}{\sqrt{x}\,(1+x)} & \text{si } x \in]1,\ +\infty[. \end{cases}$$

Le signe de $f'(x)$ s'en déduit immédiatement.
Pour compléter le tableau de variations, on calcule la limite de f en $+\infty$:

```
» limit(fDeX,x,inf)
ans = 0
```

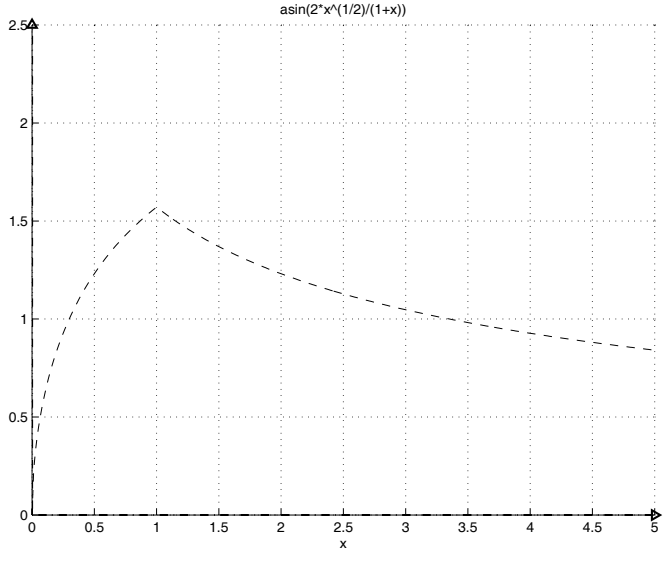
D'où

x	0		1		$+\infty$
$f'(x)$	$\|\|$	$+$	$\|\|$	$-$	
$f(x)$	0	$\nearrow$	$\pi/2$	$\searrow$	0

On construit la courbe représentative de f.

```
» hold on
» set(gca,'LineStyle','--')
» ezplot(fDeX,0,5)
» grid on
» axis ([0 5 0 2.5])
» dessineRepere
```

La commande *set(gca,'LineStyle','--')* permet d'obtenir lorsqu'on utilise ensuite *ezplot* un tracé en pointillés. Cela permettra de différencier les graphes de f et g.



5) La fonction g est définie, continue sur $\mathbb{R}_+$, dérivable sur $]0, +\infty[$. On calcule sa dérivée

```
» gDeX=2*atan(sqrt(x));
» gPrimeDeX=diff(gDeX)
gPrimeDeX = 1/x^(1/2)/(1+x)
```

Ainsi

$$g'(x) = \frac{1}{\sqrt{x}\,(1+x)},$$

et g est strictement croissante sur $[0,\ +\infty[$. On calcule les limites aux bornes $g(0)$ et $\lim_{x \longrightarrow +\infty} g(x)$

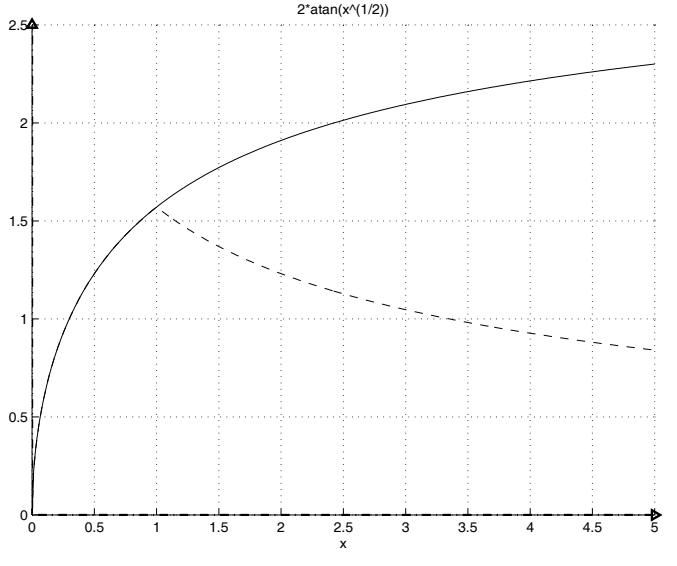
```
» gDe0 =simplify(subs(gDeX,x,sym('0')))
gDe0 =0
» limit(gDeX,x,inf)
ans = pi
```

D'où le tableau de variations

x	$0 \qquad +\infty$
$g'(x)$	$\|\|\ +$
$g(x)$	$0 \nearrow \pi$

On trace la courbe représentative de g en traits pleins :

```
» set(gca,'LineStyle','-')
» ezplot(gDeX,0,5)
» axis ([0 5 0 2.5])
» dessineRepere
```



6) Sur l'intervalle $]0,\ 1[$, l'expression donnée par $Matlab$ de $f'(x)$ se simplifie par :

```
» fPrime0_1=subs(fPrimeDeX,'signum(x-1)',-1)
fPrime0_1= 1/x^(1/2)/(1+x)
» fPrime0_1-gPrimeDeX
ans = 0
```

Sur cet intervalle, on a $f'(x) - g'(x) = 0$, d'où

$$f(x) - g(x) = Cste.$$

Pour déterminer cette constante, on calcule par exemple

$$f(1/3) - g(1/3).$$

```
» fDe1Tiers=simplify(subs(fDeX,x,sym('1/3')))
fDe1Tiers =1/3*pi
» gDe1Tiers=simplify(subs(fDeX,x,sym('1/3')))
gDe1Tiers =1/3*pi
» fDe1Tiers-gDe1Tiers
ans = 0
```

Donc, pour tout $x \in]0, 1[$, on a $f(x) = g(x)$.
On a aussi vérifié que $f(0) = g(0) = 0$, et on a

$$f(1) = g(1) = \pi/2.$$

Sur l'intervalle $]1, +\infty[$, on simplifie $f'(x)$, puis on calcule $f'(x) + g'(x)$

```
» fPrime1_inf=subs(fPrimeDeX,'signum(x-1)',1)
fPrime1_inf = -1/x^(1/2)/(1+x)
»fPrime1_inf+gPrimeDeX
ans =0
```

d'où

$$f(x) + g(x) = Cste.$$

On évalue cette constante comme précédemment :

```
» fDe3=simplify(subs(fDeX,x,sym('3')))
fDe3 = 1/3*pi
» gDe3=simplify(subs(gDeX,x,sym('3')))
gDe3 =2/3*pi
»fDe3+gDe3
ans=pi
```

D'où $f(x) = \pi - g(x)$ si $x \in]1, +\infty[$.

2.6. Comparaison de deux fonctions

2.6.1. *Notion de voisinage*

Soit x_0 un point de $\mathbb{R}$. Une fonction f est dite définie dans un voisinage de x_0, si elle est définie en tout point d'un intervalle ouvert $I =]a, b[$ contenant x_0, sauf peut-être en x_0.

Une fonction f est définie au voisinage de $+\infty$ (respectivement $-\infty$) si elle est définie sur un intervalle de la forme $]a, +\infty[$ (respectivement $]-\infty, a[$).

2.6.2. *Notations dites de Landau*

Soit f définie au voisinage de x_0 (pouvant être éventuellement $\pm\infty$). Il arrive fréquemment qu'au voisinage du point x_0 les valeurs de f soient du même ordre de grandeur que celles d'une autre fonction g ayant une expression analytique plus simple à utiliser. Par exemple, les expressions

$$f(x) = \frac{x^2+1}{x-1} \quad \text{et} \quad g(x) = x,$$

vérifient

$$\lim_{x\to+\infty} \frac{f(x)}{g(x)} = \lim_{x\to+\infty} = \frac{x^2+1}{x^2-x} = 1.$$

De même, pour

$$f(x) = \sin(x) \quad \text{et} \quad g(x) = x,$$

on a

$$\lim_{x\to 0} \frac{f(x)}{g(x)} = 1.$$

Il est alors préférable de travailler sur la deuxième expression au voisinage du point considéré. D'où les définitions :

- *on dira que f est équivalente à g au voisinage de x_0 si*
  $\lim_{x\to x_0} \frac{f(x)}{g(x)} = 1.$ *On notera :* $f(x) \sim g(x) \quad (x \longrightarrow x_0)$,

- *on dira que f est un petit o de g au voisinage de x_0 si*
  $\lim_{x\to x_0} \frac{f(x)}{g(x)} = 0.$ *On notera :* $f(x) = o(g(x)) \quad (x \longrightarrow x_0)$,

- *on dira que f est un grand O de g au voisinage de x_0 si*
  le rapport $\frac{f(x)}{g(x)}$ *est borné au voisinage de* x_0.
  On notera : $f(x) = O(g(x)) \quad (x \longrightarrow x_0)$.

2.6.3. *Exemples*

On a

$$\sin(x) \sim x \qquad (x \to 0)\,,$$

et pour tout $n \geqslant 0$,

$$\begin{aligned} x^n &= o(1) \quad (x \to 0)\,, \\ x^{n+1} &= o(x^n) \quad (x \to 0)\,, \end{aligned}$$

Les définitions ci-dessus s'appliquent aux suites. Ainsi, pour $n \in \mathbb{N}$

$$\frac{3}{4}n^3 + 2n - 14 = O(n^3) \qquad (n \to +\infty)\,.$$

2.7. Formules de Taylor et développements limités

Nous avons vu précédemment que la dérivation est essentielle dans l'étude des fonctions. On va voir que les développements limités fournissent encore plus de précision dans l'allure et le comportement d'une fonction au voisinage d'un point donné. Une autre application concrète et importante de cette notion est le calcul approché de la valeur d'une fonction en un point, en particulier pour celles qui ne sont pas de type polynomial.

2.7.1. *Diverses formules de Taylor*

Dans toute cette section, on se donne une fonction $f : \mathbb{R} \longrightarrow \mathbb{R}$ indéfiniment dérivable.

2.7.1.1. *Formule de Taylor-Lagrange*

On considère f sur un intervalle fini $[a, b]$ et $n \in \mathbb{N}$.
Alors il existe un point $c \in]a, b[$ tel que :
$f(b) - f(a) = (b-a)f'(a) + \frac{1}{2!}(b-a)^2 f''(a) + \ldots + \frac{1}{n!}(b-a)^n f^{(n)}(a)$
$+\frac{1}{(n+1)!}(b-a)^{n+1} f^{(n+1)}(c)$.

C'est la formule de Taylor-Lagrange à l'ordre n pour f.
Pour $n = 0$, la formule précédente est celle du théorème des accroissements finis.
Lorsqu'on sait que la dérivée d'ordre $(n+1)$ de f est majorée par une constante M, la formule précédente fournit une approximation polynomiale de $f(a+h)$ pour h petit, en fonction des dérivées successives de f en a. En effet, en posant $b - a = h$, on a

$$\begin{aligned} & \left| f(a+h) - \left(f(a) + hf'(a) + \tfrac{1}{2!}h^2 f''(a) + \ldots \tfrac{1}{n!}h^n f^{(n)}(a) \right) \right| \\ = & \left| \tfrac{1}{(n+1)!} h^{n+1} f^{(n+1)}(c) \right| \\ \leqslant & \frac{M\,|h|^{n+1}}{(n+1)!}. \end{aligned}$$

2.7.1.2. *Exemple*

On se propose de :

1) afficher la valeur $\arctan(1,01)$,

2) calculer les dérivées première et deuxième de $\arctan(x)$,

3) donner une majoration de $(\arctan)''(x)$ sur l'intervalle $[a,b]=[1,1.01]$,

4) utiliser la formule de Taylor-Lagrange pour donner une approximation $\overline{v}$ de $v=\arctan(1,01)$, et une estimation de l'erreur commise en remplaçant v par $\overline{v}$.

Solution

1) On obtient avec *Matlab*

```
» format long
» atan(1.01)
ans=0.79037324672830
```

2) On calcule les dérivées successives de $f(x)=\arctan x$:

```
» syms x real
» FdeX=atan(x);
»Fdif1=diff(FdeX)
Fdif1=1/(1+x^2)
»Fdif2=diff(Fdif1)
Fdif2=-2/(1+x^2)^2*x
```

3) De

$$f''(x)=\frac{-2x}{\left(1+x^2\right)^2},$$

on déduit, en majorant la valeur absolue du numérateur et en minorant le dénominateur, la majoration de la dérivée seconde sur l'intervalle $[1,1,01]$

```
» M=2*1.01/(1+1)^2
M=0.50500000000000
```

4) On peut donc appliquer la formule de Taylor-Lagrange à l'ordre $n=1$ et l'inégalité du paragraphe 2.7.1.1 pour avoir une majoration de l'erreur lorsqu'on confond la valeur de $v=f(b)=\arctan(1,01)$ avec

$$\begin{aligned}\overline{v} &= f(a)+(b-a)f'(a)\\ &= \arctan(1)+(1,01-1)\times 0,5\end{aligned}$$

On a

$$|v-\overline{v}|\leqslant\frac{M\,|h|^2}{2!}.$$

Avec *Matlab*, on obtient

```
» vBarre=atan(1)+1/2*0.01
vBarre=0.79039816339745
» MajErreur=M*(1.01-1)^2/2
MajErreur=2.525000000000004e-005
```

2.7.1.3. *Formule de Mac-Laurin*

On se place dans le cas où l'intervalle $[a, b]$ est de la forme $[0,\ x]$, x étant une variable positive réelle quelconque. Alors la formule de Taylor-Lagrange à l'ordre n devient :

$$\left\| \begin{array}{l} f(x) - f(0) \\ = xf'(0) + \dfrac{1}{2!}x^2 f''(0) + ... \dfrac{1}{n!}x^n f^{(n)}(0) + \dfrac{1}{(n+1)!}x^{n+1} f^{(n+1)}(c), \end{array} \right.$$

où $c \in]0,\ x[$.

2.7.1.4. *Formule de Taylor-Young*

Considérons maintenant le cas où f est une application d'un intervalle I vers $\mathbb{R}$, et soit a et x deux points de I. Alors on montre qu'il existe une fonction ε définie au voisinage de a telle que

$$\left\| \begin{array}{l} f(x) \\ = \displaystyle\sum_{k=0}^{n} \frac{(x-a)^k}{k!} f^{(k)}(a) + (x-a)^n \varepsilon(x) \end{array} \right.$$

avec

$$\varepsilon(x) \to 0 \quad \text{pour } x \to a.$$

Le terme $(x-a)^n \varepsilon(x)$ est appelé reste d'ordre n pour $x \to a$.

Noter qu'à l'opposé des autres formules données précédemment, cette dernière précise le comportement du reste d'ordre n pour x tendant vers a.

Dans le cas particulier où $a = 0$, on obtient

$$\left\| \begin{array}{l} f(x) = \displaystyle\sum_{k=0}^{n} \frac{x^k}{k!} f^{(k)}(0) + x^n \varepsilon(x) \\ \qquad = f(0) + xf'(0) + \dfrac{1}{2!}x^2 f''(0) + ... \dfrac{1}{n!}x^n f^{(n)}(0) + x^n \varepsilon(x) \end{array} \right.$$

On peut noter, en utilisant les notations de Landau

$$x^n \varepsilon(x) = o\left(x^n\right) \qquad (x \longrightarrow 0)$$

et

$$f(x) = f(0) + xf'(0) + \frac{1}{2!}x^2 f''(0) + ...\frac{1}{n!}x^n f^{(n)}(0) + o(x^n).$$

On dira que

$$f(0) + xf'(0) + \frac{1}{2!}x^2 f''(0) + ...\frac{1}{n!}x^n f^{(n)}(0),$$

est le **développement limité** (en abrégé **D.L.**) d'ordre n de f au voisinage de 0.

Les **D.L.** sont très utiles pour l'étude **locale** des fonctions puisqu'ils permettent :

- *une expression plus simple de f, (au voisinage du point),*
- *une recherche facile de limites,*
- *un tracé plus précis de la courbe (recherche d'asymptotes, position de la courbe par rapport à celles-ci,...).*

2.7.2. *Exemples de calculs de D.L.*

On va donner le D.L. à l'ordre 2 et au voisinage de 0 de la fonction définie par

$$\sqrt{1+x}\sin(x).$$

On utilise la fonction **taylor**(f, n, a) de *Matlab* qui donne le développement de Taylor, au point a, **mais à l'ordre** $n-1$.

```
» syms x real
» taylor(sqrt(1+x)*sin(x),3,0)
ans=x+1/2*x^2
```

Le D.L. de $\sqrt{1+x}\sin(x)$ à l'ordre 2 est donc

$$x + \frac{1}{2}x,$$

on écrira aussi que

$$\sqrt{1+x}\sin(x) = x + \frac{1}{2}x + x^2.\varepsilon(x).$$

avec

$$\varepsilon(x) \to 0 \quad \text{pour} \quad x \to 0.$$

On affiche le D.L. à l'ordre 2 au voisinage de 1 de la fonction e^x

```
» syms x real
» taylor(exp(x),3,1)
ans=exp(1)+exp(1)*(x-1)+1/2*exp(1)*(x-1)^2
```

Ainsi

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

avec

$$\varepsilon(x-1) \to 0 \quad \text{pour } x \to 1.$$

2.7.3. *Application des D.L.*

2.7.3.1. *Calcul de limites*

Calculons

$$\lim_{x\to 0} \frac{e^{\sin x} - e^{\tan x}}{\sin x - \tan x}.$$

Cette limite se présente sous la forme indéterminée $\dfrac{0}{0}$.

On cherche des développements limités à un ordre suffisant, pour lever l'indétermination. Ici l'ordre 2 ne suffit pas. En effet

```
» syms x real
» % Au numérateur :
» U = taylor(exp(sin(x)),3,0)
» % l'ordre est 3-1=2.
U= 1+x+1/2*x^2
» V =taylor(exp(tan(x)),3,0)
V = 1+x+1/2*x^2
» % Par différence :
» U -V
ans = 0
» % De même, au dénominateur :
» W= taylor(sin(x)-tan(x),3,0)
ans = 0
```

A l'ordre 3, on a

```
» syms x real
» % Au numérateur :
» U = taylor(exp(sin(x)),4,0)
U= 1+x+1/2*x^2
» V =taylor(exp(tan(x)),4,0)
V = 1+x+1/2*x^2+1/2*x^3
» % Par différence :
» U -V
ans = 1/2*x^3
» % De même, au dénominateur :
» W= taylor(sin(x)-tan(x),4,0)
ans = -1/2*x^3
```

d'où

$$\begin{aligned}\lim_{x\to 0}\frac{e^{\sin x}-e^{\tan x}}{\sin x-\tan x} &= \lim_{x\to 0}\frac{-x^3/2+x^3\varepsilon_1(x)}{-x^3/2+x^3\varepsilon_2(x)}\\ &= \lim_{x\to 0}\frac{-1/2+\varepsilon_1(x)}{-1/2+\varepsilon_2(x)}\\ &= 1.\end{aligned}$$

2.7.3.2. *Recherche d'asymptote oblique*

Trouver l'asymptote oblique à la courbe représentative de la fonction définie par

$$f(x)=\sqrt{1+x+x^2},$$

au voisinage de $+\infty$, en effectuant le changement de variable

$$\frac{1}{x}=t,$$

et en calculant le D.L. à un ordre convenable. Donner l'allure de cette courbe.

1) On vérifie avec *Matlab* que

$$\lim_{x\to+\infty} f(x)=+\infty,$$

```
» syms x real
» f= sqrt(1+x+x^2);
» limit(f,x,inf)
ans = inf
```

2) On calcule le D.L. à l'ordre 2 de

$$\frac{f(x)}{x},$$

en effectuant le changement de variable

$$\frac{1}{x} = t,$$

au point $t = 0$.

```
» g=f/x
g = 1+x+x^2)^(1/2)/x
» syms t real
» h = subs(g,x,1/t) ;
» h1 = taylor(h,3,0)
h1 = 1+1/2*t+3/8*t^2
```

Ainsi, en remplaçant t par $1/x$, on a, au voisinage de $+\infty$

$$\frac{f(x)}{x} = 1 + \frac{1}{2x} + \frac{3}{8x^2} + \frac{1}{x^2}\varepsilon\left(\frac{1}{x}\right),$$

puis

$$f(x) = x + \frac{1}{2} + \frac{3}{8x} + \frac{1}{x}\varepsilon\left(\frac{1}{x}\right).$$

L'asymptote oblique est donc la droite d'équation

$$y = x + \frac{1}{2}$$

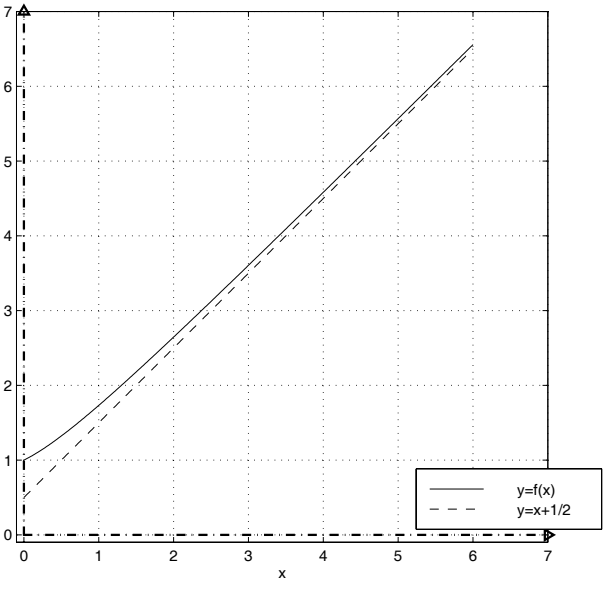
et la courbe représentative de f est au-dessus de cette asymptote au voisinage de $+\infty$, car

$$\frac{3}{8x} + \frac{1}{x}\varepsilon\left(\frac{1}{x}\right) = \frac{1}{x}\left(\frac{3}{8} + \varepsilon\left(\frac{1}{x}\right)\right) \geqslant 0$$

au voisinage de $+\infty$.

3) On construit la courbe et son asymptote

```
» ezplot(f,0,6)
» hold on ;grid on
» g=x+1/2
» set(gca,'LineStyle','–')
» ezplot(g,0,6)
» axis auto ; axis equal
» legend ('y=f(x)','y=x+1/2',4)
» title(' ')
» dessineRepere
```



2.8. Exercices

2.8.1. *Bijection réciproque*

On donne la fonction définie de $\mathbb{R}$ dans $]-1,1[$ par

$$f(x) = \frac{x}{1+|x|}.$$

Montrer qu'elle est bijective et donner sa fonction réciproque.

(solution p. 79)

2.8.2. *Etude de fonction et construction de courbe*

On considère la fonction f définie sur $[-a,\ a]$ $(a > 0)$ par

$$f(x) = \frac{x}{a}\sqrt{a^2 - x^2}.$$

1) Montrer que la fonction f est impaire.

2) Etudier les variations de f sur $[0, a]$.

3) Préciser les tangentes à la courbe représentative aux points d'abscisses 0 et a.

4) Construire la courbe représentative de f pour $a = 4$.

5) En déduire le tracé de la courbe d'équation

$$a^2y^2 = x^2(a^2 - x^2).$$

(solution p. 80)

2.8.3. *Etude d'une fonction périodique*

Soit g la fonction définie sur $\mathbb{R}$ par $g(x) = x - E(x)$ où $E(x)$ désigne la partie entière de x définie de la manière suivante :

$E(x)$ est **le plus grand entier** tel que

$$E(x) \leqslant x.$$

Vérifier que g est périodique de période 1 et donner son graphe.

(solution p. 82)

2.8.4. *Fonction trigonométrique inverse*

On considère la fonction f définie par

$$f(x) = \arccos(\cos x).$$

1) Montrer que f est définie sur $\mathbb{R}$.

2) Utiliser *Matlab* pour représenter graphiquement f sur $[-3\pi,\ 3\pi]$.

3) Montrer que si $x \in [0,\ \pi]$, $f(x) = x$.

4) Montrer que f est paire et 2π-périodique et en déduire l'expression de $f(x)$ pour x appartenant à $[-\pi,\ 0]$, puis à

$$[2k\pi - \pi,\ 2k\pi + \pi[, \qquad k \in \mathbb{Z}.$$

5) Vérifier graphiquement

(solution p. 84)

2.8.5. *D.L. et étude de limite (1)*

On propose de calculer la limite pour $x \to 1$ de la fonction définie par

$$f(x) = \frac{\sqrt{2x - x^4} - \sqrt[3]{x}}{1 - \sqrt[4]{x^3}} = \frac{f_1(x) - f_2(x)}{1 - f_3(x)}.$$

On pose $x = 1 + h$.

1) Donner les D.L. à l'ordre 1 et au voisinage de $h \to 0$ des expressions

$$f_1(1+h), \quad f_2(1+h), \quad f_3(1+h),$$

2) En déduire la limite.

(solution p. 86)

2.8.6. *D.L. et recherche d'asymptote*

1) Déterminer l'asymptote oblique (pour $x \to \infty$) à la courbe représentative de la fonction donnée par

$$f(x) = (x+1)e^{1/x}.$$

On posera $t = 1/x$ et on effectuera un D.L. de l'exponentielle à l'ordre 2 au voisinage de 0.

2) Préciser la position de la courbe par rapport à cette asymptote au voisinage de $+\infty$..

3) Tracer son graphe.

(solution p. 87)

2.8.7. *D.L. et étude de limite (2)*

Soit

$$f(x) = \frac{\sin(\tan(x)) + \sin(x) - 2x}{x^5}.$$

1) Un D.L. à l'ordre 4 au voisinage de 0 suffit-il pour calculer la limite de cette expression pour $x \to 0$?

2) Calculer cette limite. Donner l'allure de cette fonction au voisinage de 0.

(solution p. 88)

2.9. Solutions

Exercice 2.8.1

Pour montrer que la fonction est injective on part de

$$f(x_1) = \frac{x_1}{1+|x_1|} = f(x_2) = \frac{x_2}{1+|x_2|}$$

qui implique nécessairement que x_1 et x_2 sont de même signe. On a donc, pour x_1 et x_2 positifs :

$$D = \frac{x_1}{1+x_1} - \frac{x_2}{1+x_2} = 0$$

```
» syms x1 x2 x y real
» FdeX1=x1/(1+x1);FdeX2=x2/(1+x2);
» D=FdeX1-FdeX2;
» simplify(D)
ans=-(-x1+x2)/(1+x1)/(1+x2)
```

la réponse implique que $x_1 = x_2$. On fait de même pour les variables négatives.

Pour montrer que f est surjective et trouver l'expression de l'application réciproque, on doit résoudre l'équation (en la variable x)

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

pour y quelconque dans $]-1,1[$. Le même raisonnement que précédemmment montre que si la donnée y est positive alors x l'est aussi. On considère donc deux cas :

```
» % Cas y positif
» solve('x/(1+x)=y')
ans=-y/(-1+y)
» % Cas y négatif
» solve('x/(1-x)=y')
ans=y/(1+y)
```

On déduit que f est surjective, et la fonction réciproque de f est l'application

$$\begin{aligned} g\ ]-1,+1[&\longrightarrow \mathbb{R} \\ y &\longmapsto x = \frac{y}{1-|y|}. \end{aligned}$$

Exercice 2.8.2

1) On définit f et on vérifie l'imparité avec *Matlab* .

```
» syms x real ; syms a positive
» fDeX = x/a * sqrt(a^2-x^2) ;
» % Imparité
» fDeMoinsX = subs(fDeX, x,-x)
fDeMoinsX = -x/a*(a^2-x^2)^(1/2)
» fDeMoinsX + fDeX
ans = 0
```

On a bien vérifié que $f(-x) = -f(x)$.

2) Pour étudier les variations de f, on étudie le signe de la dérivée sur l'intervalle $[0, a[$.

```
» fPrimeDeX = diff(fDeX) ;
» fPrimeDeX = factor(fPrimeDeX)
fPrimeDeX = -(-a^2+2*x^2)/a/(-(x-a)*(x+a))^(1/2)
```

La commande *numden* permet de séparer numérateur et dénominateur dans l'expression de $f'(x)$.

```
» [N,D] = numden(fPrimeDeX)
N = a^2-2*x^2
D = a*(a^2-x^2)^(1/2)
» S =solve(N)
S = [ 1/2*2^(1/2)*a]
  [ -1/2*2^(1/2)*a]
```

Ainsi sur $[0\ a[$,

$$f'(x) = \frac{a^2 - 2x^2}{a\sqrt{a^2 - x^2}} = \frac{(a - \sqrt{2}x)(a + \sqrt{2}x)}{a\sqrt{a^2 - x^2}}$$

s'annule pour $x = a\frac{\sqrt{2}}{2}$ et est positif pour

$$x < a\frac{\sqrt{2}}{2}.$$

On calcule les extrema, en utilisant *subs* :

```
» fDe0 = subs(fDeX,x,0)
fDe0 = 0
» fDeA = subs(fDeX,x,a)
fDeA = 0
» fMax = simplify(subs(fDeX,x, S(1)))
fMax = 1/2*a
```

D'où le tableau de variations

x	0		$a\frac{\sqrt{2}}{2}$		a
$f'(x)$		$+$	0	$-$	$\|\|$
$f(x)$	0	$\nearrow$	$\frac{a}{2}$	$\searrow$	0

3)

```
» fPrimeDe0 = simple(subs(fPrimeDeX,x,0))
fPrimeDe0 = 1
```

La tangente au point d'abscisse 0 a pour coefficient directeur

$$f'(0) = 1.$$

Pour obtenir la tangente au point d'abscisse a, on étudie la limite du taux d'accroissement de f en a à gauche :

```
» limit((fDeX-fDeA)/(x-a),x,a,'left')
ans = -inf
```

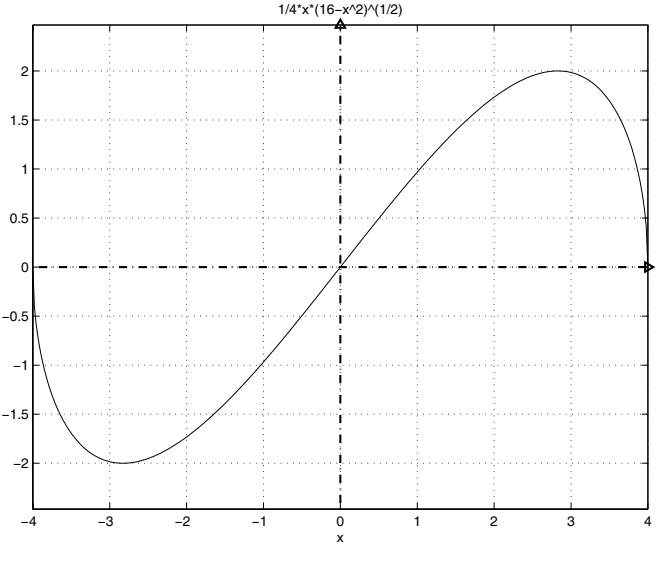
On a

$$\lim_{x \xrightarrow{>} a} \frac{f(x) - f(a)}{x - a} = -\infty,$$

et la courbe représentative de f admet une demi-tangente verticale au point $(a, 0)$.

4) La courbe représentative pour $a = 4$ s'obtient par :

```
» f4DeX =subs (fDeX, a,4)
» ezplot(f4DeX,-4,4)
» grid on ;dessineRepere
```



5) On résout l'équation d'inconnue y

$$a^2y^2 = x^2(a^2 - x^2)$$

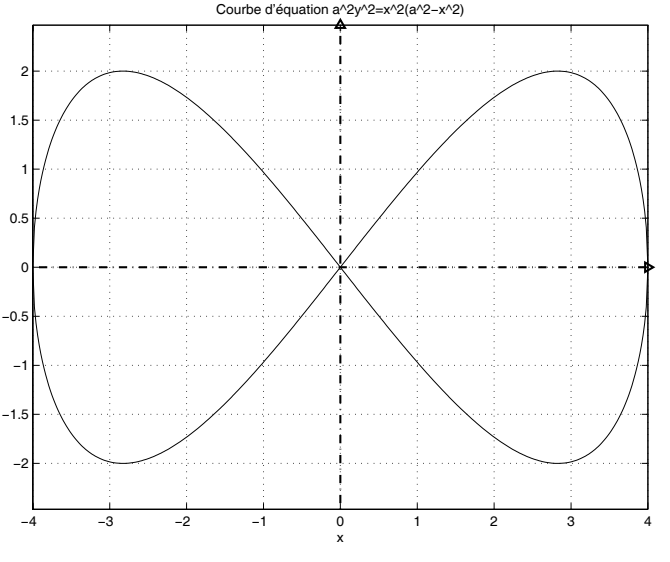
```
» syms y real
» E = a^2*y^2 - x^2*(a^2-x^2);
» Se = solve(E,y)
Se = [ x/a*(a^2-x^2)^(1/2)]
   [ -x/a*(a^2-x^2)^(1/2)]
```

ce qui montre que

$$E = 0 \Longleftrightarrow (y = f(x) \text{ ou } y = -f(x))\,.$$

On peut donc compléter le tracé à l'aide de la courbe représentative de $-f$.

```
» hold on
» ezplot (-f4DeX, -4,4)
» title('Courbe d"équation a^2y^2=x^2(a^2-x^2)')
```



Exercice 2.8.3

Sous *Matlab*, la fonction mathématique $x \longmapsto E(x)$ est définie par **floor(x)**, qui ne permet pas d'effectuer de calcul symbolique :

```
» floor(pi)
ans = 3
» syms x real
» gDeX=x-floor(x)
? ? ? Function 'floor' not defined for values of class 'sym'.
```

Pour prouver la périodicité, posons $E(x) = n$. Ainsi

$$n = E(x) \leqslant x < n + 1,$$

d'où

$$n + 1 \leqslant x + 1 < n + 2,$$

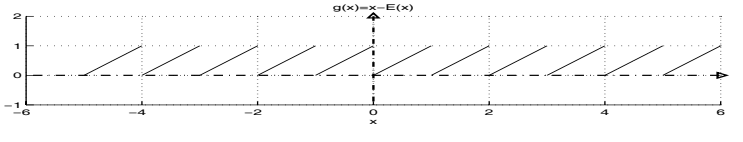
donc $E(x + 1) = n + 1$. On a alors

$$g(x + 1) = (x + 1) - (n + 1) = x - n = g(x).$$

d'où la périodicité de g.

Pour le graphe, on effectue le tracé sur les intervalles $[i, i + 1[$

```
» syms x n real
» gDeX=x-n ;
» clf
» hold on
» for i=-5 :5,
    fi=subs(gDeX,n,i) ;
    ezplot(fi,i,i+1) ;
end
» axis [-6 6 -1 1] ;axis equal ; grid on
» title('g(x)=x-E(x)')
» dessineRepere
```

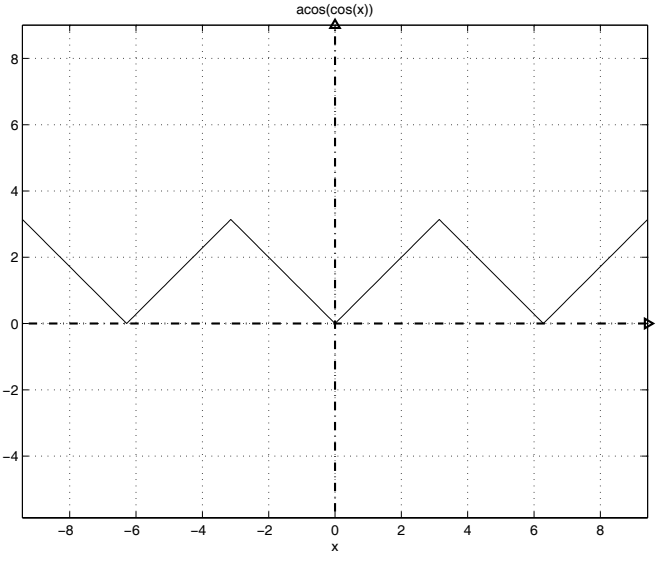


Exercice 2.8.4

1) Pour tout réel x, $\cos x \in [-1,\ 1]$, donc $\arccos(\cos x)$ est définie.

2) On utilise *ezplot* pour représenter graphiquement f sur $[-3\pi,\ 3\pi]$:

```
» syms x real
» fDeX=acos(cos(x))
» set(gca,'LineStyle','- -')
» ezplot(fDeX,-3*pi,3*pi)
» hold on ;grid on
» axis equal
» T=axis ;     % pour mémoriser Xmin, Xmax, Ymin, Ymax
» dessineRepere
```



3) Par définition, la fonction $\arccos$ est la réciproque de la fonction $\cos$, restreinte à $[0,\ \pi]$. Donc on a , pour tout $x \in [0,\ \pi]$

$$f(x) = \arccos(\cos x) = x.$$

4) La fonction $\cos$ étant paire et $2\pi-$périodique, on a :

$$f(-x) = \arccos(\cos(-x)) = \arccos(\cos x) = f(x)$$

et

$$f(x + 2\pi) = \arccos(\cos(x + 2\pi)) = \arccos(\cos x) = f(x).$$

Matlab donne aussi ces résultats

```
» fDeMoinsX=simplify(subs(fDeX,x,-x))
fDeMoinsX = acos(cos(x))
» fDeXplus2PI=simplify(subs(fDeX,x,x+2*pi))
fDeXplus2PI = acos(cos(x))
```

Pour x appartenant à $[-\pi,\ 0]$, on a

$$-x \in [0,\ \pi],$$

et

$$f(x) = f(-x) = -x.$$

donc, pour x appartenant à l'intervalle $[-\pi,\ \pi]$,

$$f(x) = |x|\,.$$

Par périodicité, pour $k \in \mathbb{Z}$, et pour $x \in [2k\pi - \pi,\ 2k\pi + \pi[$, on a

$$f(x) = f(x - 2k\pi) = |x - 2k\pi|\,.$$

On remarquera que

$$2k\pi - \pi \leqslant x < 2k\pi + \pi$$

implique

$$k \leqslant \frac{x+\pi}{2\pi} < k + 1,$$

d'où

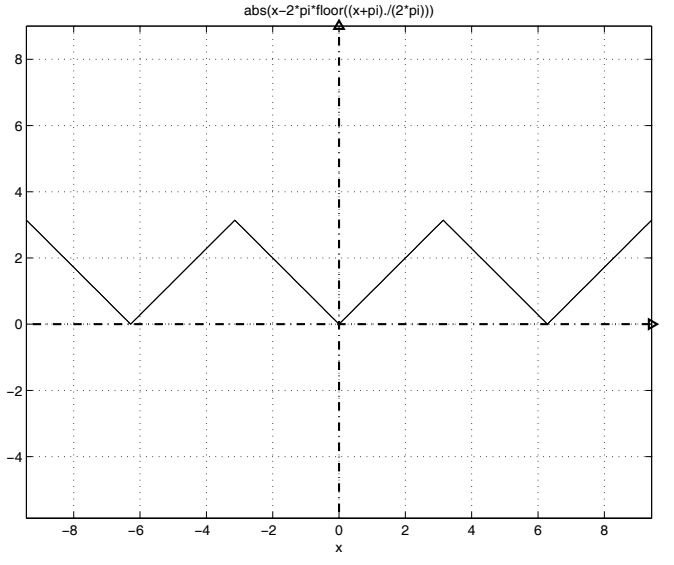
$$k = E\left(\frac{x+\pi}{2\pi}\right),$$

et, pour tout réel x,

$$f(x) = \left|x - 2\pi E\left(\frac{x+\pi}{2\pi}\right)\right|.$$

5) On utilise cette dernière expression pour représenter graphiquement f

```
» set(gca,'LineStyle','-')
» ezplot('abs(x-2*pi*floor((x+pi)./(2*pi)))',[-3*pi 3*pi])
» axis(T)
```



Exercice 2.8.5.

1) On calcule les trois expressions $f_i(1+h)$ $(i=1,2,3)$ comme fonctions de h :

```
» syms x h real
» f1=sqrt(2*x-x^4);f2=x^(1/3); f3=x^(3/4);
» f=(f1-f2)/(1-f3)
f =((2*x-x^4)^1/2)-x^(1/3))/(1-x^(3/4))
» g1=subs(f1,x,1+h);
» g2=subs(f2,x,1+h);
» g3=subs(f3,x,1+h);
```

On donne les D.L. à l'ordre demandé :

```
» Tg1=taylor(g1,h,0,2)
Tg1 =1-h
» Tg2=taylor(g2,h,0,2)
Tg2 =1+1/3*h
» Tg3=taylor(g3,h,0,2)
Tg3 =1+3/4*h
```

2) On en déduit que pour $h \to 0$, on a

$$\begin{aligned} f(1+h) &= \frac{1-h-(1+h/3)+h.\varepsilon_1(h)}{1-1-3h/4+h.\varepsilon_2(h)} \\ &= \frac{-4/3+\varepsilon_1(h)}{-3/4+\varepsilon_2(h)}, \end{aligned}$$

d'où la limite cherchée :

```
» rapport=(Tg1-Tg2)/(1-Tg3)
rapport =16/9
```

On peut effectuer un calcul direct de limite avec *Matlab* :

```
» limit(f,x,1)
ans =16/9
```

Exercice 2.8.6

1) Le changement de variable (permettant l'analyse au voisinage de zéro) donne

```
» syms x t real
» f=(x+1)*exp(1/x);
» g=subs(f,x,1/t)
g =(1/t+1)*exp(t)
```

d'où :

```
» Texp=taylor(exp(t),3)
Texp =1+t+1/2*t^2
```

En développant et en revenant à la variable initiale, on a :

```
» g1=expand((1/t+1)*Texp)
g1 =1/t+2+3/2*t+1/2*t^2
» f1=expand(subs(1/t+2+3/2*t,t,1/x))
f1 =x+2+3/2/x
```

d'où l'équation de la droite asymptote $y = x + 2$.

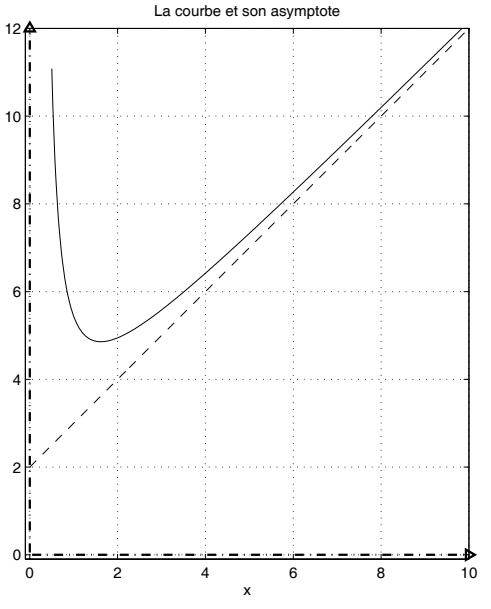
2) La courbe est au-dessus de cette droite car pour $x \to \infty$:

$$f(x) - (x+2) = \frac{3}{2x} + \frac{1}{x}.\varepsilon\left(\frac{1}{x}\right) = \frac{1}{x}\left(\frac{3}{2} + \varepsilon\left(\frac{1}{x}\right)\right).$$

Lorsque x tend vers $+\infty$, $f(x) - (x+2)$ tend vers 0, par valeurs positives.

3) On construit la courbe et son asymptote

```
» ezplot(f,0.5,10)
» hold on ;grid on
» set(gca,'LineStyle','–')
» ezplot(x+2,0,10)
» axis equal ;axis ([-0.1 10 -0.1 12])
» dessineRepere
» title('La courbe et son asymptote')
```



Exercice 2.8.7

La fonction est définie sur $]-\pi/2,\ 0[\cup]0,\ +\infty[$.

A l'ordre 4, le D.L du numérateur ne permet pas de conclure (il donne une indétermination) :

```
» syms x real
» num = sin(tan(x))+sin(x)-2*x;
» Tnum4=taylor(num,5)
Tnum4 = 0
```

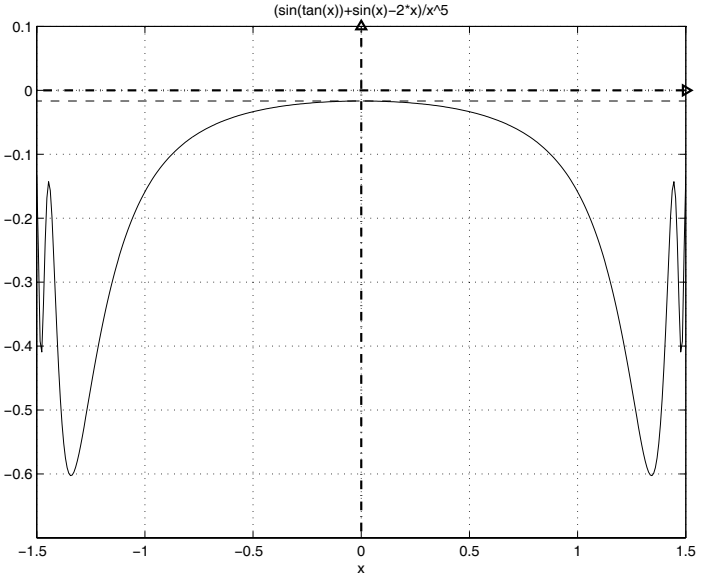
A l'ordre 5 on a

```
» Tnum5=taylor(num,6)
Tnum5 = -1/60*x^5
```

et donc la limite est $-1/60$. On représente graphiquement cette fonction :

```
» ezplot(num/x^5,-2,2)
» hold on ;grid on
» plot([-2 2],[-1/60 -1/60],'–')
» axis([-1.5 1.5 -0.7 0.1])
» dessineRepere
```

On obtient



Le graphe au voisinage de zéro montre bien que la fonction est assez plate et se comporte comme $-1/60$.

Chapitre 3

Intégration

3.1. Intégrale de Riemann

3.1.1. *Définitions*

3.1.1.1. *Sommes de Darboux et intégrale de Riemann*

On donne une fonction numérique f **continue, monotone et positive** sur un intervalle $[a, b]$. On considère une subdivision de cet intervalle à pas équidistants :

$$a,\ a+h,\ a+2h, ..., a+ih, ..., a+nh,$$

où $n \in \mathbb{N}^*$ et

$$h = (b-a)/n.$$

On appelle sommes de Darboux les quantités :

$$\begin{cases} S_n = \dfrac{b-a}{n} \displaystyle\sum_{k=1}^{k=n} f(a+kh) \\ s_n = \dfrac{b-a}{n} \displaystyle\sum_{k=1}^{k=n} f(a+(k-1)h). \end{cases}$$

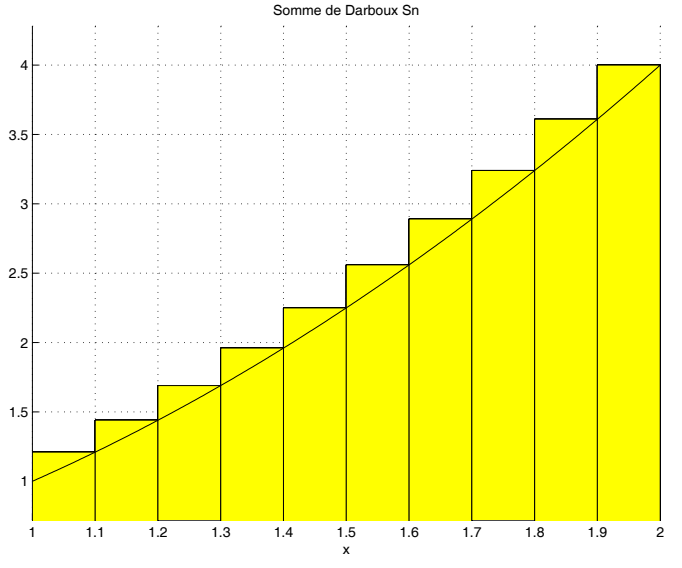
On montre que pour une telle fonction f, les suites $(s_n)_{n\geqslant 1}$et $(S_n)_{n\geqslant 1}$ forment deux suites adjacentes (l'une croissante, l'autre décroissante) convergeant vers une même limite qui est égale à l'aire A de la surface délimitée par la courbe représentative de f (dans un repère orthonormé), l'axe des x et les droites d'équations

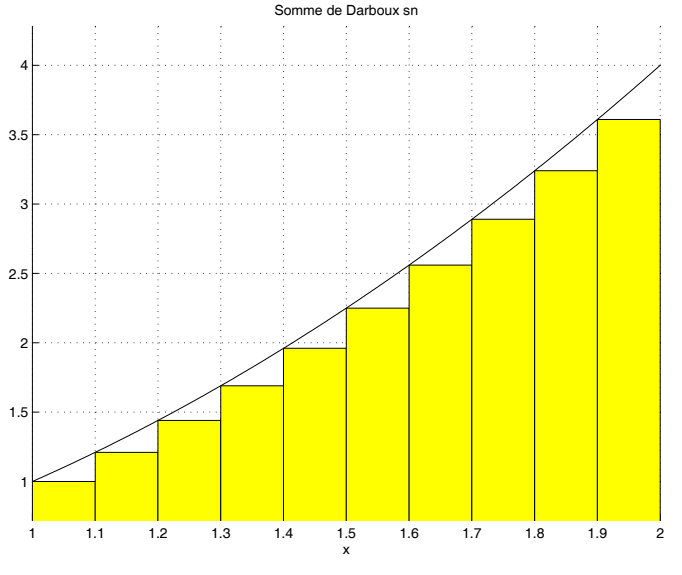
$$x = a, \quad x = b.$$

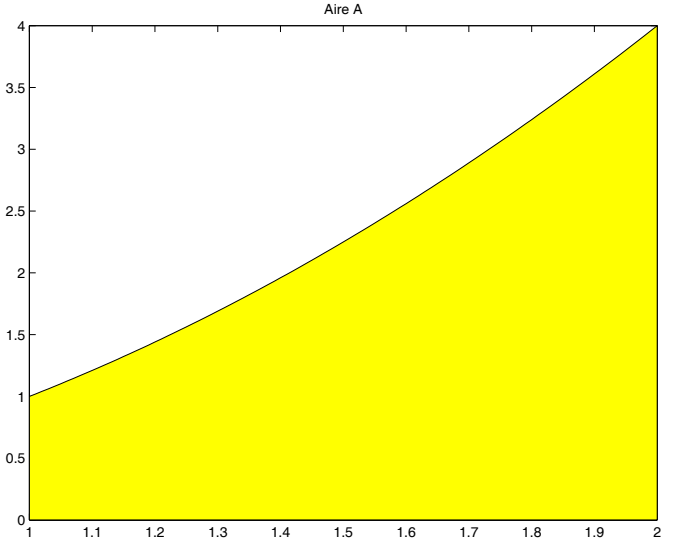
On a l'inégalité (dans le cas où f est croissante)

$$s_n < A < S_n.$$

Graphiquement, $(b-a)\,/n.f(a+kh)$ représente l'aire d'un rectangle de largeur h et de hauteur $f(a+kh)$, et on dessine ci-dessous les surfaces correspondant à la réunion de ces rectangles, d'aires respectives S_n et s_n, ainsi que la surface d'aire A.







On posera par définition

$$\int_a^b f(x)dx = \lim_{n\longrightarrow\infty} S_n = \lim_{n\longrightarrow\infty} s_n = A,$$

et on dira que la fonction f est **intégrable** (au sens de Riemann) entre a et b.

Il est important de remarquer dès maintenant que **la variable à l'intérieur de l'intégrale** peut être notée d'une manière quelconque x, t,... variant entre a et b.

3.1.1.2. *Intégrale d'une fonction continue*

On admet qu'on peut aussi construire l'intégrale dans le cas où f est seulement continue sur $[a, b]$:

Théorème.
Toute fonction continue est intégrable sur $[a, b]$.

3.1.1.3. *Généralisation*

De même, on peut aussi considérer le cas où f présente un nombre fini de discontinuités sur $[a, b]$, c'est-à-dire qu'en un certain nombre de points

$$x_0, x_1, ..., x_p$$

de $[a, b]$, les limites à gauche et à droite

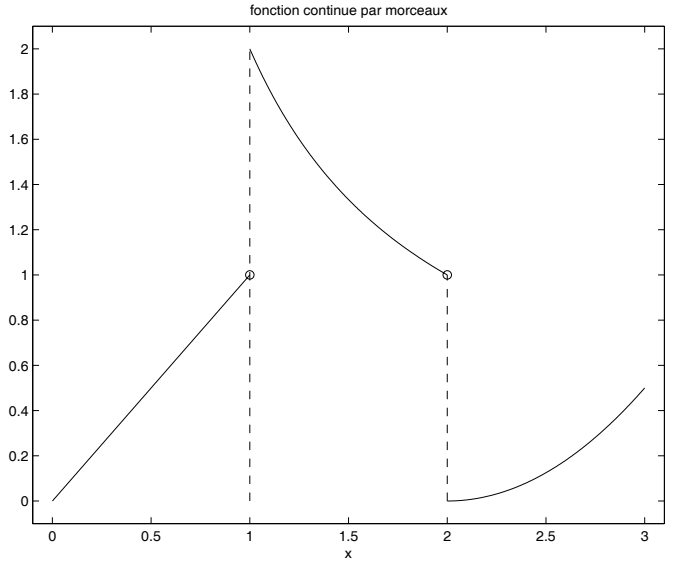
$$\lim_{x \xrightarrow{<} x_i} f(x_i), \qquad \lim_{x \xrightarrow{>} x_i} f(x_i), \qquad i = 1, 2, ..., p$$

existent et ne sont pas nécessairement égales. On dit alors que f est continue par morceaux.

Le résultat concernant le cas général des **fonctions bornées** définies sur l'intervalle $[a, b]$ est donné par le :

Théorème.
Toute fonction bornée et monotone par morceaux est intégrable sur $[a, b]$.

On verra que de nombreuses fonctions concrètes (comme certains signaux) sont de cette nature. Ci-dessous, un exemple.



3.1.1.4. *Définition*

Si $b < a$ on pose par définition

$$\int_a^b f(x)dx = -\int_b^a f(x)dx,$$

et si $a = b$

$$\int_a^a f(x)dx = 0.$$

3.1.2. *Exemple*

On peut, sur un exemple effectuer ce calcul de sommes de Darboux. Considérons la fonction définie par

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

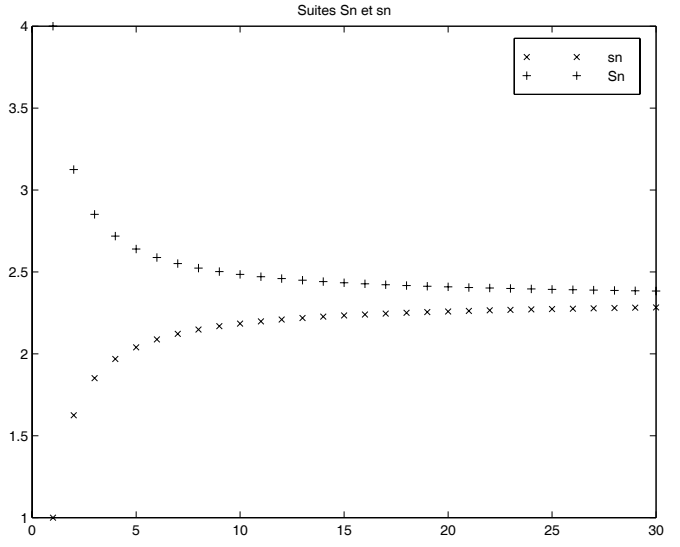
sur l'intervalle $[1,\ 2]$.

```
» a=1 ; b=2 ;
» syms x
» f=x^2 ;
» syms n k real
» h=(b-a)/n ;
» Sn=h*symsum(subs(f,x,a+k*h),k,1,n)
Sn = 1/n*(n+1/n*(n+1)^2-(n+1)/n+
      1/3/n^2*(n+1)^3-1/2/n^2*(n+1)^2+1/6/n^2*(n+1))
» sn=h*symsum(subs(f,x,a+(k-1)*h),k,1,n)
sn = 1/n*(n-3*(n+1)/n+13/6/n^2*(n+1)+
      1/n*(n+1)^2-3/2/n^2*(n+1)^2+1/3/n^2*(n+1)^3+2/n-1/n^2)
```

(On a utilisé la commande **symsum(sk,k,1,n)** qui permet le calcul symbolique de la somme des n premiers termes de la suite s_k).

On représente graphiquement les deux suites s_n et S_n, pour $n \in [1,\ 30]$:

```
» N=1 :1 :30 ;
» sN=double(subs(sn,N)) ;
» SN=double(subs(Sn,N)) ;
» plot(sN,'x')
» hold on
» plot(SN,'+')
» title('Suites Sn et sn')
» legend('sn','Sn')
```



On peut aussi calculer la limite de ces deux suites :

```
» limit(sn,n,inf,'left')
ans =7/3
» limit(Sn,n,inf,'left')
ans =7/3
```

On retrouve bien

$$\int_1^2 x^2 dx = \frac{7}{3}.$$

3.1.3. *Propriétés générales*

Les propriétés essentielles de cette intégrale sont :

1) relation de Chasles : pour tout $c \in [a, b]$,

$$\int_a^b f(x)dx = \int_a^c f(x)dx + \int_c^b f(x)dx,$$

2) linéarité : pour tous réels λ, μ,

$$\int_a^b (\lambda f + \mu g)(x)dx = \lambda \int_a^b f(x)dx + \mu \int_a^b f(x)dx,$$

3) positivité (pour $b > a$) :

$$f \geqslant 0 \quad \Longrightarrow \int_a^b f(x)dx \geqslant 0,$$

4) majoration (pour $b > a$) :

$$\left| \int_a^b f(x)dx \right| \leqslant \int_a^b |f(x)|\, dx,$$

5) formule dite de la moyenne lorsque f est **continue** :

$$\exists c \in]a, b[\ : \quad \int_a^b f(x)dx = (b-a)f(c).$$

3.2. Primitive d'une fonction

3.2.1. *Cas d'une fonction continue*

Soit f une fonction continue sur $[a, b]$. Posons, pour $t \in [a, b]$

$$F(t) = \int_a^t f(x)dx.$$

On montre grâce à la formule de la moyenne ci-dessus que F est **la primitive** de f qui s'annule en a. Autrement dit,

$$F'(t) = f(t) \quad \text{et} \quad F(a) = 0.$$

Si G est une primitive quelconque de f alors

$$\int_a^b f(x)dx = G(b) - G(a).$$

Cette différence se note usuellement

$$[G(x)]_a^b .$$

Par exemple, on a

$$\int_0^{\pi/2} (1 - \sin x)dx = [x + \cos x]_0^{\pi/2} = \frac{\pi}{2} - 1.$$

3.2.2. *Cas d'une fonction intégrable quelconque*

Lorsque f est seulement intégrable , on montre que la fonction F est encore **continue,** mais non nécessairement dérivable.

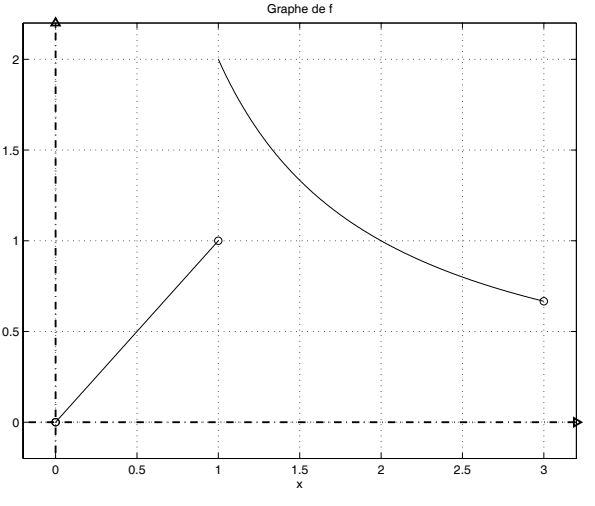
Exemple

Soit f la fonction définie sur $[0,\ 3]$ par

$$\begin{cases} f(x) = x & \text{si } x \in [0,\ 1] \\ f(x) = 2/x & \text{si } x \in\ ]1,\ 3]\,. \end{cases}$$

La représentation graphique s'obtient par :

```
» clf
» ezplot('x',0,1);
» hold on
» ezplot('2/x',1,3);
»axis([-0.2 3.2 -0.2 2.2])
» plot(0,0,'o')
» plot(1,1,'o')
» plot(3,2/3,'o')
» grid on
» dessineRepere
» title('Graphe de f')
```



Calculons

$$F(t) = \int_0^t f(x)dx.$$

– Pour $0 \leqslant t \leqslant 1$, on a

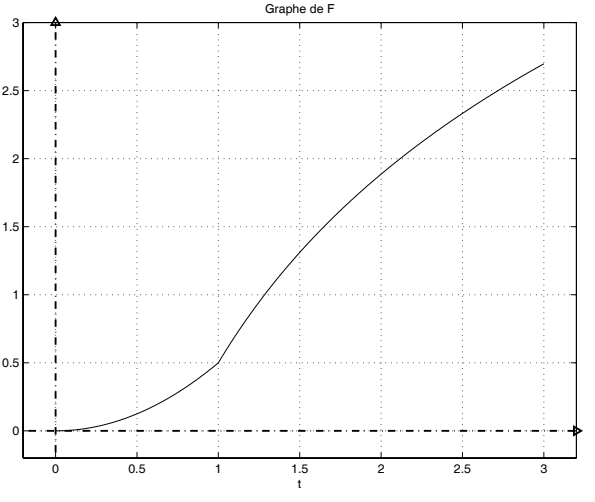
$$F(t) = \int_0^t x dx = \left[\frac{x^2}{2}\right]_0^t = \frac{t^2}{2}.$$

– Pour $1 < t \leqslant 3$, la relation de Chasles donne

$$F(t) = \int_0^1 x dx + \int_1^t \frac{2}{x} dx = \frac{1}{2} + [2 \ln x]_0^t = \frac{1}{2} + 2 \ln t.$$

On construit le graphe de F :

```
» clf
» ezplot('t^2/2',0,1);
» hold on
» ezplot('1/2+2*log(t)',1,3);
» axis([-0.2 3.2 -0.2 3])
» title('Graphe de F')
» grid on
» dessineRepere
```



F est continue en 1 car

$$F(1) = \frac{1}{2} = \lim_{t \xrightarrow{<} 1} \frac{t^2}{2} = \lim_{t \xrightarrow{>} 1} \left(\frac{1}{2} + 2 \ln t\right),$$

mais F n'est pas dérivable en 1 : on peut l'observer sur le graphique et le vérifier en calculant les limites à gauche et à droite du taux d'accroissement :

$$\frac{F(t) - F(1)}{t - 1}.$$

```
» syms t real
» F0=t^2/2 ;
» F1=1/2+2*log(t) ;
» Fde1=subs(F0,t,1)
Fde1 =1/2
» limitG=limit((F0-Fde1)/(t-1),t,1,'left')
limitG =1
» limitD=limit((F1-Fde1)/(t-1),t,1,'right')
limitD =2
```

Ainsi

$$\lim_{t \overset{<}{\longrightarrow} 1} \frac{F(t)-F(1)}{t-1} \neq \lim_{t \overset{>}{\longrightarrow} 1} \frac{F(t)-F(1)}{t-1}.$$

3.2.3. *Notation*

On conviendra par la suite que la notation (imprécise)

$$\int f(x)dx,$$

désignera une primitive quelconque de f. On devra faire attention à l'usage de cette notation en particulier par rapport à la variable des primitives considérées et l'intervalle de travail.

3.3. Calcul intégral

3.3.1. *Calcul intégral avec Matlab*

La commande **int** permet de calculer :

– des intégrales définies : on détermine par exemple la valeur de

$$I = \int_{-1}^{1} \frac{\sqrt{1-x}}{2+x} dx$$

en utilisant les commandes :

```
» syms x real
» I= int(sqrt((1-x)/(2+x)),-1,1)
I = 3/4*pi-2^(1/2)+3/2*asin(1/3)
» double(I)
ans = 1.4517
```

d'où

$$I = \frac{3}{4}\pi - \sqrt{2} + \frac{3}{2}\arcsin(1/3) \simeq 1,4517.$$

– des primitives : on obtient

$$F(x) = \int \frac{x}{\sqrt{1+x}}dx,$$

par

```
» syms x real
» FdeX=int(x*sqrt(1+x))
FdeX = -2/3*(1+x)^(3/2)+2/5*(1+x)^(5/2)
```

Ainsi

$$F(x) = -\frac{2}{3}(1+x)^{3/2} + \frac{2}{5}(1+x)^{5/2}.$$

On peut, dans l'utilisation de *int*, préciser la variable d'intégration :

```
» syms t x
» int((t*x+1)/(t^2+1),t)
ans = 1/2*x*log(t^2+1)+atan(t)
» int((t*x+1)/(t^2+1),x)
ans = 1/(t^2+1)*(1/2*t*x^2+x)
```

ce qui montre que

$$\int \frac{tx+1}{t^2+1}dt = \frac{1}{2}x\ln\left(t^2+1\right) + \arctan t,$$

alors que

$$\int \frac{tx+1}{t^2+1}dx = \frac{1}{t^2+1}\left(\frac{1}{2}tx^2 + x\right).$$

3.3.2. *Changement de variable*

3.3.2.1. *Cas d'une intégrale définie*

Soit f une fonction continue sur $[a,b]$. Dans l'intégrale

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

on peut effectuer un **changement de variable :**

$$\varphi : [\alpha,\beta] \to [a,b],\ t \mapsto x = \varphi(t),$$

où $\varphi([\alpha,\beta]) \subset [a,b]$, et φ est continûment dérivable sur $[\alpha,\beta]$. En effet, on peut former les fonctions

$$f \circ \varphi \text{ et } F \circ \varphi$$

où F est une primitive de f dans $[a,b]$. En dérivant, on obtient

$$(F' \circ \varphi).\varphi' = (f \circ \varphi).\varphi',$$

donc $F \circ \varphi$ est une primitive de $(f \circ \varphi).\varphi'$.

Si $\varphi(\alpha) = a$ et $\varphi(\beta) = b$, on a

$$\int_a^b f(x)dx = \int_\alpha^\beta f(\varphi(t))\varphi'(t)dt.$$

3.3.2.2. *Exemple*

Calculer

$$\int_1^2 \frac{e^x}{1+e^x}dx,$$

en effectuant le changement de variable

$$x = \varphi(t) = \ln(t).$$

La démarche à suivre est :

1) on définit l'intervalle $[\alpha,\beta]$: à la valeur $t = e$, correspond $x = \varphi(e) = 1$, et à la valeur $t = e^2$, correspond $x = 2$. De plus on a

$$\ln\left(\left[e,\ e^2\right]\right) = \left[1,\ 2\right],$$

2) on vérifie que la fonction $\varphi = \ln$ est continument dérivable,

3) on calcule dx, ce qui revient en fait à calculer $\varphi'(t)$:

$$dx = \varphi'(t)dt = \frac{dt}{t},$$

4) on peut alors calculer l'intégrale :

$$\begin{aligned}\int_1^2 \frac{e^x}{1+e^x}dx &= \int_e^{e^2} \frac{t}{1+t}\frac{dt}{t} = \int_e^{e^2} \frac{dt}{1+t} \\ &= [\ln(1+t)]_e^{e^2} = \ln\frac{1+e^2}{1+e}.\end{aligned}$$

Sous *Matlab* on a :

– par calcul direct de l'intégrale :

```
» syms x real
» FdeX=exp(x)/(1+exp(x));
» int(FdeX,1,2)
ans=log(1+exp(2))-log(1+exp(1))
```

– par le changement de variable :

```
» syms t real
» GdeT=subs(FdeX,x,log(t))
exp(log(t))/(1+exp(log(t)))
» B= int(GdeT*diff(log(t)),sym('exp(1)'),sym('exp(2)'))
B = log(1+exp(2))-log(1+exp(1))
```

3.3.2.3. *Cas d'une intégrale indéfinie*

On a vu précédemment que $F \circ \varphi$ est une primitive de $(f \circ \varphi).\varphi'$. D'où :

$$\int f(\varphi(t))\varphi'(t)dt = (F \circ \varphi)(t) + Cste.$$

En particulier

$$\int a.f(at+b)dt = F(at+b) + Cste.$$

3.3.3. *Intégration par parties*

En intégrant la dérivée d'un produit

$$(u.v)'(x) = u'(x)v(x) + u(x)v'(x)$$

où u et v sont deux fonctions supposées continûment dérivables sur $[a, b]$, on obtient les formules d'intégration par parties :

$$\int u(x)v'(x)dx = u(x)v(x) - \int u'(x)v(x)dx$$

et

$$\int_a^b u(x)v'(x)dx = [u(x)v(x)]_a^b - \int_a^b u'(x)v(x)dx.$$

L'emploi de ces formules est utile lorsqu'on intègre un produit de fonctions telles que

$$P(x)e^x,\ P(x)\ln(x),\ P(x)\sin(x),\ \sin(x)e^x,\ P(x)Arc\tan(x),\ \text{etc,}$$

où P est une fonction polynomiale.

3.3.3.1. *Exemple*

Donner une primitive de la fonction $x \longmapsto x\exp(x)$.

On pose

$$\left|\begin{array}{l} u(x) = x \\ v'(x) = \exp(x) \end{array}\right| \quad \begin{array}{l} u'(x) = 1 \\ v(x) = \exp(x), \end{array}$$

d'où

$$\begin{aligned} \int x\exp(x)dx &= x\exp(x) - \int \exp(x)dx \\ &= x\exp(x) - \exp(x) + Cste. \end{aligned}$$

Avec *Matlab* :

```
» syms x
» u = x ;
» vPrime = exp(x) ;
» uPrime = diff(u)
uPrime = 1
» v = int(vPrime)
v = exp(x)
» I = u*v-int(uPrime*v)
I = x*exp(x)-exp(x)
```

On pouvait évidemment obtenir ce résultat directement par :

```
» int(x*exp(x))
ans = x*exp(x)-exp(x)
```

3.4. Décomposition en éléments simples

Nous avons inséré volontairement cette section (sur le calcul polynomial et les fractions rationnelles) dans ce chapitre en vue de ses applications concernant le calcul de primitives et d'intégrales. Cette décomposition utilise les nombres complexes, rappelés au chapitre 1 du volume 3.

3.4.1. *Les fonctions polynômes*

3.4.1.1. *Définitions*

Une fonction polynôme P **de degré** n est définie sur une partie quelconque de $\mathbb{R}$ ou $\mathbb{C}$ par

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

où a_i, $i = 0, 1, ..., n$, sont réels ou complexes, et $a_n \neq 0$. On notera $d^\circ(P) = n$.

La fonction polynomiale nulle, définie par $P(x) = 0$, n'a pas de degré (certains auteurs fixent à $-\infty$ son degré).

On dit que x_0 est une racine d'ordre $m \leqslant n$ de P si

$$P(x) = (x - x_0)^m S(x),$$

où S est une fonction polynôme telle que $S(x_0) \neq 0$.

3.4.1.2. *Théorème de d'Alembert*

Le résultat fondamental sur les racines d'une fonction polynôme est le suivant (de d'Alembert) :

Théorème.
Une fonction polynôme de degré n strictement positif admet exactement n racines dans $\mathbb{C}$ (comptées avec leur multiplicité).

3.4.1.3. *Division euclidienne*

On peut effectuer **la division euclidienne** de deux fonctions polynômes $A(x)$ et $B(x)$, cette dernière étant supposée non nulle. On a le résultat :

Théorème.
Il existe un couple unique (Q, R) de fonctions polynômes telles que :
$A(x) = B(x)Q(x) + R(x)$ *avec* $d^\circ(R) < d^\circ(B)$.

3.4.1.4. *Calculs avec Matlab*

Sous *Matlab*, une fonction polynôme peut être définie sous forme d'expression symbolique, mais aussi sous forme d'un tableau numérique formé de ses coefficients. Certaines opérations (en particulier la division euclidienne) nécessitent l'utilisation de ces tableaux numériques. La conversion d'une forme à l'autre se fait à l'aide des fonctions **poly2sym** et **sym2poly**.

3.4.1.5. *Exemple*

On donne les polynômes

$$\begin{cases} A(x) = x^6 - 12x^4 + 5, \\ B(x) = 2x^4 - 11x^3 + 9x^2 + 27x - 27. \end{cases}$$

1) Calculons les racines de $B(x)$ et factorisons ce polynôme.

```
» syms x
» A=x^6-12*x^4+5;
» B=2*x^4-11*x^3+9*x^2+27*x-27;
» solve(B)
ans = [1 -3/2 3 3]
» factor(B)
ans  = (x-1)*(2*x+3)*(x-3)^2
```

Ainsi les racines de ce polynôme sont $1, -3/2, 3$, cette dernière étant racine double.

2) Effectuons la division euclidienne de A par B : pour cela, on doit :

- convertir les polynômes A et B en tableaux numériques, contenant les coefficients du polynôme dans l'ordre décroissant des degrés :

```
» A1 = sym2poly(A)
A1 = 1 0 -12 0 0 0 5
» B1=sym2poly(B)
B1 = 2 -11 9 27 -27
```

- utiliser **deconv**, pour obtenir les quotient et reste sous forme de tableaux numériques $Q1$ et $R1$:

```
» [Q1,R1]=deconv(A1,B1)
Q1 = 0.5000 2.7500 6.8750
R1 = 0 0 0 37.3750 -122.6250 -111.3750 190.6250
```

- convertir les résultats en polynômes symboliques :

```
» Q= poly2sym(Q1)
Q = 1/2*x^2+11/4*x+55/8
» R = poly2sym(R1)
R = 299/8*x^3-981/8*x^2-891/8*x+1525/8
```

Les polynômes quotient et reste sont donc :

$$\begin{aligned} Q(x) &= (1/2)x^2 + (11/4)x + 55/8 \\ R(x) &= (299/8)x^3 - (981/8)x^2 - (891/8)x + 1525/8. \end{aligned}$$

3.4.2. *Fractions rationnelles*

3.4.2.1. *Définitions*

Ici, on appelle fraction rationnelle toute fonction de la forme

$$F(x) = \frac{A(x)}{B(x)},$$

où $A(x)$ et $B(x)$ sont deux fonctions polynomiales et $B(x)$ n'est pas la fonction polynomiale nulle. On dira que cette fraction rationnelle est **irréductible** s'il n'y a pas de racine commune aux deux polynômes $A(x)$ et $B(x)$. Une racine d'ordre n du polynôme $B(x)$ est appelée un **pôle** d'ordre n de la fraction F.

Lorsque $d^\circ(A) \geqslant d^\circ(B)$, la division euclidienne permet d'écrire :

$$\begin{aligned} F(x) &= \frac{B(x)Q(x) + R(x)}{B(x)} \\ &= Q(x) + \frac{R(x)}{B(x)}, \end{aligned}$$

avec $d^\circ(R) < d^\circ(B)$. Le polynôme $Q(x)$ est appelé la **partie entière** de F.

Nous considérons par la suite des fractions rationnelles

$$F(x) = \frac{A(x)}{B(x)},$$

avec $d^\circ(A) < d^\circ(B)$.

3.4.2.2. *Décomposition en éléments simples*

On appelle **élément simple** (de première espèce) tout élément de la forme

$$\frac{\lambda}{(x - x_0)^k},$$

où λ, x_0 sont des réels ou complexes et k un entier positif. De même un élément simple (de 2ème espèce) est un élément de la forme

$$\frac{\alpha x + \beta}{(x^2 + ax + b)^k},$$

où α, β, a et b sont des rééls, k un entier positif et où $a^2 - 4b < 0$. Cette dernière condition signifie que l'équation $x^2 + ax + b = 0$ n'a pas de racine réelle.

Le résultat essentiel de la **décomposition en éléments simples** d'une fraction

$$F(x) = \frac{A(x)}{B(x)},$$

avec $d^\circ(A) < d^\circ(B)$, est le suivant :

Théorème.
• *Sur* $\mathbb{C}$, $F(x)$ *se décompose d'une manière unique en somme finie d'éléments simples de 1ère espèce.*
• *Sur* $\mathbb{R}$, $F(x)$ *se décompose d'une manière unique en somme finie d'éléments simples de 1ère et 2ème espèce.*

Cette décomposition s'obtient à partir d'une factorisation du dénominateur $B(x)$. Les exemples suivants illustrent les cas usuels.

3.4.3. *Exemples*

3.4.3.1. *Cas de pôles réels simples*

La décomposition en éléments simples de la fraction rationnelle

$$F(x) = \frac{2x}{(x-1)(x+2)},$$

est de la forme

$$F(x) = \frac{\alpha}{(x-1)} + \frac{\beta}{(x+2)}.$$

Dans ce cas simple, on peut obtenir les coefficients α et β de la manière suivante :

– en multipliant par $(x-1)$ les deux membres de l'égalité ci-dessus, on a

$$\frac{2x}{(x+2)} = \alpha + \frac{\beta(x-1)}{(x+2)},$$

d'où, en faisant tendre x vers 1, on obtient $\alpha = 2/3$.

– on fait de même pour avoir $\beta = 4/3$.

Sous *Matlab*, on obtient une décompsition en utilisant la fonction **residue** (qui s'applique à des polynômes donnés sous forme de tableaux numériques) :

```
» syms x
» AdeX=2*x
» BdeX=(x-1)*(x+2)
» A1=sym2poly(AdeX)
A1=2 0
» B1=sym2poly(BdeX)
B1= 1 1 -2
» [R1, P1]=residue(A1,B1)
R1 = 1.3333
          0.6667
P1 = -2
       1
» sym(R1)
ans= [4/3]
      [ 2/3]
```

le résultat signifie que les valeurs contenues dans R_1 ($4/3$ et $2/3$) sont les numérateurs (appelés aussi résidus) des éléments simples de première espèce correspondant aux pôles donnés dans P_1 (dans l'ordre -2 et 1). On a donc

$$\frac{2x}{(x+2)(x-1)} = \frac{4/3}{(x+2)} + \frac{2/3}{(x-1)}.$$

3.4.3.2. *Cas général*

Considérons la fraction rationnelle

$$F(x) = \frac{x^4 + 2x^3 - x + 11}{(x-2)(x-1)^3(x^2+1)^2}.$$

Elle admet le pôle simple réel 2, le pôle triple réel 1 et deux pôles doubles complexes conjugués i et $-i$.
La décomposition en éléments simples de $F(x)$ sur $\mathbb{R}$ est de la forme

$$\begin{aligned} F(x) \quad = \quad & \frac{a_1}{(x-2)} + \frac{b_1}{(x-1)} + \frac{b_2}{(x-1)^2} + \frac{b_3}{(x-1)^3} + \\ & \frac{c_1 x + d_1}{(x^2+1)} + \frac{c_2 x + d_2}{(x^2+1)^2} \end{aligned}$$

On utilise le calcul symbolique et la méthode d'identification pour obtenir ces coefficients :

– on définit F et sa forme décomposée notée F_{Decomp} :

```
» syms x
» F=(x^4+2*x^3-x+11)/((x-2)*(x-1)^3*(x^2+1)^2);
» syms a1 b1 b2 b3 c1 d1 c2 d2 real
» FDecomp=a1/(x-2)+b1/(x-1)+b2/(x-1)^2+b3/(x-1)^3+...
          (c1*x+d1)/(x^2+1)+(c2*x+d2)/(x^2+1)^2;
```

– par réduction au même dénominateur, on écrit F_{Decomp} sous forme de quotient

$$F_{Decomp} = \frac{N_{Decomp}}{D_{Decomp}}$$

en utilisant la fonction **numden** de *Matlab*

```
» [NDecomp, DDecomp]=numden(FDecomp)
NDecomp =
2*c2*x+c1*x^7-5*c1*x^6-a1+3*a1*x-7*a1*x^4-5*a1*x^2+7*a1*x^3
-3*a1*x^6+a1*x^7+5*a1*x^5-2*b1+2*b2+2*c1*x+b1*x^7+7*b1*x^5
+5*b2*x^2-3*b2*x^5-6*b2*x^3+4*b2*x^4+b2*x^6+11*b1*x^3
-10*b1*x^4+5*b1*x-8*b1*x^2+2*d2-4*b1*x^6+b3*x^5-3*b2*x
-7*c1*x^2-7*d1*x-4*b3*x^2+2*b3*x^3+b3*x-2*b3*x^4-2*b3
+11*d1*x^2-12*d1*x^3+10*d1*x^4-12*c1*x^4+11*c1*x^3
+10*c1*x^5-5*d2*x^3+9*d2*x^2+d2*x^4+9*c2*x^3+c2*x^5
-5*c2*x^4-7*d2*x-7*c2*x^2+d1*x^6-5*d1*x^5+2*d1
DDecomp =
(x-2)*(x-1)^3*(x^2+1)^2
```

en utilisant *collect*, on réduit et ordonne le numérateur N_{Decomp} pour l'identifier à celui de $F(x)$:

```
» collect(NDecomp,x)
ans =
(c1+a1+b1)*x^7
+(-5*c1-3*a1+b2-4*b1+d1)*x^6
+(7*b1+10*c1+5*a1+c2+b3-3*b2-5*d1)*x^5
+(4*b2-2*b3-10*b1+d2+10*d1-12*c1-7*a1-5*c2)*x^4
+(-5*d2+2*b3+7*a1+11*c1+9*c2-6*b2+11*b1-12*d1)*x^3
+(-5*a1+11*d1+5*b2-4*b3-7*c2-8*b1-7*c1+9*d2)*x^2
+(-3*b2+5*b1+3*a1+2*c2+2*c1+b3-7*d2-7*d1)*x
+2*d1-2*b3-a1+2*b2+2*d2-2*b1
```

– cette identification conduit à un système linéaire de huit équations à huit inconnues :

```
» [a1 b1 b2 b3 c1 d1 c2 d2] = solve('c1+a1+b1=0', ...
                              'd1-4*b1+b2-5*c1-3*a1=0', ...
                              '7*b1+c2-5*d1-3*b2+5*a1+b3+10*c1=0', ...
                              '-2*b3-7*a1-10*b1+4*b2+10*d1-12*c1-5*c2+d2=1', ...
                              '2*b3-5*d2+9*c2-6*b2+7*a1+11*c1+11*b1-12*d1=2', ...
                              '9*d2-5*a1-7*c1-7*c2+11*d1-8*b1+5*b2-4*b3=0', ...
                              '2*c2-7*d2+2*c1+3*a1+b3+5*b1-3*b2-7*d1=-1', ...
                              '-2*b1-2*b3+2*d2+2*b2+2*d1-a1=11')
```

la solution est donnée par

```
a1 = 41/25
b1 =-4
b2 =1
b3 =-13/4
c1 =59/25
d1 =21/20
c2 =-7/25
d2 =-33/20
```

La décomposition de $F(x)$ en éléments simples est donc

$$F(x) = \frac{41/25}{(x-2)} - \frac{4}{(x-1)} + \frac{1}{(x-1)^2} - \frac{13/4}{(x-1)^3} + \frac{\frac{59}{25}x - \frac{21}{20}}{(x^2+1)} - \frac{\frac{7}{25}x + \frac{33}{20}}{(x^2+1)^2}.$$

3.4.3.3. *Remarques*

Dans le cas de pôles multiples, la fonction *residue* de *Matlab* peut conduire à des erreurs importantes dans le calcul numérique des coefficients $a_1, b_1, \ldots$.

Dans le cas de pôles complexes, cette fonction *residue* donne la décomposition dans $\mathbb{C}$ et non dans $\mathbb{R}$.

La méthode d'identification présentée ci-dessus est générale, mais les calculs, s'ils sont effectués à la main, peuvent être fastidieux. Il existe des méthodes spécifiques pour obtenir les éléments simples de première ou de deuxième espèce. Nous ne les présenterons pas ici.

3.5. Intégration de fractions rationnelles

Le calcul de primitives de fonctions sous forme de fractions rationnnelles utilise la décomposition en éléments simples.

Illustrons ce calcul sur quelques exemples.

Exemple 1

Pour chercher les primitives

$$\int \frac{x}{(x+1)(x+5)}dx,$$

on doit décomposer en éléments simples la fraction

$$\frac{x}{(x+1)(x+5)}.$$

La décomposition (voir paragraphe 3.4.3.1) est de la forme :

$$\frac{A(x)}{B(x)} = \frac{x}{(x+1)(x+5)} = \frac{\alpha}{(x+1)} + \frac{\beta}{(x+5)},$$

avec

$$\alpha = \lim_{x\to -1} \frac{(x+1)A(x)}{B(x)}, \quad \beta = \lim_{x\to -5} \frac{(x+5)A(x)}{B(x)}.$$

D'où

```
» syms x
» AdeX=x ;
» FdeX=(x+1)*(x+5) ;
» alpha=limit((x+1)*AdeX/FdeX,x,-1)
alpha= -1/4
» beta=limit((x+5)*AdeX/FdeX,x,-5)
beta = 5/4
```

ainsi

$$\frac{x}{(x+1)(x+5)} = \frac{-1/4}{(x+1)} + \frac{5/4}{(x+5)},$$

et

$$\int \frac{x}{(x+1)(x+5)}dx = -\frac{1}{4}\ln|x+1| + \frac{5}{4}\ln|x+5| + cste.$$

Exemple 2

Pour chercher les primitives de la fonction

$$F(x) = \frac{x}{(x-1)^2(x+1)},$$

on décompose cette fraction en éléments simples sous la forme

$$\frac{x}{(x-1)^2(x+1)} = \frac{a_1}{x-1} + \frac{a_2}{(x-1)^2} + \frac{b_1}{x+1}.$$

On procède par identification (voir paragraphe 3.4.3.2) : on réduit au même dénominateur l'expression

$$F_{Decomp} = \frac{a_1}{x-1} + \frac{a_2}{(x-1)^2} + \frac{b_1}{x+1},$$

et on écrit son numérateur Num sous forme de polynôme réduit et ordonné

```
» syms x
» syms a1 a2 b1 real
» FDecomp=a1/(x-1)+a2/(x-1)^2+b1/(x+1);
» [Num,Den]=numden(FDecomp)
Num = a1*x^2-a1+a2*x+a2+b1*x^2-2*b1*x+b1
Den =(x-1)^2*(x+1)
» Num=collect(Num)
Num = (a1+b1)*x^2+(-2*b1+a2)*x-a1+b1+a2
```

Par identification de Num et du numérateur x de F(x), on obtient un système de trois équations aux trois inconnues a_1, a_2, b_1 :

```
» [a1 a2 b1]=solve('a1+b1=0','-2*b1+a2=1','-a1+b1+a2=0')
a1 = 1/4
a2 =1/2
b1 =-1/4
```

d'où la décomposition

$$\frac{x}{(x-1)^2(x+1)} = \frac{1/4}{x-1} + \frac{1/2}{(x-1)^2} - \frac{1/4}{x+1}.$$

Les primitives cherchées sont donc

$$\begin{aligned} & \int \frac{x}{(x-1)^2(x+1)} dx \\ = \quad & \frac{1}{4}\ln|x-1| - \frac{(1/2)}{(x-1)} - \frac{1}{4}\ln|x+1| + cste. \end{aligned}$$

On obtenait directement avec *Matlab* :

```
» A=x ; B=(x-1)^2*(x+1) ;
» int(A/B)
ans = -1/2/(x-1)+1/4*log(x-1)-1/4*log(x+1)
```

3.6. Exercices

3.6.1. *Calculs de primitives usuelles*

Donner sous Matlab les primitives des fonctions usuelles suivantes définies par :

1) $x^a, \quad a \neq -1,$

2) $e^x,$

3) $\ln(x),$

4) $\sin(x), \cos(x), \tan(x),$

5) $1/(1+x^2).$

(solution p. 116)

3.6.2. *Linéarisations d'expressions trigonométriques*

1) Utiliser l'instruction *simple* de *Matlab* (option *combine(trig)*) pour exprimer $\sin^4(x)$ en fonction de $\cos(2x)$ et $\cos(4x)$ (on dira qu'on a linéarisé $\sin^4(x)$).

2) En déduire

$$\int \sin^4(x)dx.$$

3) Utiliser la même méthode pour calculer

$$\int_0^{\pi/2} \cos^2(x)dx.$$

(solution p. 116)

3.6.3. *Changement de variable (1)*

On donne l'intégrale

$$I = \int_{-1}^{+1} \sqrt{1 - x^2}\, dx.$$

Vérifier que la fonction $\varphi : [-\pi/2, +\pi/2] \to [-1, 1]$ définie par

$$\varphi(t) = x = \sin(t),$$

est un changement de variable (voir § 3.3.2) et utiliser ce changement de variable pour calculer I.

(solution p. 117)

3.6.4. *Changement de variable (2)*

Soient

$$I = \int_0^{\pi/2} \frac{\sqrt{\sin(x)}}{\sqrt{\sin(x)} + \sqrt{\cos(x)}} dx, \quad J = \int_0^{\pi/2} \frac{\sqrt{\cos(x)}}{\sqrt{\sin(x)} + \sqrt{\cos(x)}} dx.$$

1) Que donne *Matlab* pour ces deux intégrales ?

2) Montrer grâce au changement de variable $x = \pi/2 - t$ que $I = J$.

3) Calculer $I + J$. En déduire I et J.

(solution p. 118)

3.6.5. *Décomposition en éléments simples*

1) Factoriser sur $\mathbb{R}$ le polynôme $B(x) = x^3 + 1$.

2) Donner la décomposition en éléments simples de la fraction rationnelle

$$\frac{1}{x^3 + 1}.$$

3) En déduire la valeur de l'intégrale

$$\int_0^1 \frac{dx}{x^3 + 1}.$$

4) Vérifier ce résultat par un calcul direct de cette intégrale sous *Matlab*.

(solution p. 119)

3.7. Solutions

Exercice 3.6.1

Il suffit d'utiliser la commande *int.*

On prendra garde à certaines réponses données par Matlab, qui font abstraction des domaines de définitions des fonctions considérées.

Voici certaines réponses aux primitives demandées

```
» syms x a real
» int(x^a,x)
ans=x^(a+1)/(a+1)
» int(log(x))
ans=x*log(x)-x
» int(tan(x))
ans=-log(cos(x))
```

la dernière réponse est correcte si on écrit

$$-\ln(|\cos(x)|\,.$$

Exercice 3.6.2

1) En écrivant

```
» syms x real
» simple(sin(x)^4)
```

on obtient plusieurs réponses, parmi lesquelles

```
combine(trig) :
3/8+1/8*cos(4*x)-1/2*cos(2*x)
```

d'où

$$\sin^4 x = 3/8 + (1/8)\cos(4x) - (1/2)\cos(2x).$$

2) De cette forme linéarisée de $\sin^4 x$, on déduit

$$\int (\sin x)^4\, dx = (3/8)\, x + (1/32)\sin(4x) - (1/4)\sin(2x) + Cste.$$

On peut le vérifier avec *Matlab*

```
» I=int(3/8+1/8*cos(4*x)-1/2*cos(2*x))
I =3/8*x+1/32*sin(4*x)-1/4*sin(2*x)
```

3) De même

```
» simple(cos(x)^2)
...
combine(trig) :
1/2*cos(2*x)+1/2
...
» I=int(1/2*cos(2*x)+1/2,0,pi/2)
I =1/4*pi
```

Ainsi

$$\int_0^{\pi/2} (\cos x)^2 \, dx = \left[\frac{1}{2} + \frac{1}{2}\cos(2x)\right]_0^{\pi/2} = \frac{\pi}{4}.$$

Exercice 3.6.3

La fonction $\varphi : [-\pi/2, +\pi/2] \to [-1, 1]$ définie par

$$\varphi(t) = x = \sin(t)$$

est continument dérivable sur $[-\pi/2, +\pi/2]$. De plus :

– pour $t = -\pi/2$, on a $x = \sin(t) = -1$,

– pour $t = \pi/2$, on a $x = \sin(t) = 1$.

La formule de changement de variable

$$\int_a^b f(x)dx = \int_\alpha^\beta f(\varphi(t))\varphi'(t)dt.$$

donne ici

$$\int_{-1}^{1} \sqrt{1-x^2}dx = \int_{-\pi/2}^{\pi/2} \sqrt{1-\sin^2(t)}\cos(t)\,dt = \int_{-\pi/2}^{\pi/2} \cos^2(t)dt.$$

car le signe de $\cos(t)$ est positif.

$$\int_{-1}^{1} \sqrt{1-x^2}dx = \int_{-\pi/2}^{\pi/2} \sqrt{\cos^2(t)}\cos(t)\,dt = \int_{-\pi/2}^{\pi/2} \cos^2(t)dt.$$

Avec *Matlab,* on peut vérifier le résultat obtenu

```
» syms x t rea1
» I1=int(sqrt(1-x^2),x,-1,1)
I1 = 1/2*pi
» I2=int((cos(t))^2,-pi/2,pi/2)
I2 = 1/2*pi
```

On peut aussi effectuer pas à pas les calculs du changement de variable :

```
» fDEx=sqrt(1-x^2);
» phiDEt=sin(t);
» alpha=sym('-pi/2');beta=sym('pi/2');
» phiDEalpha=simplify(subs(phiDEt,t,alpha))
phiDEalpha = -1
» phiDEbeta=simplify(subs(phiDEt,t,beta))
phiDEbeta = 1
» gDEt=simplify(subs(fDEx,x,phiDEt)*diff(phiDEt))
gDEt = signum(cos(t))*cos(t)^2
» I3=int(gDEt,t,alpha,beta)
I3 = 1/2*pi
```

Exercice 3.6.4

1) Le calcul direct des deux intégrales ne peut pas s'effectuer sous *Matlab,* faute de primitive calculable :

```
» syms x real
» FdeX=sqrt(sin(x))/(sqrt(sin(x))+sqrt(cos(x)));
» int(FdeX,0,pi/2)
Warning : Explicit integral could not be found
```

2) On doit donc utiliser le changement de variable donné. La fonction φ, définie par

$$\varphi(t) = x = \frac{\pi}{2} - t,$$

est continument dérivable sur $\mathbb{R}$. De plus :

$$\begin{cases} x = 0 \text{ pour } t = \pi/2, \\ x = \pi/2 \text{ pour } t = 0, \end{cases}$$

et

$$dx = \varphi'(t)dt = -dt.$$

La formule de changement de variable donne ici

$$\begin{aligned} I &= \int_0^{\pi/2} \frac{\sqrt{\sin(x)}}{\sqrt{\sin(x)}+\sqrt{\cos(x)}}dx \\ &= \int_{\pi/2}^{0} \frac{\sqrt{\sin(\frac{\pi}{2}-t)}}{\sqrt{\sin(\frac{\pi}{2}-t)}+\sqrt{\cos(\frac{\pi}{2}-t)}}(-1)dt \\ &= \int_0^{\pi/2} \frac{\sqrt{\cos(t)}}{\sqrt{\sin(t)}+\sqrt{\cos(t)}}dt \\ &= J. \end{aligned}$$

3) Il est par ailleurs facile de vérifier que $I + J = \pi/2$:

```
» GdeX=sqrt(cos(x))/(sqrt(sin(x))+sqrt(cos(x)));
» SdeX=simplify(FdeX+GdeX)
SdeX = 1
» IplusJ=int(SdeX,0,pi/2)
IplusJ = 1/2*pi
```

D'où $I = J = \pi/4$.

Exercice 3.6.5

1) On factorise le polynôme :

```
» syms x real
» BdeX=x^3+1;
» factor(BdeX)
ans = (x+1)*(x^2-x+1)
```

On a deux éléments simples, l'un de première espèce, l'autre de deuxième espèce, d'où la décomposition :

$$F(x) = \frac{1}{x^3+1} = \frac{a}{x+1} + \frac{bx+c}{x^2-x+1}.$$

2) On va déterminer les coefficients a, b, c (sous *Matlab*) par identification :

```
» FdeX=1/BdeX ;
» syms a b c
» GdeX=a/(x+1)+(b*x+c)/(x^2-x+1)
GdeX =a/(x+1)+(b*x+c)/(x^2-x+1)
» [N,D]=numden(GdeX)
N =a*x^2-a*x+a+b*x^2+b*x+c*x+c
D =(x+1)*(x^2-x+1)
» collect(N,x)
ans =(b+a)*x^2+(b-a+c)*x+a+c
```

ainsi

$$\frac{1}{x^3+1} = \frac{(b+a)x^2+(b-a+c)x+a+c}{x^3+1},$$

et de

```
» S=solve('b+a=0','b-a+c=0','a+c=1');
» [S.a S.b S.c]
ans =[ 1/3, -1/3, 2/3]
```

on a

$$F(x) = \frac{1}{x^3+1} = \frac{1/3}{(x+1)} + \frac{(-1/3)x+2/3}{x^2-x+1}.$$

3) La primitive de la première fraction est simple. Pour la deuxième fraction, on doit l'écrire sous forme

$$\frac{(-1/3)x+2/3}{x^2-x+1} = \frac{b_1 u'(x)}{u(x)} + \frac{a_1}{u(x)},$$

avec $u(x) = x^2 - x + 1$. D'où les calculs :

```
» syms a1 b1
» u=x^2-x+1 ;
» u1=diff(u);
» num=collect(a1+b1*u1,x)
num =2*b1*x-b1+a1
» S=solve('2*b1=-1/3','-b1+a1=2/3');
» [S.a1 S.b1]
ans =[ 1/2, -1/6]
```

Et donc

$$\frac{(-1/3)x+2/3}{x^2-x+1} = \frac{-1}{6}\frac{2x-1}{x^2-x+1} + \frac{1}{2}\frac{1}{x^2-x+1}.$$

Par ailleurs

$$\frac{-1}{6}\int \frac{2x-1}{x^2-x+1}dx = \frac{-1}{6}\ln(x^2-x+1) + Cste.$$

Le calcul de la primitive de

$$\frac{1}{x^2-x+1}$$

nécessite l'écriture du polynôme x^2-x+1 sous forme canonique :

$$x^2-x+1 = a_1\left[a_2\left(x-b\right)^2+1\right].$$

On recherche là encore les coefficients a_1, a_2, b par identification

```
» syms a1 a2 b real
» collect(a1*(a2*(x-b)^2+1),x)
ans = a1*a2*x^2-2*a1*a2*b*x+a1*(a2*b^2+1)
» [a1,a2,b]=solve('a1*a2=1','-2*a1*a2*b=-1','a1*(a2*b^2+1)=1')
a1 = 3/4
a2 = 4/3
b = 1/2
```

d'où l'on déduit que

$$\frac{1}{x^2-x+1} = \frac{4/3}{\left[2/\sqrt{3}\left(x-1/2\right)\right]^2+1},$$

et par le changement de variable

$$2/\sqrt{3}\left(x-1/2\right) = u,$$

on trouve

$$\begin{aligned}\int \frac{dx}{x^2-x+1} &= \int \frac{4/3}{\left[2/\sqrt{3}\left(x-1/2\right)\right]^2+1}dx \\ &= \int \frac{2\sqrt{3}/3}{u^2+1}du \\ &= \frac{2\sqrt{3}}{3}\mathrm{Arc\,tan}\left(2/\sqrt{3}\left(x-1/2\right)\right)+cste.\end{aligned}$$

En regroupant les différents résultats, on obtient

$$\begin{aligned}\int \frac{dx}{x^3+1} &= \int \frac{1/3}{(x+1)}dx + \int \frac{(-1/3)x+2/3}{x^2-x+1}dx \\ &= \frac{1}{3}\ln|x+1| - \frac{1}{6}\int \frac{2x-1}{x^2-x+1}dx + \frac{1}{2}\int \frac{dx}{x^2-x+1} \\ &= \frac{1}{3}\ln|x+1| - \frac{1}{6}\ln(x^2-x+1) + \\ &\quad \frac{\sqrt{3}}{3}\mathrm{Arc\,tan}\left(2/\sqrt{3}\left(x-1/2\right)\right)+cste.\end{aligned}$$

4) Le calcul direct sous *Matlab* est évidemment plus rapide et donne :

```
» int(FdeX)
ans =1/3*log(1+x)-1/6*log(x^2-x+1)
+1/3*3^(1/2)*atan(1/3*(2*x-1)*3^(1/2))
```

DEUXIÈME PARTIE

Analyse numérique élémentaire

Chapitre 4

Arithmétique de l'ordinateur

Dans ce chapitre, on va rappeler l'essentiel sur l'arithmétique des ordinateurs travaillant sur les flottants en double précision.

Il est important de rappeler que toute opération mathématique sur des nombres réels (ou complexes) effectuée sur ordinateur ne peut se faire que si les nombres en question ont un nombre fini de chiffres significatifs. Et donc dès le départ, une opération arithmétique quelconque obéira à la contrainte technique du nombre possible de chiffres utilisables : il faudra *arrondir* ou *tronquer.*

4.1. Représentation des entiers

4.1.1. *Généralités*

Soit b un entier naturel tel que $b \geqslant 2$. On sait que :

Théorème :
pour tout entier $x \in \mathbb{N}$, *il existe une unique suite finie d'entiers* $x_0, x_1, \ldots, x_p$ *tels que*

$$x = x_0 + x_1 b + x_2 b^2 + \cdots + x_p b^p = \sum_{i=0}^{p} x_i b^i,$$

où x_i *appartient à* $\{0, 1, 2, ..., b-1\}$.

L'écriture de x en base b est alors

$$x = (x_p x_{p-1} .... x_0)_b .$$

Les bases utilisées couramment sont :

$b = 2$ base binaire,
$b = 8$ base octale,
$b = 10$ base décimale,
$b = 16$ base hexadécimale.

Pour cette dernière, par convention, les x_i sont pris dans l'ensemble

$$\{0, 1, ..., 9, A, B, C, D, E, F\}$$

où A représente 11, B représente 12, etc...
On a par exemple :

$$(10110)_2 = 2^4 + 2^2 + 2^1 = (22)_{10},$$
$$(3AF)_{16} = 3 \times 16^2 + 10 \times 16^1 + 15 = (943)_{10}.$$

On obtient successivement la suite $(x_0, x_1, \ldots, x_p)$ des chiffres d'un entier x dans la base b de la manière suivante :

– le terme x_0 est le reste de la division euclidienne de x par b, car

$$x = x_0 + b(x_1 + x_2 b... + x_p b^{p-1}).$$

On note

$$q_1 = x_1 + x_2 b... + x_p b^{p-1}$$

le quotient de cette division euclidienne.

– Le terme x_1 est alors le reste de la division euclidienne de q_1 par b.

– Plus généralement les termes x_i pour $i = 1, ..., b-1$, sont les restes de la division euclidienne par b des quotients successifs.

4.1.2. *Exemples*

On dispose en pratique les calculs de la manière suivante :

– pour convertir le nombre décimal 11 en base 2 :

```
11 |2
 1  5 |2
 ↖  1  2 |2
    ↖  0  1 |2
       ↖  1  0
```

ainsi $(11)_{10} = (1011)_2$,

– de même, pour convertir le nombre décimal 92 en base 16 :

$$\begin{array}{l|l} 92 & 16 \\ \hline 12 & 5 \quad \vert 16 \\ \hline \nwarrow\ 5 & 0 \end{array}$$

ainsi $(92)_{10}$=$(5C)_{16}$.

4.1.3. *Fonctions prédéfinies de Matlab*

Les fonctions **dec2bin**, **dec2hex** et **dec2base** permettent de convertir un entier donné en une chaîne de caractères représentant cet entier dans la base choisie.
Les fonctions **bin2dec**, **hex2dec** et **base2dec** effectuent la conversion inverse.
Ainsi :

```
» dec2bin(82)
ans=1010010
» bin2dec('10111')
ans=23
»base2dec('77',8)
ans=63
```

4.2. Représentation des réels positifs en virgule fixe

4.2.1. *Notations*

L'écriture usuelle en base 10, telle que

$$(0,375)_{10} = 3 \times 10^{-1} + 7 \times 10^{-2} + 5 \times 10^{-3}.$$

s'étend pour définir l'écriture en virgule fixe dans une base b quelconque :

$$(0, x_{-1}x_{-2}....x_{-n})_b = x_{-1}b^{-1} + x_{-2}b^{-2} + ... + x_{-n}.b^{-n} = \sum_{i=1}^{n} x_{-i}b^{-i}$$

On remarquera que, étant donné une suite infinie $(x_{-1}, x_{-2}, \cdots, x_{-n}, \cdots)$ d'entiers appartenant à $[0,\ b-1]$, la suite de terme général

$$u_n = (0, x_{-1}x_{-2}....x_{-n})_b$$

est :

– croissante car

$$u_n - u_{n-1} = x_{-n}.b^{-n} > 0,$$

– et majorée par 1 car

$$\begin{aligned} u_n &= x_{-1}.b^{-1} + x_{-2}b^{-2} + ... + x_{-n}.b^{-n} \\ &\leqslant (b-1).b^{-1} + (b-1).b^{-2} + ... + (b-1).b^{-n} \\ &\leqslant (b-1)\frac{b^{-1} - b^{-n-1}}{1 - b^{-1}} \\ &\leqslant (b-1)\frac{1 - b^{-n}}{b-1} \leqslant 1. \end{aligned}$$

Elle converge donc (voir chapitre 1, paragraphe 1.4, p. 24) vers un nombre réel, noté

$$(0,\ x_{-1}x_{-2}....x_{-n}...)_b$$

ou encore

$$\sum_{i=1}^{\infty} x_{-i}b^{-i}.$$

On montre plus généralement que tout réel positif peut s'écrire

$$\begin{aligned} x &= (x_p x_{p-1}....x_0,\ x_{-1}x_{-2}....x_{-n}...)_b \\ &= \underbrace{\sum_{i=0}^{p} x_i b^i}_{y} + \underbrace{\sum_{i=1}^{\infty} x_{-i}b^{-i}}_{z} \end{aligned}$$

C'est la représentation usuelle des nombres réels positifs en base b, appelée représentation en virgule fixe. Le nombre x est la somme d'une partie entière

$$y = E(x)$$

et d'une partie dite fractionnaire (éventuellement nulle)

$$z \in [0, 1[.$$

Le calcul des x_i pour $i \geqslant 0$ se fait comme précédemment. Pour $i < 0$ on a

$$x_{-1} = E(bz),$$

et si z_1 est la partie fractionnaire de bz alors

$$x_{-2} = E(bz_1),$$

on réitère ensuite le procédé pour déterminer $x_{-3}, \cdots, x_{-n}$.

4.2.2. *Exemple en base 2*

On veut écrire $x = (1/3)_{10}$ en virgule fixe en base $b = 2$. La partie entière est $y = 0$. Pour obtenir la partie fractionnaire, on peut disposer les calculs de la façon suivante :

z_i	bz_i	$E(bz_i)$
$z_0 = 1/3$	$2/3$	$x_{-1} = 0$
$z_{-1} = 2/3$	$4/3$	$x_{-2} = 1$
$z_{-2} = 1/3$	$2/3$	$x_{-3} = 0$
...	...	...

Ainsi par périodicité on trouve :

$$x = (1/3)_{10} = (0, 0101\underline{01}...)_2$$

A l'inverse, pour convertir la partie fractionnaire de ce réel de la base 2 vers la base 10, on a

$$(0, 010101...)_2 = (1/2)^2 + (1/2)^4 + ... = \lim_{n\to\infty} \sum_{k=1}^{n} \left(\frac{1}{2}\right)^{2k}.$$

C'est la limite de la somme des termes d'une suite géométrique de premier terme $(1/2)^2$, de raison $(1/2)^2$, limite qui est égale à

$$\left(\frac{1}{2}\right)^2 \frac{1}{1-(1/2)^2} = \frac{\frac{1}{4}}{\frac{3}{4}} = \frac{1}{3}.$$

Donc on retrouve

$$(0, 010101...)_2 = (1/3)_{10}$$

4.2.3. *Exemple en base 8*

On a de même

z_i	bz_i	$E(bz_i)$
$z_0 = 1/3$	$8/3$	$x_{-1} = 2$
$z_{-1} = 2/3$	$16/3$	$x_{-2} = 5$
$z_{-2} = 1/3$	$8/3$	$x_{-3} = 2$
...	...	...

d'où

$$x = (1/3)_{10} = (0, 252525...)_8$$

4.2.4. *Calculs avec Matlab*

Les fonctions prédéfinies de conversion (*dec2bin*, *bin2dec,* etc...) ne s'appliquent qu'à des entiers. Mais on peut définir une fonction

function s=fracDec2bin(frac,n)

qui calcule en base 2 les n premiers chiffres de la partie fractionnaire $frac$ d'un réel, donnée en base 10. On obtient le résultat sous forme d'une chaîne de caractères s.

En reprenant les notations ci-dessus, on utilise avec $Matlab$ la variable Z qui contiendra à chaque itération la partie fractionnaire z_{-i} et la variable X qui contiendra la partie entière x_{-i}. Suivant la valeur de X, on ajoute le caractère $'0'$ ou $'1'$ à la chaîne s.

```
function s=fracDec2bin(frac,n)
Z=frac ;
for i=1 :n,
      v=2*Z ;
      X=floor(v) ;
      Z=v-X ;            %Z est la nouvelle partie fractionnaire
      if X==1,s(i)='1' ;else s(i)='0' ;end
end
```

Exemple d'utilisation :

```
» fracDec2bin(1/3,14)
ans =01010101010101
```

4.3. Représentation des réels en virgule flottante

4.3.1. *Généralités*

Soit b une base. On montre alors le résultat suivant :

tout réel x non nul peut s'écrire sous la forme

$$\begin{aligned} x &= \pm b^q \sum_{i=1}^{\infty} x_{-i} b^{-i} \\ &= \pm b^q . m \\ &= \pm b^q . (0,\ x_{-1} x_{-2} .... x_{-n} ...)_b, \text{ avec } x_{-1} \neq 0. \end{aligned}$$

q s'appelle l'exposant, m la mantisse.

On écrira aussi en décalant la virgule d'une position vers la droite :

$$x = \pm b^{q-1} . (\ x_{-1},\ x_{-2} .... x_{-n} ...)_b \ .$$

C'est l'écriture **scientifique** d'un réel.

Les deux écritures ci-dessus sont appelées représentation en **virgule flottante normalisée** (VFN) du nombre x.

4.3.2. *Exemple*

Le réel $x = 61,123$ en base 10 s'écrit

$$\begin{aligned}(61,123)_{10} &= 10^2.(0,61123)_{10} \\ &= 10^1.(6,1123)_{10}.\end{aligned}$$

On utilise *Matlab* pour donner les 25 premiers chiffres de l'écriture en base 2 de ce même réel : à l'aide de fonctions décrites précédemment, on obtient la partie entière, qui comprend 6 chiffres, et les 19 premiers chiffres de la partie fractionnaire en base 2 :

```
» dec2bin(61)
ans = 111101
» fracDec2bin(0.123,19)
ans = 0001111101111100111
```

En concaténant les deux expressions, on obtient

$$x = (111101,0001111101111100111...)_2$$

puis, en décalant convenablement la virgule

$$x = 2^6.\,(0,1111010001111101111100111...)_2\,,$$

ou bien, en notation scientifique

$$x = 2^5.\,(1,1110100011111011111001 11...)_2\,.$$

4.4. Les réels en V.F.N à t chiffres

Une machine (ordinateur, calculatrice,...) ne peut stocker qu'un nombre fini de chiffres pour représenter un réel donné. On utilise pour cela la représentation en virgule flottante normalisée, en arrondissant ou en tronquant la mantisse à t chiffres.

4.4.1. *En base 10*

4.4.1.1. *Troncature et arrondi*

Soit x un réel positif quelconque (en $V.F.N$)

$$x = 10^q.\,(0,\ x_{-1}x_{-2}....x_{-n}...)_{10}\,.$$

Alors sa représentation en virgule flottante normalisée à t chiffres en **troncature** est le nombre

$$x = 10^q.\,(0,\ x_{-1}x_{-2}....x_{-t})_{10}\,.$$

Sa représentation machine en virgule flottante normalisée à t chiffres en **arrondi** est le nombre

$$fl(x) = 10^q.m$$

où

$$m = \begin{cases} 0, x_{-1}x_{-2}....x_{-t} & \text{si } x_{-t-1} < 5 \\ 0, x_{-1}x_{-2}....x_{-t} + 10^{-t} & \text{sinon.} \end{cases}$$

Dans le deuxième cas, on renormalisera si nécessaire l'écriture de $fl(x)$.

L'arrondi est en fait le nombre le *plus proche* de x dont la mantisse possède t chiffres.

4.4.1.2. *Exemple*

Soit $t = 3$. Alors, en arrondi

$$\begin{array}{rcll} x & = & 1579 & fl(x) = 10^4.(0,158) \\ x & = & 1573 & fl(x) = 10^4.(0,157) \\ x & = & 1575 & fl(x) = 10^4.(0,158) \\ x & = & 1995 & fl(x) = 10^4.(0,2). \end{array}$$

4.4.1.3. *Utilisation de vpa*

Sous *Matlab*, l'instruction

$$vpa(S,t)$$

ou la séquence

$$digits(t)\ ;\ vpa(S)$$

donne la représentation de l'expression S en virgule flottante, en base 10, et en arrondi à t chiffres. S représente le plus souvent une expression symbolique pouvant être

évaluée. Mais S peut être aussi de type numérique, chaîne de caractères, ou tableau. Le résultat est de type symbolique.

```
» vpa(sym('pi'),30)
ans = 3.14159265358979323846264338328
» x= sym('50/3333');
» x1 =vpa(x,10)
x1= .1500150015e-1
» x2=vpa(x,29)
x2=.15001500150015001500150015002e-1
```

4.4.2. *En base 2*

Le principe de l'arrondi et de la troncature est le même qu'en base 10.

Soit x un réel positif quelconque (en VFN)

$$x = 2^q.\,(0,\ x_{-1}x_{-2}....x_{-n}...)_2\,.$$

Alors sa représentation machine en virgule flottante normalisée à t chiffres en **troncature** est

$$x = 2^q.\,(0,\ x_{-1}x_{-2}....x_{-t})_2\,.$$

Sa représentation machine en virgule flottante normalisée à t chiffres en **arrondi** est le nombre

$$fl(x) = 2^q.m$$

où

$$m = \begin{cases} 0, x_{-1}x_{-2}....x_{-t} & \text{si } x_{-t-1} = 0 \\ 0, x_{-1}x_{-2}....x_{-t} + 2^{-t} & \text{si } x_{-t-1} = 1. \end{cases}$$

4.4.2.1. *Exemple*

Le nombre

$$x = 2^6.\,(0, 11110100011111101111100111...)_2$$

a pour représentation en V.F.N :

– en troncature à 3 chiffres

$$\overline{x} = 2^6.\,(0,111)_2$$

– en arrondi à 4 chiffres

$$\begin{aligned} \widetilde{x} &= 2^6.\,[(0,111)_2 + (0,001)_2] \\ &= 2^6.\,(1,000)_2 \\ &= 2^7.\,(0,100)_2\,. \end{aligned}$$

4.4.3. *Les formats machine float et double*

En pratique, la plupart des ordinateurs utilise la base 2. La norme actuelle retient deux formats standards de représentation des réels :

4.4.3.1. *Le format double (flottants en double précision)*

Pour un réel x non nul donné, on considère son écriture en virgule flottante normalisée arrondie à 53 chiffres

$$fl(x) = \pm 2^q \, (a_0, a_{-1}a_{-2} \ldots a_{-t})_2 \, .$$

avec

$$a_0 \neq 0, \ t = 52.$$

– La mantisse a son premier chiffre a_0 différent de 0, donc nécessairement égal à 1 en base 2 et il n'est pas nécessaire de le représenter en machine. C'est la convention du "bit implicite". On ne mémorisera donc que la partie fractionnaire de la mantisse

$$f = a_{-1}a_{-2} \ldots a_{-52}.$$

– Pour représenter l'exposant en base 2 avec 11 chiffres, on impose

$$-2^{10} + 2 = -1022 \leqslant q \leqslant 1023 = 2^{10} - 1,$$

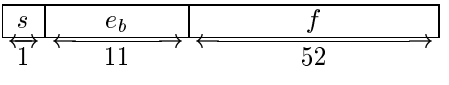
et on représentera l'exposant biaisé

$$e_b = 1023 + q$$

compris entre 1 et 2046.[1].

– Enfin le signe sera mémorisé à l'aide d'un bit s égal à 1 si le nombre x est négatif, égal à 0 si x est positif.

Avec ces notations, le nombre x sera représenté sur 64 bits répartis ainsi :



4.4.3.2. *Le format float (flottants en simple précision)*

Pour ce format, les réels sont stockés sur 32 bits, 1 pour le signe, 8 pour l'exposant et 23 pour la mantisse, suivant le même principe.

1. Les exposants biaisés 0 et 2047 sont utilisés pour représenter des nombres dénormalisés, notamment 0 et inf.

4.4.3.3. *Calculs avec Matlab*

Les calculs numériques s'effectuent en binaire, au format *double,* en arrondi même si les résultats sont ensuite affichés en décimal.

Il ne faut pas confondre cet affichage en base 10, avec l'utilisation de la fonction vpa qui, elle, effectue directement les calculs en arithmétique flottante en base 10 (voir 4.4.1.3).

Si on veut connaître avec *Matlab* le format interne de représentation d'un réel, il faut utiliser le format d'affichage **hex** (hexadécimal).

Considérons par exemple le nombre $x = (0.6)_{10}$

```
» x=0.6 ;
» format hex
» x
x =3fe3333333333333
```

en convertissant chacun des chiffres hexadécimaux en binaire, on obtient la représentation binaire interne de x :

0	011 1111 1110	0011 0011 0011 0011 ... 0011

– Le bit de signe est donc $s = 0$ (nombre positif) ;

– L'exposant biaisé s'écrit en binaire

$$e_b = 011\ 1111\ 1110,$$

donc en base 10

```
» format
» bin2dec('01111111110')
ans = 1022
```

et l'exposant non biaisé est

$$q = e_b - 1023 = 1022 - 1023 = -1.$$

Enfin, la mantisse est, après ajout du bit implicite

$$1,0011\ 0011\ 0011\ 0011\ 0011\ 0011\ 0011\ 0011\ 0011\ 0011\ 0011\ 0011\ 0011$$

On peut retrouver la valeur décimale de cette mantisse en utilisant, pour convertir la partie fractionnaire, la fonction *fracBin2dec* obtenue à l'exercice 4.6.3

```
» fracBin2dec('0011001100110011001100110011001100110011')
ans = 0.2000
```

On retrouve alors la valeur décimale du nombre x

$$x = +2^{-1} \cdot (1,2) = 0,6.$$

4.5. Opérations de base sur les nombres machine

Nous allons décrire la multiplication, l'addition et la division sur une machine travaillant en arithmétique flottante. Ces opérations élémentaires sont effectuées dans une mémoire auxiliaire (accumulateur) travaillant généralement en double précision.

Le schéma simplifié est le suivant :

– les nombres exacts x, y sont représentés (en machine) par leurs équivalents nombres machines $fl(x), fl(y)$ à t chiffres ;

– l'opération arithmétique est exécutée sur $2t$ chiffres pour les mantisses ;

– le résultat est ensuite arrondi à t chiffres.

Nous illustrons ces opérations sur des exemples.

4.5.1. *Multiplication*

4.5.1.1. *Principe*

On part de deux nombres exacts a et a' en base 10 par exemple, en vue de calculer leur produit p.

Le produit machine P

$$\begin{aligned} P &= fl(a).fl(a') \\ &= 10^q.(0,\ d_{-1}d_{-2}....d_{-t})_{10}.10^{q'}.\left(0,\ d'_{-1}d'_{-2}....d'_{-t}\right)_{10} \\ &= 10^q.m.10^{q'}.m' = 10^{q+q'}m.m' \end{aligned}$$

est effectué exactement en double précision ($2t$ chiffres). Après renormalisation éventuelle de la mantisse $m.m'$ et changement de puissance, la machine arrondit à t chiffres et affiche le résultat P_1.

4.5.1.2. *Exemple*

Soient $a = 24,321$ et $a' = 680,76$. On suppose que $t = 4$.

Alors

$$\begin{aligned} fl(a) &= 10^2.(0,2432)_{10} \\ fl(a') &= 10^3.(0,6808)_{10} \\ P = fl(a).fl(a') &= 10^5.(0,16557056)_{10} \text{ (résultat sur } 2t \text{ chiffres)} \end{aligned}$$

La machine donnera le résultat

$$P_1 = 10^5.(0,1656)_{10} = 16560.$$

On notera que la valeur exacte du produit est

$$a.a' = 16556,76396$$

En utilisant *vpa*, on peut retrouver avec *Matlab* les étapes de ce calcul en base 10 :

```
» fla=vpa(24.321,4)
fla = 24.32
»flb=vpa(680.76,4)
flb =680.8
» P = fla*flb
P = 16557.056
» P1 = vpa(P,4)
P1 =.1656e5
```

4.5.2. *Division*

Le principe est le même que pour la multiplication.. Si on note, en base 10

$$fl(a) = \pm 10^q.m, \quad fl(a') = \pm 10^{q'}.m', \quad \text{avec } m' \neq 0,$$

on a

$$0,1 \leqslant m < 1 \quad \text{et} \quad 0,1 \leqslant m' < 1$$

d'où $0,1 < m/m' < 10$

L'opération machine en base 10 $fl(a)/fl(a')$ s'effectue donc suivant deux cas :

– si $m < m'$, le quotient m/m' est calculé exactement sur $2t$ chiffres puis le résultat mantisse est arrondi à t chiffres pour obtenir

$$10^{q-q'}.fl(m/m').$$

– si $m \geqslant m'$ on écrit $m = 10^{-1}.m''$; le calcul m''/m' s'effectue exactement comme précédemment et le résultat machine est

$$10^{q-q'+1}.fl(m''/m').$$

4.5.3. *Addition*

4.5.3.1. *Principe*

L'addition machine de

$$fl(a) + fl(a')$$

s'effectue sur les mantisses après avoir "dénormalisé" l'un des nombres, pour obtenir des exposants égaux. La suite des opérations à effectuer est :

– ramener l'exposant du plus petit à celui du plus grand,

– ajouter les mantisses sur $2t$ chiffres,

– arrondir le résultat à t chiffres.

Deux cas sont à distinguer :

– Si $(q - q') > t$ alors le résultat machine est $fl(a)$.

– Si $(q - q') \leqslant t$ on ajoute $(q - q')$ zéros à gauche de la mantisse m' pour avoir la nouvelle mantisse m''. La somme $m+ m''$ est ensuite calculée exactement en double précision et le résultat est arrondi à t chiffres.

4.5.3.2. *Exemples*

On suppose que $t = 4$ et on donne

$$\begin{aligned} fl(a) &= 10^5.(0,3218) \\ fl(a') &= 10^{-1}.(0,2432), \end{aligned}$$

on a ici $(q - q') = 6 > 4$, on écrit

$$\begin{aligned} fl(a) &= 10^5.(0,3218) \\ fl(a') &= 10^5.(0,\underbrace{000000}_{q-q' \text{ zéros}}2432) \end{aligned}$$

d'où

$$fl(a) + fl(a') = 10^5.(0,32180024)$$

(mantisse sur $2t$ chiffres). Le résultat machine est donc

$$10^5.(0,3218) = fl(a).$$

Avec la commande *vpa* de *Matlab*, on a :

```
» fla = vpa(0.3218e5,4)
fla = .3218e5
» flb = vpa(0.2432e-1,4)
flb = .2432e-1
» flaPlusbDouble = vpa(fla+flb,8)
flaPlusbDouble = 32180.024
» flaPlusb = vpa(flaPlusbDouble,4)
flaPlusb = .3218e5
```

Toujours avec $t = 4$, on donne

$$\begin{aligned} fl(a) &= 10^5.(0,3218) \\ fl(a') &= 10^2.(0,2611). \end{aligned}$$

Dans ce cas, on a $(q - q') = 3 < 4$ et

$$\begin{aligned} fl(a) &= 10^5.(0,3218) \\ fl(a') &= 10^5.\underbrace{(0,0002611)}_{m''} \end{aligned}$$

d'où

$$fl(a) + fl(a') = 10^5.(0,3220611)$$

(mantisse sur $2t$ chiffres). Le résultat machine, en arrondi, est

$$10^5.(0,3221).$$

4.5.3.3. *Calculs au format double de Matlab*

Dans ce cas $t = 53$, $b = 2$ et on peut retrouver les deux cas précédents :

1) Cas $q - q' = 54 > t$

```
» x=2226.76;
» y=x*2^(-54)
y = 1.2361e-013
» z=x+y
z = 2.2268e+003
» format hex
» x
x = 40a165851eb851ec
» z
z = 40a165851eb851ec
```

Grâce au format d'affichage héxadécimal, on vérifie qu'on a obtenu dans ce cas

$$fl(x+z) = fl(x).$$

2) Cas $q - q' = 53 \leqslant t$

```
y1=x*2^(-53);
z1=x+y1
z1 = 40a165851eb851ed
» format long
» z1
z1 = 2.226760000000001e+003
» x
x = 2.226760000000000e+003
```

On a ici

$$fl(x+z) \neq fl(x).$$

4.6. Exercices

4.6.1. *Conversion d'un entier*

Convertir en base 2 le nombre $(1321)_{10}$

(solution p. 141)

4.6.2. *Schéma de Horner*

Pour convertir un nombre

$$x = (x_p x_{p-1} .... x_0)_b$$

en base 10, il suffit d'évaluer dans cette base

$$x = x_0 + x_1 b + x_2 b^2 + \cdots + x_p b^p.$$

Cette évaluation peut se faire en utilisant le schéma de Horner

$$x = ((\cdots((x_p)\, b + x_{p-1}) + \cdots)b + x_1) + x_0.$$

1) Convertir en base 10 $(101101)_2$ et $(167)_8$

2) Ecrire une fonction

$$xd = horner(Xb, b)$$

qui applique le schéma de Horner pour convertir en base 10 l'entier dont la suite des chiffres en base b est donnée dans le tableau Xb. Tester cette fonction.

(solution p. 142)

4.6.3. *Conversion d'un nombre à virgule*

Ecrire de même une fonction

$$function\, x = fracBin2dec(s)$$

qui convertit en valeur décimale x la chaîne de caractères s représentant la partie fractionnaire

$$(s_{-1}s_{-2}\cdots s_{-n})$$

d'un nombre en base 2.

(solution p. 143)

4.6.4. *Valeurs extrêmes au format double*

En se référant au paragraphe 4.4.3 p. 134, calculer :

1) Le plus grand réel représentable en machine au format *double*.

2) Le plus petit réel strictement positif représentable dans ce même format.

3) Le réel ε tel que $1+\varepsilon$ soit le plus petit réel strictement supérieur à 1 représentable en machine.

4) Comparer avec les valeurs des constantes *realmax, realmin* et *eps* de *Matlab*.

(solution p. 143)

4.7. Solutions

Exercice 4.6.1

A chaque itération, on calcule les quotients $q(i+1)$ et le reste $r(i)$ dans la division euclidienne de $q(i)$ par b. On initialise $q(1)$ à $x = 1321$, et la base b à 2.

```
» x=1321 ;b=2 ;
» q(1)=x ;
» i=1 ;
» while q(i) ~=0
   q(i+1)=fix(q(i)/b) ;
   r(i)=q(i)-b*q(i+1) ;
   i=i+1 ;
end
» r
r = 1 0 0 1 0 1 0 0 1 0 1
```

Le résultat x_b s'obtient en inversant l'ordre des restes obtenus.

```
» n=length(r) ;
» xb=r(n :-1 :1)
xb = 1 0 1 0 0 1 0 1 0 0 1
```

On peut vérifier le résultat obtenu en utilisant la fonction prédéfinie *dec2bin.*

```
» dec2bin(x)
ans = 10100101001
```

Exercice 4.6.2

1) On applique le schéma de Horner pour convertir de la base 2 et de la base 8 vers la base 10 :

```
» x10=((((1*2+0)*2+1)*2+1)*2+0)*2+1
x10 = 45
» y10=(5*8+6)*8+7
y10 = 375
```

On effectue les vérifications à l'aide des fonctions prédéfinies *bin2dec* et *base2dec*

```
» bin2dec('101101')
ans = 45
» base2dec('567',8)
ans = 375
```

2) Selon le schéma de Horner, la valeur décimale x_d s'obtient en répétant pour les n chiffres donnés dans le tableau X_b l'opération

$$x_d = x_d.b + X_b(i),$$

pour $i \in \{1, 2, \cdots, n\}$

```
function xd=horner(Xb,b)
xd=0 ;
for i=1 :1 :length(Xb)
   xd=xd*b+Xb(i) ;
end
```

On teste cette fonction en reprenant les deux exemples de la question précédente.

```
» horner([1 0 1 1 0 1],2)
ans = 45
» horner([5 6 7],8)
ans = 375
```

Exercice 4.6.3

Pour $s = (s_{-1}s_{-2}...s_{-n})_2$, on doit calculer

$$x = s_{-1} \times 2^{-1} + s_{-2} \times 2^{-2} + ...s_{-n} \times 2^{-n} = \sum_{i=1}^{n} s_{-i}2^{-i}$$

On utilisera une variable P qui contient les valeurs successives de 2^{-i}. On initialise la valeur de x à 0 et, à chaque itération, on lui ajoute la valeur 2^{-i} seulement si $s(i) = 1$

```
function x=fracBin2dec(s)
x = 0 ;
P =1/2 ;
for i = 1 :length(s)
        if(s(i)=='1')
              x = x+P ;
        end
        P = P/2 ;
end
```

Exemple d'utilisation :

```
» fracBin2dec('01010101010101010101010101')
ans = 0.3333
```

Exercice 4.6.4

1) Au format *double*, le plus grand exposant est

$$q_{MAX} = 1023,$$

et la plus grande mantisse est, en binaire

$$m_{MAX} = 1,11...1,$$

soit

$$m_{MAX} = 2 - 2^{-52}.$$

Le plus grand nombre représentable dans ce format est donc

```
» doubleMAX=(2-2^-52)*2^1023
doubleMAX =1.7977e+308
```

2) De même on a

$$q_{MIN} = -1022,$$

et

$$m_{MIN} = (1, 0...0)_2 = 1,$$

d'où

```
» doubleMIN=1*2^(-1022)
doubleMIN = 2.2251e-308
```

3) Le nombre suivant immédiatement 1 au format *double* est

$$1 + \varepsilon = (1, 0...01)_2 = 1 + 2^{-52}$$

d'où le calcul de ε

```
» epsilon=2^(-52)
epsilon = 2.2204e-016
```

4) Ces trois valeurs correspondent à trois constantes prédéfinies de *Matlab :*

```
» realmax
ans = 1.7977e+308
» realmin
ans = 2.2251e-308
» eps
ans = 2.2204e-016
```

Chapitre 5

Gestion d'erreurs

Que se passe-t-il si on exécute avec *Matlab* les instructions suivantes ?

```
» disp('On ajoute 10 fois 0.1 ')
» S=0 ;
» for n=1 :10,
   S=S+0.1 ;
end
» if S==1
   disp('et on trouve exactement 1')
else
   disp('mais on ne trouve pas exactement 1')
end
```

On obtient pour réponse :

```
On ajoute 10 fois 0.1
mais on ne trouve pas exactement 1
```

Comme on l'a vu au chapitre précédent, lorsqu'on remplace le réel $0,1$ par sa représentation machine, on commet une erreur due à la conversion en base 2 puis à l'arrondi, appelée **erreur d'affectation.**

Dans ce chapitre, après avoir rappelé des résultats généraux sur les calculs d'erreurs absolues et relatives, on présente des calculs où les erreurs d'affectation doivent être prises en considération.

5.1. Erreur absolue et erreur relative

5.1.1. *Définition*

Soit x une quantité réelle donnée supposée exacte et $\widetilde{x}$ une approximation de x. On appelle erreur absolue commise sur x la quantité

$$\Delta x = |x - \widetilde{x}|,$$

et erreur relative commise sur x le quotient

$$\delta x = \frac{|x - \widetilde{x}|}{|x|}, \quad x \neq 0.$$

On exprimera souvent l'erreur relative en pourcentage, pour la rendre plus significative.
Ces erreurs sont souvent estimées ou majorées.

5.1.2. *Erreurs d'opérations*

Dans ce paragraphe, on note $\widetilde{x}$ et $\widetilde{y}$ des valeurs approchées de deux nombres réels x et y.

5.1.2.1. *Addition et soustraction*

On a

$$|(x + y) - (\widetilde{x} + \widetilde{y})| = |(x - \widetilde{x}) + (y - \widetilde{y})| \leqslant |x - \widetilde{x}| + |(y - \widetilde{y}|.$$

L'erreur commise sur la somme des deux nombres vérifie donc

$$\Delta(x + y) \leqslant \Delta x + \Delta y.$$

De même

$$\Delta(x - y) \leqslant \Delta x + \Delta y.$$

5.1.2.2. *Multiplication et division*

En supposant x et y strictement positifs, en utilisant la fonction logarithme et le théorème des accroissements finis, on sait qu'il existe un réel c, compris entre x et $\widetilde{x}$, tel que :

$$|\ln x - \ln \widetilde{x}| = \frac{1}{c} |x - \widetilde{x}|.$$

Comme c est proche de x, on confond dans l'estimation d'erreur, c avec x, de sorte que

$$|\ln x - \ln \widetilde{x}| \simeq \frac{1}{x} |x - \widetilde{x}| = \delta x.$$

De même :

$$|\ln y - \ln \widetilde{y}| \simeq \frac{1}{y} |y - \widetilde{y}| = \delta y$$
$$|\ln xy - \ln \widetilde{x}\widetilde{y}| \simeq \frac{1}{xy} |xy - \widetilde{x}\widetilde{y}| = \delta(xy).$$

Or

$$|\ln xy - \ln \widetilde{x}\widetilde{y}| = |(\ln x - \ln \widetilde{x}) + (\ln y - \ln \widetilde{y})| \leqslant |\ln x - \ln \widetilde{x}| + |\ln y - \ln \widetilde{y}|.$$

Donc une majoration de l'erreur relative $\delta\,(xy)$ est

$$\delta(xy) \leqslant \delta x + \delta y.$$

Ce dernier résultat reste valable pour x et y non nuls, de signes quelconques (en raisonnant sur les valeurs absolues). Par un calcul analogue au précédent, on obtient, pour un quotient de nombres non nuls :

$$\delta(x/y) \leqslant \delta x + \delta y.$$

5.1.3. *Estimation d'erreur par le théorème des accroissements finis*

Soit une application

$$\begin{array}{rccc} f: & I = [a,b] \subset \mathbb{R} & \longrightarrow & \mathbb{R} \\ & x & \mapsto & f(x). \end{array}$$

On suppose connaître une valeur approchée $\widetilde{\alpha}$ de α appartenant à I. On suppose également que la fonction f est continument dérivable sur I. On veut estimer l'erreur $|f(\widetilde{\alpha}) - f(\alpha)|$ commise sur $f(\alpha)$.
Le théorème des accroissements finis permet d'affirmer qu'il existe un nombre c compris entre a et $\widetilde{\alpha}$ tel que

$$f(\widetilde{\alpha}) - f(\alpha) = (\widetilde{\alpha} - \alpha).f'(c).$$

On en déduit l'estimation

$$|f(\widetilde{\alpha}) - f(\alpha)| \leqslant |\widetilde{\alpha} - \alpha| \max_{t \in I} |f'(t)|$$

et donc l'erreur absolue commise en confondant $f(\widetilde{\alpha})$ et $f(\alpha)$ est au plus de

$$\Delta\alpha.\max_{t \in I} |f'(t)|.$$

Exemple

Estimons l'erreur E, commise en confondant $\sqrt[3]{1000003}$ et

$$100 = \sqrt[3]{1000000}.$$

Ici on a $f(x) = \sqrt[3]{x}$ et on peut prendre $I = [1\,000\,000, 1\,000\,003]$. Le résultat précédent donne

$$\begin{aligned} E &\leqslant 3.\max_{t\in I} |f'(t)| \leqslant 3.\max_{t\in I} \left|\frac{1}{3t^{2/3}}\right| \\ &\leqslant \frac{1}{(1\,000\,000)^{2/3}} \simeq 10^{-4}. \end{aligned}$$

On peut vérifier sous *Matlab* que l'erreur

$$\left|\sqrt[3]{1000003} - 100\right|,$$

observée par un calcul direct est bien inférieure à l'estimation 10^{-4} :

```
» format long
» errObservee = abs(1000003^(1/3)-100)
errObservee = 9.999989997311332e-05
```

5.2. Erreurs d'affectation

Etant donné un nombre réel x, si on note $fl(x)$ sa représentation machine à t digits dans une base donnée, on appelle erreur d'affectation le nombre

$$\Delta x = |x - fl(x)|.$$

5.2.1. *Exemple*

On majore l'erreur d'affectation commise en base 10, en VFN à $t = 5$ chiffres, et en arrondi, sur le nombre

$$x = \frac{110}{3} = 10^2\,(0, 366666\cdots)_{10}.$$

On a

$$fl(x) = 10^2\,(0, 36667)_{10},$$

et

$$\Delta x = |x - fl(x)| = 10^2 \, (0,0000033...)_{10} \leqslant 10^2 \, (0,000004)_{10} = 10^2.4.10^{-6}.$$

On obtient une majoration de l'erreur relative en remarquant que

$$\frac{1}{|x|} = \frac{1}{10^2 \, (0,3666\cdots)_{10}} \leqslant \frac{1}{10^2 \, (0,3)_{10}} = \frac{1}{10^2.3.10^{-1}}$$

d'où

$$\frac{|x - fl(x)|}{|x|} \leqslant \frac{10^2.4.10^{-6}}{10^2.3.10^{-1}} = \frac{4}{3}10^{-5}.$$

5.2.2. *Résultat général*

On a le résultat général suivant :

Théorème
Soit le réel positif $x = b^q.(0, x_{-1}x_{-2}...x_{-t}x_{-t-1}....)_b$.
Dans une machine travaillant sur les nombres en virgule flottante normalisée à t chiffres et en arrondi, on a :

- $|x - fl(x)| \leqslant \dfrac{1}{2x_{-1}} \, |x| \, 10^{-t+1} \leqslant \dfrac{1}{2} \, |x| \, 10^{-t+1}$ *(en base* $b = 10$*)*
- $|x - fl(x)| \leqslant \dfrac{1}{2} \, |x| \, 2^{-t+1}$ *(en base* $b = 2$*).*

Montrons par exemple le deuxième point. Soit

$$x = 2^q.(0, x_{-1}x_{-2}...x_{-t}x_{-t-1}....)_2$$

alors

$$fl(x) = 2^q.m$$

où

$$m = \begin{cases} (0, x_{-1}x_{-2}...x_{-t})_2 & \text{si } x_{-t-1} = 0 \\ (0, x_{-1}x_{-2}...x_{-t})_2 + 2^{-t} & \text{si } x_{-t-1} = 1 \end{cases}$$

et donc

$$\begin{aligned} |x - fl(x)| &= |x| \left| \frac{(m - (0, x_{-1}x_{-2}...x_{-t}x_{-t-1}....)_2}{(0, x_{-1}x_{-2}...x_{-t}x_{-t-1}....)_2} \right| \\ &\leqslant |x| \, \frac{1}{(0, x_{-1})_2} \, |m - (0, x_{-1}x_{-2}...x_{-t}x_{-t-1}....)_2| \, , \end{aligned}$$

or

$$\frac{1}{(0,x_{-1})_2} = \frac{1}{1/2} = 2$$

et

$$\begin{aligned} & |(m-(0,x_{-1}x_{-2}...x_{-t}x_{-t-1}....)_2| \\ \leqslant & \begin{cases} 2^{-t-1} & \text{si } x_{-t-1}=0 \\ 2^{-t}-2^{-t-1}=2^{-t-1} & \text{si } x_{-t-1}=1, \end{cases} \end{aligned}$$

le résultat s'en déduit.

Remarque

Le théorème ci-dessus permet d'obtenir, en base b quelconque, la majoration

$$\frac{|x-fl(x)|}{|x|} \leqslant \frac{1}{2}b^{-t+1} = \varepsilon_M .$$

Ce nombre ε_M s'appelle "*epsilon machine"*.

5.2.3. *Cas des formats float et double*

On déduit du paragraphe précédent que l'erreur relative due à l'approximation d'un nombre x par sa représentation binaire en machine est :

$$\frac{|x-fl(x)|}{|x|} \leqslant \frac{1}{2}2^{-53+1} = 2^{-53} \simeq 1,1.10^{-16} \text{ (au format } \textit{double}\text{)}$$
$$\frac{|x-fl(x)|}{|x|} \leqslant \frac{1}{2}2^{-24+1} = 2^{-24} \simeq 6.10^{-8} \qquad \text{(au format } \textit{float}\text{)}.$$

On retiendra en particulier pour les calculs numériques avec *Matlab*, qui se font au format *double,* que

$$\varepsilon_M = 2^{-53}.$$

On pourra vérifier que ce nombre ε_M est égal à la constante *eps* de *Matlab*.

5.2.4. *Erreurs d'affectation et opérations*

Même si on effectue une opération à partir des valeurs exactes des nombres x et x', on peut commettre une erreur d'affectation sur le résultat de l'opération. Cette erreur d'affectation est donnée par le résultat suivant.

Théorème
Les résultats machines $fl(x+x')$, $fl(x.x')$ *et* $fl(x/x')$ *vérifient les majorations :*

- $|fl(x+x') - (x+x')| \leqslant \varepsilon_M . |x+x'|$
- $|fl(x.x') - (x.x')| \leqslant \varepsilon_M . |x.x'|$
- $|fl(x/x') - (x/x')| \leqslant \varepsilon_M . |x/x'|$.

Il suffit d'appliquer le théorème énoncé au paragraphe 5.2.2.

5.3. Cumul d'erreurs d'affectation et d'opération

Généralement, dans les calculs numériques, on effectue plusieurs opérations élémentaires pour l'obtention de la valeur cherchée. Il est clair que les erreurs d'affectation commises au début sur les nombres réels exacts, entre autres, vont avoir un impact sur le calcul final. Plus précisément, pour chaque opération, il faudra tenir compte

– de l'erreur d'opération calculée à partir de l'erreur commise sur chaque opérande,

– de l'erreur d'affectation du résultat de l'opération.

5.3.1. *Cas d'une somme*

5.3.1.1. *Cas d'une somme de deux nombres*

Soient deux nombres x et y dont on donne des valeurs approchées $\widetilde{x}$ et $\widetilde{y}$, avec des erreurs absolues majorées par Δx et Δy. Leur somme $x+y$ sera approchée en machine par

$$fl(\widetilde{x}+\widetilde{y}),$$

et on aura

$$\begin{aligned} |fl(\widetilde{x}+\widetilde{y}) - (x+y)| &\leqslant |fl(\widetilde{x}+\widetilde{y}) - (\widetilde{x}+\overline{\widetilde{y}})| + |(\widetilde{x}+\widetilde{y}) - (x+y)| \\ &\leqslant \varepsilon_M . |\widetilde{x}+\widetilde{y}| + \Delta x + \Delta y. \end{aligned}$$

Résultat :
L'erreur absolue commise sur le résultat final de la somme est obtenue en ajoutant l'erreur d'opération dans le calcul de la somme et l'erreur d'affectation de cette somme.

Si les erreurs commises sur x et y proviennent elles-mêmes uniquement de l'erreur d'affectation de ces deux nombres, le résultat machine de l'addition de x et y est

$$fl\left(fl(x)+fl(y)\right),$$

noté abusivement

$$fl(x+y),$$

et on a

$$\begin{aligned} |fl(x+y)-(x+y)| &\leqslant \varepsilon_M.|fl(x)+fl(y)|+\varepsilon_M.|x|+\varepsilon_M.|y| \\ &\simeq 2\varepsilon_M.(|x|+|y|). \end{aligned}$$

5.3.1.2. *Généralisation*

A l'aide d'un raisonnement par récurrence, on généralise ce résultat à la somme de n nombres $x_1, x_2, \cdots, x_n$:

$$\begin{aligned} & |fl(x_1+x_2+\cdots+x_n)-(x_1+x_2+\cdots+x_n)| \\ \leqslant\ & n\varepsilon_M(|x_1|+|x_2|+\cdots+|x_n|). \end{aligned}$$

5.3.2. *Cas d'un produit*

5.3.2.1. *Cas d'un produit de deux nombres*

On donne deux nombres non nuls x et y dont on connaît des valeurs approchées $\widetilde{x}$ et $\widetilde{y}$, avec une erreur relative estimée à δx et δy. Leur produit $x.y$ sera approché en machine par

$$fl(\widetilde{x}.\widetilde{y}),$$

et on aura

$$\frac{|fl(\widetilde{x}.\widetilde{y})-(x.y)|}{|x.y|} \leqslant \frac{|fl(\widetilde{x}.\widetilde{y})-(\widetilde{x}.\widetilde{y})|}{|x.y|} + \frac{|(\widetilde{x}.\widetilde{y})-(x.y)|}{|x.y|}.$$

Utilisant le fait que

$$\frac{|fl(\widetilde{x}.\widetilde{y})-(\widetilde{x}.\overline{y})|}{|x.y|} \simeq \frac{|fl(\widetilde{x}.\widetilde{y})-(\widetilde{x}.\overline{y})|}{|(\widetilde{x}.\widetilde{y}|} \leqslant \varepsilon_M,$$

on en déduit

$$\frac{|fl(\widetilde{x}.\widetilde{y})-(x.y)|}{|x.y|} \leqslant \varepsilon_M + \delta x + \delta y.$$

Résultat :
L'erreur relative commise sur le résultat final du produit est estimée en ajoutant l'erreur relative d'opération dans le calcul du produit, et l'erreur relative d'affectation de ce produit.

Là encore, si les erreurs commises sur x et y proviennent elles-mêmes uniquement de l'erreur d'affectation de ces deux nombres, on note

$$fl(x.y)$$

pour

$$fl\left(fl(x).fl(y)\right),$$

et on a

$$\frac{|fl(x.y)-(x.y)|}{|x.y|} \leqslant \varepsilon_M + \varepsilon_M + \varepsilon_M \leqslant 3\varepsilon_M.$$

5.3.2.2. *Exemple*

Dans une arithmétique en base 10, en VFN et en arrondi à $t = 3$ chiffres, on calcule le produit $p = a.b$, avec

$$a = b = 1,075$$

```
» a=1.075 ;b=1.075 ;t=3 ;
» digits(t) ;
» flA=vpa(a) ;
» flB=vpa(b) ; ;
» flP=vpa(flA*flB)
flP = 1.17
```

On calcule ensuite les erreurs relatives commises sur a, b et p, en considérant comme exactes les valeurs de ces nombres au format *double*.

```
» errA=abs(a-double(flA))/a
errA = 0.0047
» errB=abs(b-double(flB))/b
errB = 0.0047
» p=a*b
p = 1.1556
» errP=abs(p-double(flP))/p
errP = 0.0124
```

On notera que les valeurs calculées avec *vpa* sont de type symbolique et qu'il faut les convertir dans le type *double* pour effectuer numériquement les calculs d'erreurs.

On vérifie les majorations

$$\frac{|fl(a.b)-(a.b)|}{|a.b|} \leqslant \varepsilon_M. + \delta_a + \delta_b \leqslant 3\varepsilon_M.$$

```
» epsM=1/2*10^(-t+1)
epsM = 0.0050
» errA+errB+epsM
ans = 0.0143
» 3*epsM
ans = 0.0150
```

On remarque bien que, dans cette arithmétique, l'erreur relative commise dans le calcul de ab

$$errP = 0,0124$$

est strictement supérieure à la somme des erreurs relatives sur a et sur b

$$errA + errB = 0,094.$$

5.3.2.3. *Généralisation*

A l'aide d'un raisonnement par récurrence, on montre que, étant donnés n nombres non nuls $x_1, x_2, \cdots, x_n$:

$$\frac{|fl(x_1 x_2 \cdots x_n) - (x_1 x_2 \cdots x_n)|}{|x_1 x_2 \cdots x_n|} \leqslant (2n+1)\,\varepsilon_M.$$

5.4. Erreurs d'absorption

Soient deux nombres donnés par leur représentation machine en base b, en arrondi à t chiffres :

$$\begin{cases} x = \pm b^q.(0, x_{-1}x_{-2}...x_{-t})_b \\ y = \pm b^{q'}.(0, y_{-1}y_{-2}...y_{-t})_b. \end{cases}$$

On a vu au chapitre précédent que, si

$$q - q' > t,$$

on a

$$fl(x+y) = x.$$

On dit qu'il y a **absorption** de y par x. On peut vérifier aussi que cette erreur se produit lorsque

$$\left|\frac{y}{x}\right| \leqslant \varepsilon_M.$$

Plus généralement, une erreur d'absorption se produit lorsqu'on ajoute deux nombres d'ordres de grandeur très différents. Dans ce cas

$$fl((x+y) - x) \neq y.$$

Une conséquence importante de l'erreur d'absorption est la non associativité de l'addition machine, comme l'illustrent les exemples ci-dessous.

5.4.1. *Exemples*

5.4.1.1. *Une somme de trois nombres*

Effectuons les deux additions machines suivantes dans une arithmétique d'arrondi à $t = 3$ digits, travaillant en double précision et en base 10

$$\begin{aligned} S_1 &= 1 + (0,004 + 0,003) \\ S_2 &= (1 + 0,004) + 0,003. \end{aligned}$$

La première addition s'effectue comme suit

$$\begin{aligned} S_1 &= [fl(1) + fl(0,004)] + fl(0,003) \\ &= [10^1.0,1 + 10^1.0,0004] + 10^1.0,0003 \\ &= 10^1.0,1 + 10^1.0,0003 \\ &= 10^1.0,1 = 1. \end{aligned}$$

On peut aussi effectuer ce calcul avec *Matlab*, en utilisant la fonction *vpa* :

```
» digits(3);
» x = vpa(1); y = vpa(0.004);z=vpa(0.003);
» S1= vpa(vpa(x+y)+z)
S1 = 1.00
```

La deuxième addition donne :

$$\begin{aligned} S_2 &= fl(1) + [fl(0,004) + fl(0,003)] \\ &= 10^1.0,1 + \left(10^{-2}.0,4 + 10^{-2}.0,3\right) \\ &= 10^1.0,1 + 10^{-2}.0,7 \\ &= 10^1.0,1 + 10^1.0,001 \\ &= 10^1.0,101 \\ &= 1,01 \end{aligned}$$

```
» S2=vpa(x+vpa(y+z))
S2 = 1.01
```

5.4.1.2. *Une somme de plusieurs nombres*

Dans cette même arithmétique, si on calcule

$$S = 1 + \underbrace{0.004 + 0.004 + \cdots + 0.004}_{100 \text{ fois}}$$

dans l'ordre indiqué, on a :

```
» S=1 ;
» for i = 1 :100,
      S=vpa(S+0.004, 3) ;
end
» S
S = 1.00
```

Si on ajoute d'abord entre eux les nombres 0.004, puis le nombre 1 à la fin, on obtient :

```
» S=0 ;
» for i = 1 :100,
      S=vpa(S+0.004, 3) ;
end
» S =vpa(1+S,3)
S = 1.40
```

5.4.2. *Conséquence pratique*

On retiendra donc que dans une somme de nombres positifs, il faut d'abord commencer par additionner les petits nombres pour éviter l'erreur d'absorption.

5.5. Erreurs de cancellation

Une erreur de cancellation se produit généralement lorsqu'on soustrait deux quantités de même signe très proches.

5.5.1. *Présentation sur un exemple*

Considérons deux nombres exacts x et y approchés par

$$\begin{aligned} \widetilde{x} &= 100000,2 \\ \widetilde{y} &= 100000,0. \end{aligned}$$

Supposons que les erreurs absolues commises sur ces nombres sont au plus de

$$\begin{aligned} \Delta x &= 0,05 \\ \Delta y &= 0,05. \end{aligned}$$

Alors, les erreurs relatives sont

$$\begin{aligned} \delta x &\simeq 0,05/\widetilde{x} = 5.10^{-6} \\ \delta y &\simeq 0,05/\widetilde{y} = 5.10^{-6}. \end{aligned}$$

L'erreur relative commise sur la différence approchée $\widetilde{x} - \widetilde{y} = 0,2$ est

$$\begin{aligned} \delta(x-y) &= \frac{\Delta(x-y)}{|x-y|} = \frac{|(x-y)-(\widetilde{x}-\widetilde{y})|}{|\widetilde{x}-\widetilde{y}|} \\ &\leqslant \frac{0,05+0,05}{0,2} = 5.10^{-1}. \end{aligned}$$

Ce résultat signifie que l'estimation d'erreur relative commise sur les nombres x et y a été multipliée par 10^5 en calculant leur différence. Cela s'explique par le fait que les deux nombres considérés sont très proches.

5.5.2. *Exemple traité avec Matlab*

Dans une arithmétique en base 10, en arrondi, à 4 chiffres, on calcule la différence

$$d = \frac{22}{7} - \pi.$$

```
» digits(4)
» r=22/7 ;
» piApp=vpa(pi)
piApp = 3.142
» rApp=vpa(r)
rApp = 3.143
» diffApp=vpa(rApp-piApp)
diffApp = .1e-2
```

On a obtenu pour valeur approchée de d le nombre 0,01.
En considérant comme exactes les valeurs numériques au format *double* de π, $22/7$ et d, on calcule les erreurs relatives commises sur ces nombres.

```
» errPi=abs(pi-double(piApp))/pi
errPi = 1.2966e-004
» errR=abs(r- double(rApp))/r
errR = 4.5455e-005
» diff=r-pi ;
» errDiff=abs(diff-double(diffApp))/diff
errDiff = 0.2092
```

L'erreur relative sur d est de l'ordre de 21%, soit environ 1600 fois l'erreur relative commise au départ sur π.

```
» rapp=errDiff/errPi
rapp = 1.6132e+003
```

5.5.3. *Remarque*

Dans le calcul numérique d'une somme finie de termes, certains algorithmes produisent des erreurs de cancellation et d'absorption. Considérons par exemple le calcul de la somme

$$S(x) = 1 + x + \frac{x^2}{2!} + ... + \frac{x^n}{n!},$$

qui représente le développement de Taylor à l'ordre n en 0 de $\exp(x)$. Nous effectuerons successivement le calcul pour $x = -20$ (les termes sont alors de signes alternés), puis pour $x = 20$.

L'erreur commise en remplaçant $f(x) = \exp(x)$ par $S(x)$ est

$$err = \frac{x^{n+1}}{(n+1)!} f^{(n+1)}(c) = \frac{x^{n+1}}{(n+1)!} e^c,$$

où c est un réel compris entre 0 et x (cf. chapitre 2, paragraphe 2.7.1.1, p. 69). Si on suppose

$$x \in [-20,\ 20],$$

cette erreur est majorée par

$$M = \frac{20^{n+1}}{(n+1)!} e^{20}.$$

Pour $n = 80$, on obtient

```
» N= 80 ;
» X0 =20 ;
» M=X0^(N+1)/prod(1 :1 :N+1)*exp(X0)
M = 2.0235e-007
```

$$M \simeq 2.10^{-7}$$

Pour $X = -20$, on effectue le calcul de $S(X)$ en base 10, en virgule flottante normalisée, en arrondi à $t = 10$ chiffres, en utilisant les fonctions *digits* et *vpa*. On utilise trois tableaux $Puiss$, $Fact$ et T, permettant de mémoriser les valeurs de

$$\begin{cases} Puiss(i) = X^i = X.Puiss(i-1), \\ Fact(i) = i! = i.Fact(i-1), \\ T(i) = X^i/i! = Puiss(i)/Fact(i). \end{cases}$$

```
» digits(10)
» X = -20;
» Puiss(1)=sym(X);Fact(1)=sym(1);
» T(1)=vpa(X);
» S=vpa(1+T(1));
» for i=2 :N,
   Puiss(i)=vpa(Puiss(i-1)*X);
   Fact(i)=vpa(Fact(i-1)*i);
   T(i)=vpa(Puiss(i)/Fact(i));
   S=vpa(S+T(i));
end
» S
S = -.8654019579e-2
```

On obtient pour valeur approchée de $S = S(-20)$ la valeur

$$\widetilde{S} = -0,01339394792.$$

On compare cette valeur à celle de $E = e^{-20}$ donnée avec toute la précision du format *double* de *Matlab*.

```
» E =exp(X)
E = 2.0612e-009
» errObs=double(abs(E-S))
errObs = 0.0087
» Q=errObs/M
Q = 4.2767e+04
```

L'erreur absolue constatée est d'environ 42000 fois l'erreur théorique M. Elle provient d'erreurs dans le calcul de S : les termes $T(i)$ que l'on ajoute sont de signes alternés, ce qui peut provoquer dans le calcul de la somme des phénomènes de cancellation, ajoutés au risque d'absorption des plus petits en valeur absolue par d'autres beaucoup

plus grands. Affichons quelques-uns de ces termes :

```
» T(1 :5)
ans = [ -20., 200., -1333.333333, 6666.666667, -26666.66667]
» T(16 :20)
ans = [ 31322782.64, -36850332.52, 40944813.91, -43099804.12, 43099804.12]
» T(76 :80)
ans = [ .4007323037e-12, -.1040863127e-12, .2668879812e-13,
      -.6756657752e-14, .1689164438e-14]
```

Effectuons maintenant la somme $S_1 = S(X_1)$ pour $X_1 = 20$ dans la même arithmétique :

```
» X1 = 20 ;
» Puiss(1)=sym(X1) ;Fact(1)=sym(1) ;
» T1(1)=vpa(X1) ;
» S1=vpa(1+T1(1)) ;
» for i=2 :N,
   Puiss(i)=vpa(Puiss(i-1)*X1) ;
   Fact(i)=vpa(Fact(i-1)*i) ;
   T1(i)=vpa(Puiss(i)/Fact(i)) ;
   S1=vpa(S1+T1(i)) ;
end
» S1
S1 =485165195.3
» E1 =exp(X1)
E1 = 4.8517e+008
» errObs=double(abs(E1-S1))
errObs = 0.1000
```

Dans ce cas, tous les termes

$$T(i) = \frac{(X_1)^i}{(i)!}$$

sont de signe positif, et il n'y a plus d'erreurs de cancellation dans le calcul de S_1. L'impact des erreurs d'absorption est ici très minime.

Si on veut utiliser la formule de Taylor pour calculer une valeur approchée de $\exp(-20)$, il est préférable, dans ce cas précis, de calculer d'abord une valeur approchée S_1 de $\exp(20)$, car le calcul est plus fiable, puis d'utiliser la propriété

$$\exp(-20) = \frac{1}{\exp(20)}.$$

```
» E3=1/S1
E3 = .2061153623e-8
» errObs=double(abs(E-E3))
errObs = 1.0000e-18
```

5.6. Erreurs dues aux choix des formules algébriques

5.6.1. *Exemple 1*

Des erreurs de cancellation se produisent aussi dans les expressions telles que

$$\frac{1}{P(n)} - \frac{1}{Q(n)}$$

où $P(n)$ et $Q(n)$ sont des polynômes de même degré. Par exemple, pour n très grand l'expression

$$d = \frac{1}{n} - \frac{1}{n+1}$$

produira une erreur de cancellation. Pour éviter cela on utilisera l'expression équivalente

$$d = \frac{1}{n(n+1)}.$$

On peut le vérifier, en effectuant les calculs dans une arithmétique flottante à trois chiffres en base 10, avec $n = 1100$.

```
» digits(3)
» n=1100;
» v1=vpa(1/n);
» v2=vpa(1/(n+1));
» d1=vpa(v1-v2)
d1 = .1e-5
» d2=vpa(v1*v2)
d2 = .825e-6
```

En calculant numériquement la valeur de d avec toute la précision de la machine, on constate que d_2 est une bien meilleure approximation que d_1.

```
» res=1/(n*(n+1))
res = 8.2570e-007
» res1=1/n-1/(n+1)
res1 = 8.2570e-007
```

De même dans la résolution de certaines équations algébriques, on prendra garde d'utiliser, dans certains cas, les formules adéquates pour éviter la cancellation. Ainsi, pour une équation de second degré

$$ax^2 + bx + c = 0,$$

l'utilisation des formules

$$x = \frac{-b \pm \sqrt{\Delta}}{2a}$$

pourra produire une erreur de cancellation pour le calcul d'une des deux racines lorsque $|b|$ est très proche de $\sqrt{\Delta}$.

5.6.2. *Exemple 2*

Soit à résoudre l'équation

$$x^2 - 160x + 1 = ax^2 + bx + c = 0$$

en effectuant les calculs dans une arithmétique flottante normalisée à $t = 5$ digits. Si on utilise les formules classiques donnant les solutions d'une équation du second degré, on obtiendra :

```
» digits(5) ;a=vpa(1) ;b=vpa(-160) ;c=vpa(1) ;
» delta=vpa(b^2-4*a*c)
delta = 25596.
» racDelta=vpa(sqrt(delta))
racDelta = 159.99
» x1=vpa(vpa(-b+racDelta)/vpa(2*a))
x1 = 160.00
» x2=vpa(vpa(-b-racDelta)/vpa(2*a))
x2 = .50000e-2
```

Donc on a :

$$\begin{array}{ll}
\Delta & = 25596, \\
fl(\sqrt{\Delta}) & = 10^3.(0,15999)_{10}, \\
-fl(b) + fl(\sqrt{\Delta}) & = 10^3.(0,15999)_{10}, \\
fl(x_1) & = 160, \\
-fl(b) - fl(\sqrt{\Delta}) & = 10^3.(0,16000)_{10} - 10^3.(0,15999)_{10} = 10^{-2}, \\
fl(x_2) & = 10^{-2}\,(0.5)_{10}\,.
\end{array}$$

Dans le calcul de la plus petite racine x_2, une erreur de cancellation a été commise. En conservant la valeur la plus fiable x_1 et en utilisant la formule $x_2 x_1 = c/a$, on obtient pour x_2

```
» x2=vpa(c/a/x1)
x2 = .62500e-2
```

d'où

$$fl(x_2) = 10^{-2}(0,6250)_{10}.$$

La résolution de cette équation avec Matlab donne, en affichant les valeurs numériques des solutions :

```
» format short e
» double(solve('x^2-160*x+1=0'))
ans =
1.5999e+02
6.2502e-03
```

5.7. Erreurs dues aux perturbations des données

Dans certains calculs, de petites perturbations des données peuvent modifier les résultats de manière surprenante. D'où la précaution qu'on doit prendre lors de pareilles situations. Illustrons cela par deux exemples classiques.

5.7.1. *Un système d'équations linéaires*

On considère le système linéaire

$$(S_1) \begin{cases} x - 1,99995y = 3 \\ 2x - 4y = 1. \end{cases}$$

Avec le calcul symbolique de *Matlab*, on obtient sa solution :

```
» S1=solve('x-1.99995*y=3','2*x-4*y=1');
» [S1.x,S1.y]
ans = [ .10000e6, 50000.]
```

soit

$$\begin{cases} x = 100000 \\ y = 50000. \end{cases}$$

On considère le système S_2 obtenu en tronquant le premier coefficient de y à cinq chiffres significatifs :

$$(S_2) \begin{cases} x - 1,9999y = 3 \\ 2x - 4y = 1. \end{cases}$$

L'erreur relative commise sur ce coefficient est

$$\frac{|1,99995 - 1,9999|}{1,99995} \simeq 2,5 \times 10^{-5}.$$

La solution de S_2 est donnée par

```
» S2=solve('x-1.9999*y=3','2*x-4*y=1');
» [S2.x,S2.y]
ans = [ 50001., 25000.]
```

soit

$$\begin{cases} x = 50001 \\ y = 25000. \end{cases}$$

On observe une erreur relative d'environ 50% sur chacune des solutions x, y.

Dans certains sytèmes linéaires, une perturbation très petite d'un coefficient peut entraîner des résultats inattendus. On dit que ces systèmes sont mal conditionnés.

5.7.2. *Un calcul de déterminant*

On considère la matrice $(10, 10)$:

$$M = (M_{ij})_{\substack{i=1,\ldots,10 \\ j=1,\ldots,10}} = \begin{pmatrix} 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1,0001 \\ 9 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \\ 0 & 8 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \\ 0 & 0 & 7 & 7 & 6 & 5 & 4 & 3 & 2 & 1 \\ 0 & 0 & 0 & 6 & 6 & 5 & 4 & 3 & 2 & 1 \\ 0 & 0 & 0 & 0 & 5 & 5 & 4 & 3 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 4 & 4 & 3 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 3 & 3 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2 & 2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{pmatrix}$$

On veut calculer $\det(M)$, et le comparer au déterminant obtenu en tronquant $M_{1,10}$ à 1.

Pour définir la matrice M sans entrer un à un ses 100 termes, on utilise la fonction **diag** pour laquelle l'aide de *Matlab* précise :

```
» help diag
DIAG Diagonal matrices and diagonals of a matrix.
DIAG(V,K) when V is a vector with N components is a square matrix
of order N+ABS(K) with the elements of V on the K-th diagonal. K = 0
is the main diagonal, K > 0 is above the main diagonal and K < 0
is below the main diagonal.
```

En utilisant $V = (10, 9, \cdots, 1)$, et cette fonction *diag*, on définit successivement les termes sur chaque diagonale.

```
» V=[10 :-1 :1]
V = 10 9 8 7 6 5 4 3 2 1
» M=diag(V(2 :10),-1) ;
» for i =0 :9,M=M+diag(V(i+1 :10),i) ; end
» M =sym(M) ;
» M(1,10)=1.0001
M =
[ 10, 9, 8, 7, 6, 5, 4, 3, 2, 10001/10000]
[  9, 9, 8, 7, 6, 5, 4, 3, 2, 1]
[  0, 8, 8, 7, 6, 5, 4, 3, 2, 1]
[  0, 0, 7, 7, 6, 5, 4, 3, 2, 1]
[  0, 0, 0, 6, 6, 5, 4, 3, 2, 1]
[  0, 0, 0, 0, 5, 5, 4, 3, 2, 1]
[  0, 0, 0, 0, 0, 4, 4, 3, 2, 1]
[  0, 0, 0, 0, 0, 0, 3, 3, 2, 1]
[  0, 0, 0, 0, 0, 0, 0, 2, 2, 1]
[  0, 0, 0, 0, 0, 0, 0, 0, 1, 1]
» d=det(M)
d =-4411/125
```

(On a utilisé le calcul symbolique pour obtenir la valeur exacte du déterminant, évitant ainsi d'introduire des erreurs d'arrondi dues aux conversions en base 2).

Ainsi

$$\det(M) = -\frac{4411}{125} = -35,288.$$

On définit la matrice M' telle que

$$M'_{1,10} = 1$$

et, pour tout $(i, j) \neq (1, 10)$,

$$M'_{i,j} = M_{i,j}.$$

On calcule le déterminant d' de la matrice ainsi perturbée.

```
» Mprime=M ;
» Mprime(1,10)=1 ;
» dPrime=det(Mprime)
dPrime = 1
```

Alors que sur les données on a

$$\frac{\left|M'_{1,10} - M_{1,10}\right|}{M_{1,10}} = 0,01\%,$$

sur le résultat, l'erreur relative est

$$\frac{|\det(M') - \det(M)|}{\det(M)} \simeq 102,8\%.$$

On voit que, dans ces deux cas, une perturbation très petite des données a entraîné une importante variation des résultats.

5.8. Estimation probabiliste de l'erreur

Il y a aussi une vision statistique et probabiliste de la gestion des erreurs. Illustrons cette notion d'erreurs probables sur une expression simple écrite sous forme d'une somme finie de termes réels.

Soit à calculer $s = a_0 + a_1 + ... + a_n$. On suppose que l'erreur absolue commise sur a_i est Δa_i et on note

$$e_{\max} = \max_{i \in \{1,\ldots,n\}} \Delta a_i,$$

la plus grande valeur de ces erreurs.

Une estimation de l'erreur d'opération commise sur s est alors de

$$\Delta s = \Delta a_0 + \Delta a_1 + ... + \Delta a_n \leqslant n.e_{\max}.$$

Lorsque n est très grand, cette dernière estimation d'erreur est rarement atteinte, compte tenu de la compensation de signes. Aussi on peut chercher l'erreur maximale *probable* e_{prob} sur s.

On montre en calcul de probabilités que

$$e_{prob} = \sqrt{n}.e_{\max},$$

au sens où

$$\frac{e_{prob}}{\sqrt{n}.e_{\max}}$$

a de grandes chances de rester borné quand $n \to \infty$.

5.9. Exercices

5.9.1. *Erreur d'opérations*

1) Estimer l'erreur absolue commise lorsqu'on effectue la somme

$$S = \underbrace{0,1 + 0,1 + \cdots 0,1}_{1000 \text{ termes}}.$$

2) Vérifier en effectuant le calcul de cette somme avec *Matlab*, et en comparant avec le résultat exact.

(solution p. 170)

5.9.2. *Erreurs d'absorption et de cancellation*

On cherche deux nombres x_1 et x_2 dont la somme S vaut 50 millions, et le produit P vaut 1. On note x_1 le plus grand de ces deux nombres.

1) Montrer que x_1 est solution d'une équation du second degré.

2) Trouver la valeur de x_1 en résolvant cette équation.

3) Trouver la valeur approchée x_{2a} de x_2 obtenue en utilisant

$$x_1 + x_2 = S.$$

4) Trouver la valeur approchée x_{2b} de x_2 obtenue en utilisant

$$x_1 \cdot x_2 = P.$$

5) Quel est le résultat le plus fiable, et pourquoi ? Analyser les différentes causes d'erreurs et de propagation d'erreurs dans les calculs.

6) En supposant que c'est x_{2b} la bonne valeur, déterminer l'erreur relative commise en calculant x_{2a} à la place de x_{2b} et comparer avec l'erreur relative de départ commise par le stockage en machine du discriminant $delta$.

(solution p. 171)

5.9.3. *Non associativité de l'addition machine*

On considère les trois nombres $a = 1, \quad b = 2^{53}, \quad c = -2^{53}$.

1) Donner les résultats des calculs pour :

a) $s_1 = (a + b) + c$,

b) $s_2 = a + (b + c)$.

2) Expliquer les résultats obtenus.

(solution p. 172)

5.9.4. *Choix de formules de calcul*

1) Utiliser le calcul symbolique de *Matlab* pour vérifier que

$$1000 - \sqrt{999999} = \frac{1}{1000 + \sqrt{999999}}.$$

2) Calculer la valeur de ces deux expressions dans une arithmétique en virgule flottante à $t = 6$ chiffres en arrondi (utiliser *vpa* et effectuer les calculs un par un).

3) Quel est le résultat le plus fiable et pourquoi ?

(solution p. 172)

5.9.5. *Choix d'itérations de calculs*

On définit la suite $(I_n)_{n \geqslant 0}$ par

$$I_n = \int_1^e [\ln x]^n \, dx.$$

1) En utilisant les propriétés de l'intégrale, montrer que cette suite est positive et décroissante. On notera l sa limite. inconnue.

2) Montrer que cette suite vérifie

$$\begin{cases} I_0 = e - 1 \\ I_n = e - nI_{n-1} \quad , \quad \text{pour } n > 0 \end{cases}$$

(on utilisera une intégration par parties). En déduire l.

3) Utiliser les égalités ci-dessus pour calculer avec *Matlab* la valeur numérique approchée de $I_1, I_2, ..., I_{10}$ puis I_{30}.

4) Expliquer par un calcul d'erreur les résultats obtenus.

5) Utiliser la formule inverse donnant I_{n-1} en fonction de I_n, en partant d'une valeur I_{50} de votre choix, pour calculer de nouvelles valeurs approchées de I_{30}, puis $I_{10}, I_9, \cdots, I_0$.

6) Comparer les résultats obtenus.

(solution p. 173)

5.9.6. *Sujet d'étude*

La suite définie dans ce sujet d'étude est due à Jean-Michel Muller de l'Ecole Normale Supérieure de Lyon.

On considère la suite $(a_n)_{n\geqslant 1}$ définie par

$$\left\{\begin{array}{ll} a_1 & = 11/2 \\ a_2 & = 61/11 \\ a_{n+1} & = 111 - \dfrac{1130}{a_n} + \dfrac{3000}{a_n a_{n-1}}, \quad n \geqslant 2. \end{array}\right.$$

[On admet qu'aucun terme de la suite ne s'annule, et que cette suite est donc bien définie]

1) Calculer numériquement les 30 premiers termes de cette suite.

2) Utiliser le calcul symbolique pour obtenir la valeur exacte de ces 30 premiers termes, puis la valeur numérique de ces nombres. Que constatez-vous ? Quel calcul vous semble plus fiable, et pourquoi ?

3) Etude d'une suite auxiliaire.
On introduit la suite $(x_n)_{n\geqslant 0}$ définie par

$$\left\{\begin{array}{ll} x_0 & = 1 \\ x_{n+1} & = a_{n+1}.x_n\ , \quad n \geqslant 0. \end{array}\right.$$

En utilisant la relation

$$a_{n+1} = \frac{x_{n+1}}{x_n},$$

et en exprimant de même a_n et a_{n-1}, montrer que la suite (x_n) vérifie la relation linéaire, pour tout $n \geqslant 2$,

$$x_{n+1} = 111x_n - 1130x_{n-1} + 3000x_{n-2}. \qquad \text{(R)}$$

4) Une étude classique des suites de ce type consiste alors à chercher les nombres $r \neq 0$ tels que la suite définie pour $n \geqslant 0$ par

$$x_n = r^n$$

vérifie la relation (R). Montrer que r est solution d'une équation du troisième degré. Résoudre cette équation avec *Matlab*. On notera r_1, r_2, r_3 ses solutions, rangées dans l'ordre décroissant.

5) Montrer que toute suite de la forme

$$x_n = k_1.\,(r_1)^n + k_2.\,(r_2)^n + k_3.\,(r_3)^n$$

vérifie la relation (R).

6) Déterminer les valeurs des inconnues k_1, k_2, k_3 pour que les conditions initiales

$$\begin{cases} x_0 &= 1 \\ x_1 &= a_1 \\ x_2 &= a_2 a_1 \end{cases}$$

soient vérifiées.
On notera K_1, K_2, K_3 les solutions exactes, et $K_{1App}, K_{2App}, K_{3App}$ les solutions approchées obtenues par le calcul numérique de *Matlab*.

7) On définit, pour tout $n \geqslant 1$,

$$b_n = \frac{x_n}{x_{n-1}} = \frac{K_1.\,(r_1)^n + K_2.\,(r_2)^n + K_3.\,(r_3)^n}{K_1.\,(r_1)^{n-1} + K_2.\,(r_2)^{n-1} + K_3.\,(r_3)^{n-1}},$$

Montrer par récurrence que, pour tout $n \geqslant 1$, $a_n = b_n$.

8) En déduire la limite de la suite (a_n).

9) Que devient cette limite si on remplace K_1, K_2, K_3 par $K_{1App}, K_{2App}, K_{3App}$?

(solution p. 177)

5.10. Solutions

Exercice 5.9.1

1) Le nombre décimal $x = (0,1)_{10}$ est converti et stocké en machine, avec une erreur relative δ_x majorée par $\varepsilon_M = 2^{-53}$ (cf § 5.2.3). L'erreur absolue sur la somme de 1000 nombres égaux à $0,1$ est donc estimée par

$$1000\varepsilon_M \left(\underbrace{|0,1| + |0,1| + \cdots |0,1|}_{1000 \text{ termes}} \right) = 1000^2 \times 2^{-53} \times 0,1.$$

```
» epsM=2^(-53)
epsM = 1.1102e-016
» DeltaS=1000^2*epsM*0.1
DeltaS = 1.1102e-11
```

2) On effectue le calcul de S, puis de $|S - 100|$ avec *Matlab* :

```
» S=0 ;
» for i=1 :1000,
   S=S+0.1 ;
end
» errObserv=abs(S-100)
errObserv = 1.4069e-12
```

L'erreur observée est bien inférieure à la majoration d'erreur $DeltaS$ trouvée.

Exercice 5.9.2

1) On transforme le système d'équations

$$\begin{cases} x_1 + x_2 = S \\ x_1.x_2 = P \end{cases}$$

en exprimant, dans la première équation, x_2 en fonction de x_1, puis en substituant la valeur trouvée dans la seconde équation.

```
» syms S P x1 x2
» eq1=x1+x2-S ;
» eq2=x1*x2-P ;
» sol2=solve(eq1,x2)
sol2 = -x1+S
» eq=subs(eq2,x2,sol2)
eq =x1*(-x1+S)-P
» collect(eq,x1)
ans = -x1^2+S*x1-P
```

On retrouve ainsi un résultat classique : x_1 vérifie l'équation

$$(x_1)^2 - S.x_1 + P = 0.$$

2) On calcule numériquement la valeur de la plus grande des solutions :

```
» S=5e7 ;P=1 ;
» delta=S^2-4*P
delta = 2.5000e+15
» x1=(S+sqrt(delta))/2
x1 = 5.0000e+07
```

3) On calcule la deuxième solution selon les deux méthodes indiquées.

```
» x2a=S-x1
x2a = 1.4901e-08
```

4)

```
» x2b=P/x1
x2b = 2.0000e-08
```

5) Le calcul de x_{2a} engendre une erreur de cancellation. Le calcul de x_{2b} est plus fiable.

6) On calcule l'erreur relative sur le résultat :

```
» errRel=abs(x2a-x2b)/x2b
errRel = 0.2549
```

Elle est de l'ordre de 25%.
Dans le calcul machine du discriminant $delta$, un phénomène d'absorption a pu engendrer une erreur, mais on sait que celle-ci, en valeur relative, est nécessairement inférieure ou égale à ε_M.

```
» epsM=2^(-53)
epsM = 1.1102e-16
» errRel/epsM
ans = 2.2963e+15
```

La cancellation a multiplié cette erreur initiale par plus de deux millions de milliards !

Exercice 5.9.3

1) On effectue les deux calculs, en respectant l'ordre imposé par les parenthèses.

```
» a=1 ;b=2^53 ;c=-2^53 ;
» s1=(a+b)+c
s1 = 0
» s2=a+(b+c)
s2 = 1
```

2) Dans le calcul de s_1, il y a eu absorption de a par b. Lorsqu'on ajoute c, un phénomène de cancellation se produit. Dans le calcul de s_2, le phénomène de cancellation n'a pas lieu car on calcule $b + c$ avec les valeurs exactes de b et c.

Exercice 5.9.4

1) On définit les deux expressions symboliques

$$d_1 = 1000 - \sqrt{999999}, \quad d_2 = \frac{1}{1000 + \sqrt{999999}},$$

et, en calculant leur différence, on vérifie qu'elles sont bien égales.

```
» d1 = sym('1000-sqrt(999999)')
d1 =1000-sqrt(999999)
» d2=sym('1/(1000+sqrt(999999))')
d2 =1/(1000+sqrt(999999))
» simplify(d1-d2)
ans =0
```

2) On calcule les valeurs approchées de d_1 et d_2 dans l'arithmétique choisie.

```
» digits(6) ;
» racineApp=vpa('sqrt(999999)')
racineApp =999.999
» d1App=vpa(1000-racineApp)
d1App =.1e-2
» denom2App=vpa((1000+racineApp))
denom2App =2000.00
» d2App =vpa(1/denom2App)
d2App= .500000e-3
```

Des deux résultats $d_{1App} = 0,01$ et $d_{2App} = 0,005$, le second est le plus fiable, car le premier peut générer une erreur de cancellation, 1000 et $\sqrt{999999}$ étant deux nombres très proches. On vérifie cette affirmation en calculant numériquement, avec toute la précision de la machine, les valeurs de d_1 et d_2.

```
» double(d1)
ans = 5.0000e-004
» double(d2)
ans = 5.0000e-004
```

Exercice 5.9.5

1) Pour tout $x \in [1,\ e]$ et pour tout $n \geqslant 0$, on a

$$(\ln x)^n \geqslant 0.$$

L'intégrale d'une fonction positive étant elle-même positive, on a

$$I_n = \int_1^e [\ln x]^n \, dx \geqslant 0.$$

Cette même propriété montre que

$$I_n - I_{n+1} \geqslant 0,$$

car

$$I_n - I_{n+1} = \int_1^e [\ln x]^n \, (1 - \ln x) \, dx,$$

et pour $x \in [1,\ e]$

$$1 - \ln x \geqslant 1 - \ln e = 0.$$

La suite $(I_n)_{n \geqslant 0}$ étant décroissante et minorée par 0, elle admet une limite finie $l \geqslant 0$.

2) On a

$$I_0 = \int_1^e 1\,dx = [x]_1^e = e - 1.$$

On effectue une intégration par parties pour calculer I_n. On pose :

$$\begin{cases} u(x) = (\ln x)^n \\ v'(x) = 1. \end{cases}$$

On calcule u' et v :

```
» clear ; syms x n real
» e=sym('exp(1)') ;
» u=log(x)^n ;
» vPrime= 1 ;
» uPrime=simplify(diff(u,x))
uPrime = log(x)^(n-1)*n/x
» v=int(vPrime,x)
v = x
```

soit

$$\begin{cases} u'(x) = n\,(\ln x)^{n-1}.\dfrac{1}{x} \\ v(x) = x \end{cases}$$

D'où

$$I_n = \int_1^e u(x).v'(x)dx = [u(x).v(x)]_1^e - \int_1^e u'(x).v(x)dx.$$

```
» simplify(subs(u*v,x,e)-subs(u*v,x,1))
ans = exp(1)
» simplify(uPrime*v)
ans =log(x)^(n-1)*n
```

Donc

$$\begin{aligned} I_n &= [x.\,(\ln x)^n]_1^e - n\int_1^e [\ln x]^{n-1}\,dx \\ &= e - nI_{n-1}. \end{aligned}$$

De cette égalité, on déduit que nécessairement $l = 0$, sinon, on aurait par passage à la limite

$$l = \lim_{n\to\infty}(e - nI_{n-1}) = e - \infty.l = -\infty,$$

ce qui est absurde car $l \geqslant 0$.

3) On définit I_0, $e = \exp(1)$, et on place dans un tableau les valeurs successives de I_n, pour $n \geqslant 1$.

```
» clear
» e=exp(1)
» I0=e-1 ;
» I(1)=e-I0 ;
» for n=2 :30,
   I(n)=e-n*I(n-1);
end
» I(1 :10)
ans =
Columns 1 through 7
1.0000 0.7183 0.5634 0.4645 0.3956 0.3447 0.3055
Columns 8 through 10
0.2744 0.2490 0.2280
» I(30)
ans = 2.9228e+16
```

4) La valeur obtenue pour I_{30} est contradictoire avec les propriétés de la suite (I_n) , décroissante et tendant vers 0. Expliquons ce phénomène :
si on note $\widetilde{I}$ la valeur numérique approchée de I_n obtenue par la relation

$$I_n = e - nI_{n-1},$$

on a

$$\left|\widetilde{I_n} - I_n\right| \simeq n \left|\widetilde{I}_{n-1} - I_{n-1}\right|,$$

en négligeant les erreurs d'affectation commises sur e et $\widetilde{I}$. L'erreur sur I_{30} peut donc être évaluée à

$$\left|\widetilde{I_{30}} - I_{30}\right| \simeq (30!) \left|\widetilde{I_0} - I_0\right|.$$

Même si l'erreur sur I_0 est une simple erreur d'affectation évaluée à $\varepsilon_M . I_0$, l'erreur commise sur I_{30} calculée selon la formule ci-dessus donne

```
» F30=prod(1 :1 :30)
F30 = 2.6525e+32
» errI30=F30*2^(-53)*I0
errI30 = 5.0602e+16
```

$$\left|\widetilde{I_{30}} - I_{30}\right| \simeq 5 \times 10^{16}.$$

5) De l'égalité

$$I_n = e - nI_{n-1},$$

on déduit aussi

$$I_{n-1} = \frac{e - I_n}{n}.$$

Si on utilise cette formule pour calculer I_{n-1} en fonction de I_n, on aura, toujours en négligeant les erreurs d'affectation

$$\left|\widetilde{I}_{n-1} - I_{n-1}\right| \simeq \frac{\left|\widetilde{I}_n - I_n\right|}{n}.$$

Si on calcule ainsi de proche en proche I_{30} à partir d'une valeur estimée de I_{50}, on aura

$$\left|\widetilde{I}_{30} - I_{30}\right| \simeq \frac{\left|\widetilde{I} - I_{50}\right|}{50 \times 49 \times ... \times 31}.$$

```
» M=prod(50 :-1 :31)
M = 1.1466e+32
```

L'erreur initiale sur I_{50} sera ainsi divisée par plus de 10^{32}.
Compte tenu des propriétés de la suite (I_n) , il est raisonnable de partir de l'estimation $I_{50} = 0$.

```
» I=zeros(1,50);
» I(50)=0;
» for n=50 :-1 :2,
   I(n-1)=(e-I(n))/n;
end
» I(30)
ans = 0.0850
» I(1 :10)
ans =
Columns 1 through 7
1.0000 0.7183 0.5634 0.4645 0.3956 0.3447 0.3055
Columns 8 through 10
0.2744 0.2490 0.2280
```

Mais partir d'une estimation autre de I_{50} ne changera pas les valeurs obtenues pour I_{30} et pour $I_{10}, I_9, \cdots, I_0$.

```
» I=zeros(1,50);
» I(50)=1e9;
for n=50 :-1 :2,
   I(n-1)=(e-I(n))/n;
end
» I(30)
ans = 0.0850
» I(1 :10)
ans =
Columns 1 through 7
1.0000 0.7183 0.5634 0.4645 0.3956 0.3447 0.3055
Columns 8 through 10
0.2744 0.2490 0.2280
```

6) On constate ici qu'on obtient des résultats aberrants en utilisant une formule exacte, mais générant une amplification catastrophique des erreurs. L'utilisation de la formule de récurrence inverse, et d'une estimation de I_{50} donne des résultats plus fiables pour $I_{30}, \ldots, I_2, I_1$..

Exercice 5.9.6

1) On calcule numériquement les termes successifs de la suite (a_n).

```
» clear ; a(1)=11/2 ; a(2)=61/11 ;
» for n=2 :29,
   a(n+1)= 111-1130/a(n)+3000/(a(n)*a(n-1));
end
» a
a =
Columns 1 through 7
5.5000 5.5455 5.5902 5.6334 5.6746 5.7133 5.7491
Columns 8 through 14
5.7818 5.8113 5.8377 5.8609 5.8813 5.8982 5.8980
Columns 15 through 21
5.6470 0.9683 -507.3216 107.1206 100.3959 100.0235 100.0014
Columns 22 through 28
100.0001 100.0000 100.0000 100.0000 100.0000 100.0000 100.0000
Columns 29 through 30
100.0000 100.0000
```

2) On effectue de même le calcul symbolique.

```
» aSym(1)=sym('11/2');aSym(2)=sym('61/11');
» for n=2 :29,
   aSym(n+1)= 111-1130/aSym(n)+3000/(aSym(n)*aSym(n-1));
end
» double(aSym)
ans =
Columns 1 through 7
5.5000 5.5455 5.5902 5.6334 5.6746 5.7133 5.7491
Columns 8 through 14
5.7818 5.8113 5.8377 5.8610 5.8814 5.8992 5.9145
Columns 15 through 21
5.9277 5.9391 5.9487 5.9569 5.9638 5.9696 5.9746
Columns 22 through 28
5.9787 5.9822 5.9851 5.9876 5.9896 5.9913 5.9928
Columns 29 through 30
5.9940 5.9950
```

On observe à partir du quinzième terme une différence considérable entre les résultats fournis par les deux modes de calcul. On peut à priori juger plus fiables les résultats obtenus en convertissant en numérique les valeurs exactes : on évite ainsi propagation et risques d'amplification d'erreurs.

3) Partant de

$$\begin{cases} x_0 & = 1 \\ x_{n+1} & = a_{n+1}.x_n\ , \quad n \geqslant 0, \end{cases}$$

on peut affirmer que, pour tout $n \geqslant 0$, $x_n \neq 0$, et on a pour $n \geqslant 2$

$$a_{n+1} = \frac{x_{n+1}}{x_n}, \quad a_n = \frac{x_n}{x_{n-1}}, \quad a_{n-1} = \frac{x_{n-1}}{x_{n-2}}.$$

On reporte dans la relation

$$a_{n+1} = 111 - \frac{1130}{a_n} + \frac{3000}{a_n a_{n-1}},$$

et on obtient

$$\frac{x_{n+1}}{x_n} = 111 - \frac{1130}{\dfrac{x_n}{x_{n-1}}} + \frac{3000}{\dfrac{x_n}{x_{n-1}}.\dfrac{x_{n-1}}{x_{n-2}}},$$

soit

$$\frac{x_{n+1}}{x_n} = 111 - \frac{1130x_{n-1}}{x_n} + \frac{3000x_{n-2}}{x_n}.$$

D'où la relation (R)

$$x_{n+1} = 111x_n - 1130x_{n-1} + 3000x_{n-2}.$$

4) On remplace x_n par r^n, x_{n+1} par r^{n+1}, x_{n-1} par r^{n-1}, x_{n-2} par r^{n-2} et on cherche r tel que

$$r^{n+1} - (111r^n - 1130r^{n-1} + 3000r^{n-2}) = 0.$$

On obtient l'équation

$$r^{n+1} - 111r^n + 1130r^{n-1} - 3000r^{n-2} = 0.$$

r étant supposé différent de 0, on peut simplifier par r^{n-2}, d'où l'équation équivalente

$$r^3 - 111r^2 + 1130r - 3000 = 0.$$

On résout alors l'équation du troisième degré obtenue.

```
» syms r real
» eq1 = r^3-111*r^2+1130*r-3000;
» solve(eq1)
ans =
[ 100]
[ 5]
[ 6]
```

Ses solutions sont donc

$$\begin{cases} r_1 = 100 \\ r_2 = 6 \\ r_3 = 5. \end{cases}$$

5) On définit

$$x_n = k_1 . (r_1)^n + k_2 . (r_2)^n + k_3 . (r_3)^n ,$$

et, en exprimant x_{n+1}, x_{n-1}, x_{n-2} à partir de cette relation, on montre que la relation (R) est vérifiée.

```
» r1=100 ;r2=6 ;r3=5 ;
» syms k1 k2 k3 n real
» xn=k1*r1^n+k2*r2^n+k3*r3^n;
» xnPlus1=subs(xn,n,n+1);
» xnMoins1=subs(xn,n,n-1);
» xnMoins2=subs(xn,n,n-2);
» eq=simplify(xnPlus1-(111*xn-1130*xnMoins1+3000*xnMoins2))
eq = 0
```

6) On exprime x_0, x_1, x_2 en fonction de k_1, k_2, k_3.

```
» x0=subs(xn,n,0)
x0 = k1+k2+k3
» x1=subs(xn,n,1)
x1 = 100*k1+6*k2+5*k3
» x2=subs(xn,n,2)
x2 = 10000*k1+36*k2+25*k3
```

On doit donc résoudre le système

$$\begin{cases} k_1 + k_2 + k_3 = 1 \\ 100k_1 + 6k_2 + 5k_3 = a_1 \\ 10000k_1 + 36k_2 + 25k_3 = a_2 a_1. \end{cases}$$

Pour obtenir la solution exacte de ce système, on utilise le calcul symbolique : on définit la matrice A_s des coefficients du système, la matrice B_s des seconds membres, et on calcule la matrice colonne

$$K = A^{-1}B$$

des solutions :

```
» As=sym([1 1 1 ; 100 6 5 ; 100^2 6^2 5^2 ]) ;
» Bs=[1 ;aSym(1) ; aSym(2)*aSym(1)] ;
» K=As^(-1)*Bs ;
» K1 =K(1)
K1 = 0
» K2 =K(2)
K2 = 1/2
» K3=K(3)
K3 = 1/2
```

D'où

$$K_1 = 0,\ K_2 = 1/2,\ K_3 = 1/2.$$

En effectuant les mêmes calculs numériquement, on obtiendra des valeurs approchées des solutions :

```
» A=[1 1 1 ; 100 6 5 ; 100^2 6^2 5^2 ] ;
» B=[1 ;a(1) ; a(2)*a(1)] ;
» KApp=A^(-1)*B ;
» K1App =KApp(1)
K1App = 8.6736e-19
» K2App =KApp(2)
K2App = 0.5000
» K3App= KApp(3)
K3App = 0.5000
```

7) On définit

$$b_n = \frac{K_1.\,(r_1)^n + K_2.\,(r2)^n + K_3.\,(r3)^n}{K_1.\,(r_1)^{n-1} + K_2.\,(r2)^{n-1} + K_3.\,(r3)^{n-1}},$$

et on calcule b_1 et b_2 :

```
» num=K1*r1^n+K2*r2^n+K3*r3^n;
» den=subs(num,n,n-1);
» bn=num/den
bn = (1/2*6^n+1/2*5^n)/(1/2*6^(n-1)+1/2*5^(n-1))
» b1=subs(bn,n,sym(1))
b1 = 11/2
» b2=subs(bn,n,sym(2))
b2 = 61/11
```

Ainsi

$$\begin{cases} b_1 = a_1 \\ b_2 = a_2. \end{cases}$$

On suppose (hypothèse de récurrence) :

$$\begin{cases} b_{n-1} = a_{n-1} \\ b_n = a_n, \end{cases}$$

et on montre que $b_{n+1} = a_{n+1}$:

```
» an=bn ;
» anMoins1=subs(bn,n,n-1);
» bnPlus1=subs(bn,n,n+1)
bnPlus1 = (1/2*6^(n+1)+1/2*5^(n+1))/(1/2*6^(+1/2*5^n)
» anPlus1=simplify(111-1130/an+3000/(an*anMoins1))
anPlus1 = (6^(n+1)+5^(n+1))/(6^n+5^n)
» simplify(bnPlus1-anPlus1)
ans = 0
```

On a ainsi montré que, pour tout $n \geqslant 1$,

$$a_n = b_n = \frac{1/2.\,(6)^n + 1/2.\,(5)^n}{1/2.\,(6)^{n-1} + 1/2.\,(5)^{n-1}}.$$

8) Pour obtenir la limite de la suite (a_n) , on écrit

$$a_n = b_n = \frac{1/2.\,(6)^n\,[1 + (5/6)^n]}{1/2.\,(6)^{n-1}\,[1 + 1/2.\,(5/6)^{n-1}]} = 6.\frac{1 + (5/6)^n}{1 + 1/2.\,(5/6)^{n-1}}.$$

D'où

$$\lim_{n\to\infty} a_n = 6.$$

Ce résultat se vérifie avec *Matlab* :

```
» an
an = (1/2*6^n+1/2*5^n)/(1/2*6^(n-1)+1/2*5^(n-1))
» limit(an,n,inf,'left')
ans = 6
```

9) Si on remplace K_1, K_2, K_3 par $K_{1App}, K_{2App}, K_{3App}$ la suite a maintenant pour terme général

$$\begin{aligned} a_{nApp} &= \frac{K_{1App}(100)^n + K_{2App}.\,(6)^n + K_{3App}.\,(5)^n}{K_{1App}(100)^{n-1} + K_{2App}.\,(6)^{n-1} + K_{3App}.\,(5)^{n-1}} \\ &= 100\frac{K_{1App} + K_{2App}.\,(6/100)^n + K_{3App}.\,(5/100)^n}{K_{1App} + K_{2App}.\,(6/100)^{n-1} + K_{3App}.\,(5/100)^{n-1}}, \end{aligned}$$

et, comme $K_{1App} \neq 0$, on a

$$\lim_{n\to\infty} a_{nApp} = 100.$$

Le calcul sous *Matlab* donne :

```
» numApp=K1App*r1^n+K2App*r2^n+K3App*r3^n;
» denApp=subs(numApp,n,n-1);
» anApp=numApp/denApp;
» limit(anApp,n,inf,'left')
ans = 100
```

On retrouve les résultats de la première question : la suite (a_n) a pour limite 6, mais des calculs numériques approchés conduisent à une suite (a_{nApp}) convergeant vers 100.

Chapitre 6

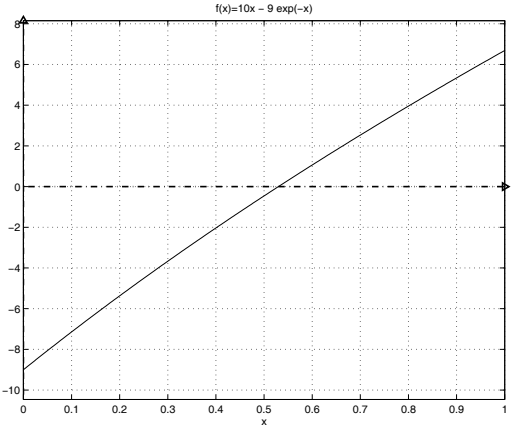
Approximation de racines d'équations

C'est seulement pour certaines équations bien particulières que les procédés classiques de résolution permettent d'exprimer les solutions exactes. Un exemple typique est celui des équations du second degré, en utilisant le discriminant. Dans de nombreux cas, on peut seulement localiser les solutions, et en calculer des valeurs numériques approchées.

Considérons par exemple, l'équation

$$10x - 9e^{-x} = 0.$$

Les méthodes usuelles de transformation (transposition, utilisation de la fonction logartihme, ...) ne permettent pas de résoudre algébriquement cette équation. Pourtant, on observe graphiquement qu'elle admet une solution unique sur $[0, 1]$.



Dans ce chapitre, on va exposer les principales méthodes itératives de résolution d'une équation de la forme

$$f(x) = 0,$$

où f est une fonction continue définie sur un intervalle $[a, b]$. On se placera dans le cas où, localement, il y a une unique racine, pour en donner un algorithme d'approximation.

6.1. Méthode de la dichotomie

Cette méthode consiste en une succession de divisions par deux de l'intervalle pour approcher de plus en plus la racine de l'équation $f(x) = 0$, jusqu'à ce qu'une précision ε soit atteinte.

6.1.1. *Hypothèses sur la fonction f*

On se place dans le cas où la fonction

$$f : [a,\ b] \to \mathbb{R}$$

vérifie les hypothèses :

$$\left\|\begin{array}{ll}(D1) & f \text{ est continue sur } [a, b],\\ (D2) & f \text{ est strictement monotone sur } [a, b],\\ (D3) & f(a).f(b) < 0,\end{array}\right.$$

ce qui assure l'existence et l'unicité de la racine $c \in [a, b]$.

6.1.2. *Algorithme de la méthode*

On partage $[a, b]$ en deux intervalles égaux $[a,\ \frac{a+b}{2}]$ et $[\frac{a+b}{2}, b]$.

Si le signe de $f((a + b)/2)$ est le même que celui de $f(a)$, la racine c appartient à l'intervalle $[\frac{a+b}{2}, b]$.

Sinon, elle appartient à l'intervalle $[a,\ \frac{a+b}{2}]$.

On réitère le procédé avec l'intervalle obtenu contenant c .

On arrête l'itération lorsque la longueur de l'intervalle devient inférieure à un nombre ε fixé au départ.

Remarque

A l'étape n, c appartient à l'intervalle de travail, qui a pour longueur

$$\frac{b-a}{2^n}.$$

D'une itération à la suivante, l'erreur est donc multipliée par $1/2$.

6.1.3. *Exemple*

On considère l'équation $10x - 9e^{-x} = 0$. La fonction f définie sur $[0, 1]$ par

$$f(x) = 10x - 9e^{-x}$$

est continue, dérivable, et sa dérivée f' vérifie

$$f'(x) = 10 + 9e^{-x} > 0,$$

donc f est strictement croissante sur $[0, 1]$.

De plus $f(0) = -9$ et $f(1) = 10 - 9/e \simeq 6,69$ sont de signes contraires.

On peut donc utiliser la méthode de dichotomie pour calculer à 10^{-6} près la solution de l'équation proposée.

Le nombre n de termes à calculer doit vérifier :

$$\frac{1-0}{2^n} \leqslant 10^{-6},$$

soit

$$n \geqslant \frac{\ln(10^6)}{\ln(2)}.$$

On crée le fichier *f1.m*

```
function y=f1(x)
y=10*x-9*exp(-x);
```

On évalue le nombre n_0 de termes à calculer (sous *Matlab*, **ceil(x)** donne le plus petit entier supérieur ou égal à x)

```
» n0=ceil(log(1e6)/log(2))
n0 = 20
```

D'où

```
» a = 0 ; b = 1 ;
» for i = 1 :20
   m=(a+b)/2 ;
   f1DeM = f1(m) ;
   if (f1DeM >0) b = m ; else a = m ; end
end
```

On affiche l'intervalle obtenu

```
» format long
» [a b]
ans = 0.52983283996582    0.52983379364014
```

A 10^{-6} près, la solution est 0.529833.

6.1.4. *En conclusion*

La méthode de dichotomie a l'avantage d'exiger peu d'hypothèses sur la fonction. Elle sert parfois de moyen de calcul d'une initialisation pour les algorithmes des autres méthodes. L'inconvénient majeur de cette méthode est la lenteur de convergence de son algorithme.

6.2. Méthode des approximations successives (ou du point fixe)

Parmi les méthodes de résolution de l'équation

$$f(x) = 0,$$

la méthode dite des approximations successives (ou du point fixe) est la plus importante. Son principe est basé sur la construction d'une suite itérative approchant de plus en plus la racine exacte, son premier élément (appelé initialisation) pouvant être n'importe quel point de l'intervalle de travail $[a, b]$.

La méthode du point fixe s'applique à des équations de la forme

$$\varphi(x) = x.$$

On peut toujours écrire l'équation $f(x) = 0$ sous une forme équivalente de ce type.

Par exemple, l'équation $10x - 9e^{-x} = 0$ est équivalente à

$$x = \frac{9}{10}e^{-x}.$$

On prendra garde de ne pas confondre la fonction f et la fonction φ.

6.2.1. *Hypothèses sur la fonction* φ

On se place dans le cas où la fonction

$$\varphi : [a, b] \to \mathbb{R}$$

vérifie les hypothèses :

$$\left\| \begin{array}{ll} (F1) & \varphi \textit{ est continue et dérivable sur } [a, b], \\ (F2) & \varphi \textit{ prend ses valeurs dans } [a, b], \\ (F3) & \exists M \in]0, 1[: \forall x \in [a, b] \quad |\varphi'(x)| \leqslant M. \end{array} \right.$$

On dira que φ est une contraction stricte.

6.2.2. *Théorème du point fixe*

Lorsque φ vérifie les trois hypothèses $(F1)$, $(F2)$, $(F3)$, il existe une unique racine c de l'équation $\varphi(x) = x$, appelée point fixe de φ.

Considérons en effet la fonction définie par

$$g(x) = \varphi(x) - x,$$

qui est strictement décroissante puisque

$$g'(x) = \varphi'(x) - 1 < 0$$

grâce à $(F3)$. Alors, d'après l'hypothèse $(F2)$, on a

$$\left\{ \begin{array}{l} g(a) = \varphi(a) - a \geqslant 0 \\ g(b) = \varphi(b) - b \leqslant 0. \end{array} \right.$$

Le théorème des valeurs intermédiaires donne alors l'existence d'un unique point c appartenant à $[a, b]$ tel que

$$g(c) = 0.$$

6.2.3. *Algorithme et estimation d'erreur*

6.2.3.1. *Algorithme*

On construit la suite des itérés de la manière suivante :

– on fixe un point x_0 quelconque de $[a, b]$,

– puis on définit

$$\left\{ \begin{array}{c} x_1 = \varphi(x_0) \\ x_2 = \varphi(x_1) \\ \vdots \\ x_{n+1} = \varphi(x_n). \end{array} \right.$$

6.2.3.2. *Majoration d'erreur*

Si c est le point fixe de φ, on a

$$\begin{aligned} |x_1 - c| &= |\varphi(x_0) - \varphi(c)| \leqslant M\,|x_0 - c| < |x_0 - c| \\ |x_2 - c| &= |\varphi(x_1) - \varphi(c)| \leqslant M\,|x_1 - c| < |x_1 - c|\,. \end{aligned}$$

En réitérant, on voit bien qu'on s'approche de plus en plus de la racine : c'est le principe des approximations successives. Plus précisément, on démontre par récurrence la **majoration d'erreur**

$$\left\|\forall n \geqslant 0 \quad |x_n - c| \leqslant M^n\,|x_0 - c| \leqslant M^n\,|b - a|\,.\right.$$

En effet, la propriété est évidemment vérifiée pour $n = 0$, et si on la suppose vérifiée à un rang $n-1$ donné, le théorème des accroissements finis implique l'existence d'un $\xi \in]a, b[$ tel que :

$$\begin{aligned} |x_n - c| &= |\varphi(x_{n-1}) - \varphi(c)| \\ &= |\varphi'(\xi)(x_{n-1} - c)| \\ &\leqslant M\,|x_{n-1} - c| \\ &\leqslant M.M^{n-1}\,|x_0 - c| \\ &\leqslant M^n\,|x_0 - c| \\ &\leqslant M^n\,|b - a|\,. \end{aligned}$$

Ainsi, la suite (x_n) converge vers c puisque, M appartenant à $]0, 1[$, on a

$$\lim_{n \longrightarrow \infty} M^n = 0.$$

6.2.3.3. *Test d'arrêt*

Fixons $\varepsilon > 0$. Pour que x_n soit une valeur approchée de c à ε près, il suffit que :

$$M^n\,|b - a| \leqslant \varepsilon$$

soit

$$n \geqslant \frac{\ln \varepsilon - \ln |b - a|}{\ln M}.$$

D'où l'algorithme de la méthode du point fixe :

Etant donné une fonction φ vérifiant les hypothèses $(F1)$, $(F2)$, $(F3)$ sur un intervalle $[a,\ b]$, et un nombre positif ε :
- *on calcule* $n_0 = E\left(\frac{\ln \varepsilon - \ln|b-a|}{\ln M}\right) + 1$,
- *on choisit* $x_0 \in [a,\ b]$,
- *pour* n *de* 1 *à* n_0, *on calcule* $x_n = \varphi(x_{n-1})$.

Une valeur approchée à ε près de la racine c est x_{n_0}.

6.2.3.4. *Remarque*

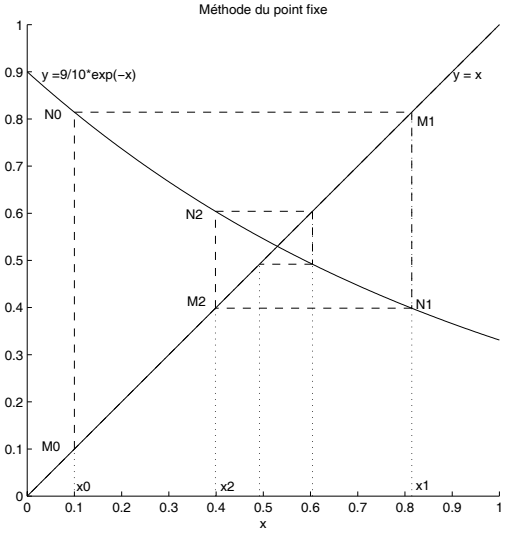
On peut construire graphiquement la suite des itérés (x_i), à l'aide de la ligne polygonale $[M_0N_0M_1N_1M_2N_2...]$, où M_i a pour coordonnées

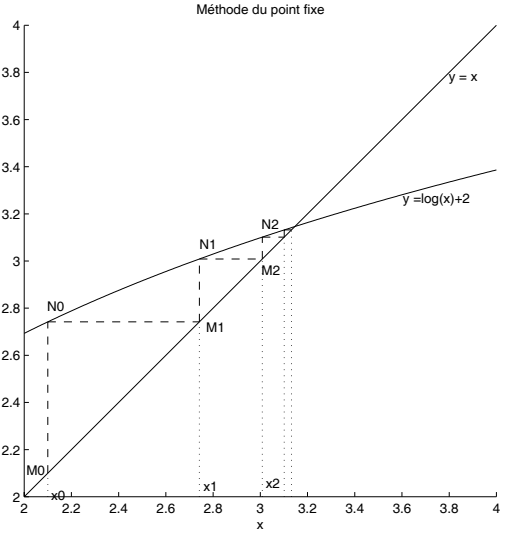
$$\begin{pmatrix} x_i \\ x_i \end{pmatrix}$$

et N_i a pour coordonnées

$$\begin{pmatrix} x_i \\ \varphi(x_i) \end{pmatrix} = \begin{pmatrix} x_i \\ x_{i+1} \end{pmatrix}.$$

Ainsi, $[M_iN_i]$ est le segment "vertical" joignant les points d'abscisse x_i de la droite d'équation $y = x$ et de la courbe d'équation $y = \varphi(x)$, et $[N_iM_{i+1}]$ est le segment "horizontal" joignant les points d'ordonnée x_{i+1} de ces deux mêmes figures (voir exercice 6.5.4).





6.2.4. *Exemple*

Pour calculer à 10^{-6} près la solution, dans l'intervalle $[0,\,1]$, de l'équation

$$x = \frac{9}{10}e^{-x}$$

par la méthode du point fixe, on procède comme suit :

– On définit la fonction φ, telle que

$$\varphi\,(x) = \frac{9}{10}e^{-x}.$$

Cette fonction est continue et dérivable sur $[0,\,1]$.

– Pour vérifier l'hypothèse $(F2)$, on étudie les variations de φ, en calculant $\varphi\,'$:

```
» syms x ;
» phiDeX = 9/10*exp(-x);
» phiPrimeDeX = diff( phiDeX )
   phiPrimeDeX = -9/10*exp(-x)
```

Comme $\varphi'(x) < 0$, φ décroît de $\varphi(1)$ à $\varphi(0)$

```
» phiDe1= subs( phiDeX ,1)
   phiDe1 = 0.3311
» phiDe0= subs( phiDeX ,0)
   phiDe0 = 0.9000)
```

Donc, $\varphi(x)$ prend ses valeurs dans l'intervalle

$$[0,3311\ldots,\ 0,9000\ldots] \subset [0,1],$$

et $(F2)$ est vérifiée.

– Pour vérifier $(F3)$, il faut en général étudier les variations de φ', donc calculer φ'', mais ici

$$|\varphi'(x)| = \varphi(x),$$

donc $|\varphi'(x)|$ a pour maximum $M = 0,9$.

– Le nombre n_0 de termes à calculer pour obtenir une valeur approchée de la solution à 10^{-6} près est donné par **:**

```
» n0=ceil ((log(10^(-6))-log(1-0))/log(9/10))
   n0 = 132
```

d'où $n_0 = 132$.

– On calcule les itérés successifs :

```
» X(1) = 0 ;
» for i = 1 :132 X(i+1) = 9/10*exp(-X(i)) ; end
» format short
» X(1 :6)
   ans =
   0 0.9000 0.3659 0.6242 0.4821 0.5557
» format long
» X(133)
   ans = 0.52983296563343
```

On retrouve la valeur approchée 0.529833 à 10^{-6} près.

6.2.5. *Vitesse de convergence*

Elle dépend de la valeur de M (voir hypothèse $F3$) :

– Si M est proche de 1, la convergence est lente. On a vu dans l'exemple précédent, où $M = 0,9$ qu'il fallait 132 termes pour obtenir une précision de $\varepsilon = 10^{-6}$.

– Si $M = 0.5$, on retrouve la vitesse de convergence de la méthode de dichotomie.

– Si M est proche de 0, on a une convergence rapide.

6.3. Méthode de Newton (ou de la tangente)

6.3.1. *Hypothèses et algorithme de Newton*

6.3.1.1. *Hypothèses*

On revient à la résolution de l'équation

$$f(x) = 0,$$

et on suppose que la fonction f vérifie les hypothèses suivantes :

$$\left\|\begin{array}{ll}(N1) & f \textit{ est continue sur } [a,b],\\ (N2) & f \textit{ est strictement monotone sur } [a,b],\\ (N3) & f(a).f(b) < 0,\\ (N4) & f \textit{ est dérivable sur } [a,b] \text{ et } f'(x) \neq 0 \textit{ sur } [a,b].\end{array}\right.$$

Les trois premières hypothèses garantissent l'existence et l'unicité d'une racine c de l'équation

$$f(x) = 0.$$

6.3.1.2. *Construction de l'algorithme*

L'idée principale de la méthode de Newton est de dire qu'au voisinage de la racine c la courbe représentative de la fonction peut être confondue avec la tangente en un point x_0 proche de c. Cela revient à confondre f avec son développement limité à l'ordre 1 en x_0 :

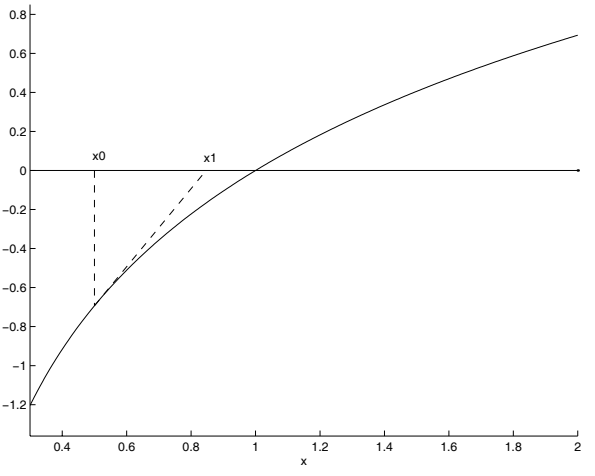
$$f(x) \simeq f(x_0) + f'(x_0)(x - x_0) \quad (x \to x_0).$$

La solution de l'équation $f(x) = 0$ peut donc être approchée par la résolution de

$$f(x_0) + f'(x_0)(x - x_0) = 0,$$

dont la solution est

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



x_1 est une première approximation de c. En réitérant le procédé ci-dessus, on construit la suite définie par :

$$\begin{cases} x_0 \text{ fixé proche de } c \\ x_{n+1} = x_n - \dfrac{f(x_n)}{f'(x_n)}, \quad n \geqslant 0, \end{cases}$$

appelée suite des itérés de l'algorithme de Newton.

6.3.1.3. *Remarque*

Cette suite est celle permettant de chercher le point fixe de la fonction

$$\varphi : x \mapsto \varphi(x) = x - \frac{f(x)}{f'(x)}$$

(voir § 6.2.1).

φ est dérivable si, et seulement si, f' l'est. Cela conduit à ajouter l'hypothèse suivante sur f

$$\|(N5) \qquad f \textit{ est deux fois dérivable sur } [a, b].$$

6.3.1.4. *Théorème*

On a le

Théorème
Sous les hypothèses $(N1), ..., (N5)$, *et pour* x_0 *choisi suffisamment proche de l'unique racine* c, *la suite des itérés de Newton converge vers* c.

On admettra ce théorème dont l'idée de démonstration repose sur le fait que la fonction φ vérifie toutes les hypothèses du point fixe dans un voisinage de c.

6.3.2. *Vitesse de convergence*

Notant toujours

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

on a

$$\varphi'(x) = \frac{f(x)f''(x)}{\left(f'(x)\right)^2}$$

et donc au point c

$$\varphi'(c) = 0.$$

La formule de Taylor-Lagrange à l'ordre deux donne l'existence d'un $\eta \in]a, b[$ tel que

$$\begin{aligned} \varphi(x) &= \varphi(c) + (x-c)\varphi'(c) + \frac{1}{2}(x-c)^2\varphi''(\eta) \\ &= c + \frac{1}{2}(x-c)^2\varphi''(\eta). \end{aligned}$$

d'où

$$\begin{aligned} |\varphi(x) - c| &= \frac{1}{2}(x-c)^2 \left|\varphi''(\eta)\right| \\ &\leqslant \frac{1}{2}M_2 \left|x-c\right|^2 \end{aligned}$$

où

$$M_2 = \max_{x\in[a,b]} \left|\varphi''(x)\right|.$$

En appliquant cette majoration aux itérés $x_n = \varphi(x_{n-1})$ et en supposant que pour tout n, $x_n \in [a, b]$, il vient

$$\frac{1}{2}M_2 \left|x_n - c\right| \leqslant \left|\frac{1}{2}M_2(x_{n-1} - c)\right|^2.$$

On montre alors par récurrence sur n :

$$\frac{1}{2}M_2 \left|x_n - c\right| \leqslant \left|\frac{1}{2}M_2(x_0 - c)\right|^{2^n}.$$

Par exemple, si on choisit l'initialisation x_0 telle que

$$|x_0 - c| \leqslant \frac{1}{5M_2},$$

il vient

$$|x_n - c| \leqslant \frac{2}{M_2} 10^{-2^n}.$$

Cette estimation traduit bien la vitesse de convergence de la méthode de Newton. Elle signifie que le nombre de décimales exactes **double** d'une itération à la suivante. Concrètement si on veut la racine c avec 1000 décimales exactes, x_{10} contiendrait ces décimales.

La convergence de l'algorithme de Newton est donc très rapide.

6.3.3. *Exemple*

L'étude de la fonction f définie sur $[0, 1]$ par

$$f(x) = 10x - 9e^{-x},$$

faite au paragraphe 6.1.3 , a permis de vérifier les hypothèses $(N1)$ à $(N4)$. La fonction dérivée

$$f' : x \longmapsto 10 + 9e^{-x}$$

est elle-même dérivable, donc l'hypothèse $(N5)$ est vérifiée. Partant de l'initialisation

$$x_0 = 0,$$

on calcule les huit premiers itérés de la méthode de Newton :

```
» syms x real
» fDeX=10*x-9*exp(-x);
» fPrimeDeX=diff(fDeX);
» X(1)=1;
for i=1 :7
   X(i+1)=X(i)-double(subs(fDeX,x,X(i))/subs(fPrimeDeX,x,X(i)));
end
» format long
» X
X =
Columns 1 through 4
1.00000000000000 0.49747368650014 0.52964976566513 0.52983295982123
Columns 5 through 8
0.52983296563343 0.52983296563343 0.52983296563343 0.52983296563343
```

En comparant aux résultats obtenus par la méthode de dichotomie (voir § 6.1.3), on observe que $x_3 = X(4)$ est une valeur approchée de la racine c à 10^{-6} près. Les itérés x_4 à x_7 contiennent les mêmes 14 premiers chiffres significatifs.

6.3.4. *Choix de l'initialisation* x_0

6.3.4.1. *Mise en garde*

Même si la fonction f vérifie toutes les hypothèses $(N1)$ à $(N5)$ sur un intervalle $[a, b]$, il se peut que pour certains choix de l'initialisation $x_0 \in [a, b]$, on ait, à un rang n, $x_n \notin [a, b]$. Dans ce cas, on n'est plus assuré de la convergence de la suite (x_n) vers la racine cherchée.

6.3.4.2. *Exemple*

Considérons l'équation

$$f(x) = 0,$$

avec

$$f(x) = x^3 - 4.53x^2 + 6.0291x - 2.218039, \quad x \in [1\ ,\ 2].$$

f, fonction polynôme est continue et dérivable sur $[1, 2]$, et on peut vérifier une à une les hypothèses d'application de la méthode de Newton :

1) $f(1)$ et $f(2)$ sont de signes contraires :

```
» syms x real
» f =x^3-4.53*x^2+6.0291*x-2.218039;
» fDe1 =subs(f,x,1)
fDe1 = 0.2811
» fDe2 = subs(f,x,2)
fDe2 = -0.2798
```

2) f' ne s'annule pas sur l'intervalle $[1,\ 2]$:

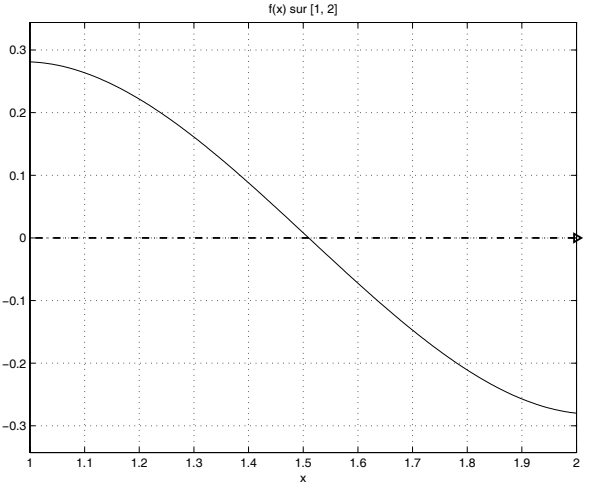
```
» fPrime=diff(f)
fPrime = 3*x^2-453/50*x+60291/10000
» S=solve(fPrime)
S =     [ 99/100]
        [ 203/100]
```

3) f' est de signe négatif, donc f est strictement décroissante sur l'intervalle $[1,\ 2]$:

```
» maple('solve(3*x^2-453/50*x+60291/10000<0)')
ans = RealRange(Open(99/100),Open(203/100))
```

On peut illustrer graphiquement ces propriétés :

```
» clf ; ezplot(f,[1 2]) ; grid on ;
» title('f(x) sur [1, 2]')
» dessineRepere
```



Pourtant, si on calcule les 10 premiers itérés de la méthode de Newton appliquée à cette fonction f, on obtient :

– en choisissant $x_1 = 1$:

```
» X(1)=1 ;
» for i=1 :10
    X(i+1)=X(i)-double(subs(f,x,X(i)))/double(subs(fPrime,x,X(i))) ;
end
» X
X = 1.0000 10.0958 7.2550 5.3716 4.1319 3.3295 2.8309
       2.5521 2.4351 2.4116 2.4107
```

– en choisissant $x_1 = 2$:

```
» X(1)=2 ;
» for i=1 :10
    X(i+1)=X(i)-double(subs(f,x,X(i)))/double(subs(fPrime,x,X(i))) ;
end
» X
X = 2.0000 -1.0785 -0.2883 0.2018 0.4742 0.5868 0.6085
       0.6093 0.6093 0.6093 0.6093
```

– en choisissant $x_1 = 1,1$:

```
» X(1)=1.1 ;
» for i=1 :10
   X(i+1)=X(i)-double(subs(f,x,X(i)))/double(subs(fPrime,x,X(i)));
end
» X
X = 1.1000 1.9591 0.6304 0.6086 0.6093 0.6093 0.6093
      0.6093 0.6093 0.6093 0.6093
```

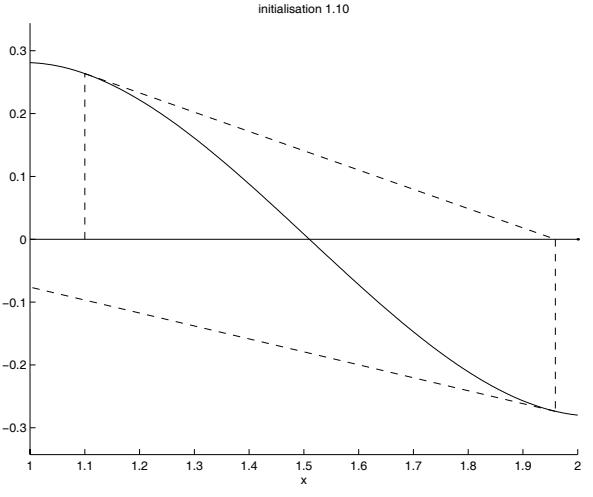
Dans les trois cas, on obtient des valeurs qui n'appartiennent pas à l'intervalle $[1, 2]$ dans lequel on cherche la solution.

Par contre, l'initialisation $x_1 = 1,15$ donne une suite qui semble converger vers la solution cherchée.

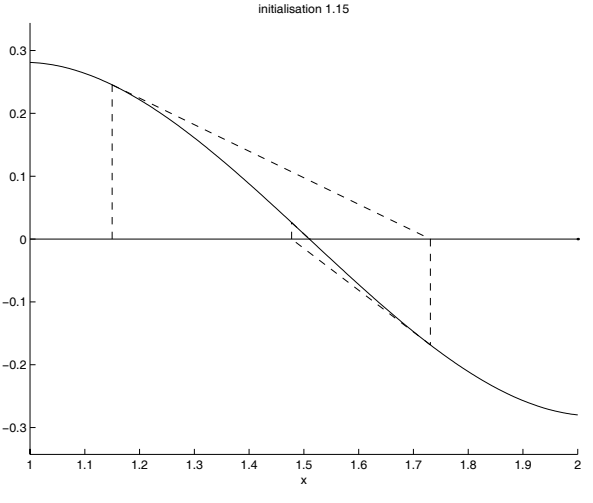
```
» X(1)=1.15 ;
» for i=1 :10
   X(i+1)=X(i)-double(subs(f,x,X(i)))/double(subs(fPrime,x,X(i)));
end
» X
X = 1.1500 1.7309 1.4776 1.5101 1.5100 1.5100 1.5100
      1.5100 1.5100 1.5100 1.5100
```

On peut expliquer graphiquement ce qui se passe (on utilise une fonction *newtonGraph* qui est présentée dans l' exercice 6.5.5) :

partant de $x_1 = 1,10$, la méthode de la tangente donne une valeur x_2 proche de 2, et une valeur x_3 n'appartenant plus à l'intervalle $[1,2]$,



alors que la suite des itérés converge vers la racine cherchée en partant de la valeur initiale $x_1 = 1,15$.



Le résultat suivant permet d'obtenir un choix de x_0 assurant la convergence de la méthode.

6.3.4.3. *Théorème*

Si f vérifie les hypothèses $(N1), ..., (N5)$,
si de plus $f''(x)$ ne s'annule pas et garde un signe constant sur $[a, b]$, alors :
en choisissant $x_0 \in [a, b]$ tel que $f(x_0)$ et $f''(x_0)$ soient de même signe,
la suite des itérés de la méthode de Newton converge vers la racine cherchée.

Mais cela suppose que $f''(x)$ ne s'annule pas sur $[a, b]$, ce qui n'est pas le cas de l'exemple précédent.

6.4. Plan pour la recherche d'une racine

A travers un exemple modèle, on met en œuvre un plan d'étude pour la recherche de solutions approchées d'une équation de la forme

$$(*) \qquad \varphi(x) = x,$$

et/ou de la forme

$$(**) \qquad f(x) = 0.$$

Ce plan est le suivant :

1) étude de f sur son domaine de définition et tracé de sa courbe représentative. Cette étape permettra la localisation de la ou des racines, et un choix adéquat d'un intervalle de travail $I = [a, b]$,

2) pour la méthode du point fixe :
   a) vérification des hypothèses $(F1)$, $(F2)$, $(F3)$ (voir § 6.2.1),
   b) choix de $x_0 \in [a, b]$ et mise en œuvre de l'algorithme de calcul approché.

3) pour la méthode de Newton :
   a) vérification des hypothèses $(N1)$ à $(N5)$ (voir § 6.3.1.1),
   b) choix de $x_0 \in [a, b]$ (en utilisant si possible le théorème 6.3.4.3), et mise en œuvre de l'algorithme de calcul approché.

6.4.1. *Exemple*

Soit à résoudre l'équation

$$(*) \qquad x = (x+1)\arctan\left(\frac{x+1}{x^2+2}\right).$$

Cette équation peut aussi s'écrire, pour $x \neq -1$, sous la forme

$$(**) \qquad \arctan\left(\frac{x+1}{x^2+2}\right) - \frac{x}{x+1} = 0.$$

1) On considère donc la fonction f définie par

$$f(x) = \arctan\left(\frac{x+1}{x^2+2}\right) - \frac{x}{x+1}.$$

Elle est définie, continue, dérivable sur $\mathbb{R}\backslash\{-1\}$. On entre son expression sous *Matlab*, on calcule et on factorise f' :

```
» syms x real
» f=atan((x+1)/(x^2+2))-x/(x+1);
» fPrime =simplify(diff(f))
fPrime= -(2*x^4+4*x^3+8*x^2+3)/(x+1)^2/(x^4+5*x^2+5+2*x)
» [N,D]=numden(fPrime)
N = -2*x^4-4*x^3-8*x^2-3
D = (x+1)^2*(x^4+5*x^2+5+2*x)
```

Ainsi

$$f'(x) = \frac{-2x^4 - 4x^3 - 8x^2 - 3}{(x+1)^2\,(x^4 + 5x^2 + 2x + 5)}.$$

La recherche des solutions de l'équation $f'(x) = 0$ avec *Matlab* montre que celle-ci n'admet pas de solution réelle :

```
» solve(fPrime);
» double(ans)
ans =
0.0996 - 0.5989i
0.0996 + 0.5989i
-1.0996 - 1.6913i
-1.0996 + 1.6913i
```

On montre par résolution d'inéquation que, pour tout $x \in \mathbb{R}\backslash\{-1\}$, on a $f'(x) < 0$.

```
» maple('solve(-(2*x^4+4*x^3+8*x^2+3)/(x+1)^2/(x^4+5*x^2+5+2*x)<0)')
ans =RealRange(-inf,Open(-1)), RealRange(Open(-1),inf)
```

On complète l'étude des variations de f par la recherche des limites aux bornes.

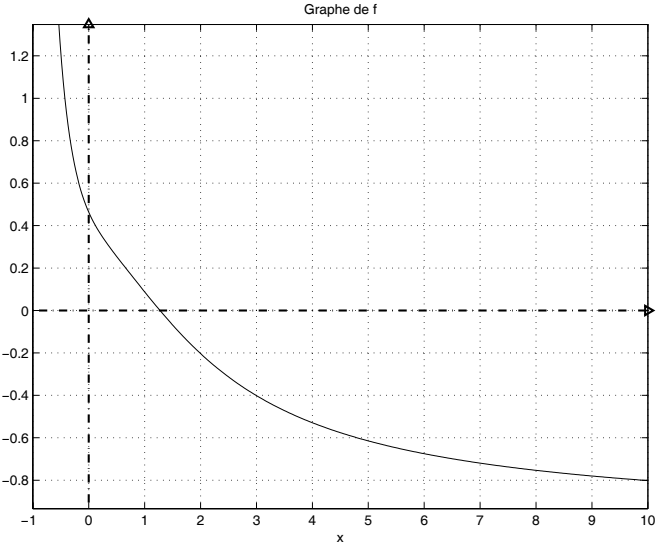
```
» limit(f,x,-inf,'right')
ans =-1
» limit(f,x,-1,'left')
ans =-inf
» limit(f,x,-1,'right')
ans = inf
» limit(f,x,inf,'left')
ans =-1
```

Ces calculs permettent d'obtenir le tableau de variations :

x	$-\infty$ $\quad -1$	-1 $\quad +\infty$
$f'(x)$	$-$	$-$
$f(x)$	$-1 \searrow -\infty$	$+\infty \searrow -1$

La stricte monotonie et la continuité de la fonction f sur l'intervalle $]-1,\ +\infty[$ permettent d'affirmer que l'équation $f(x) = 0$ admet une solution unique. On trace la courbe représentative de f,

```
» ezplot(f,-1,10)
» grid on
» title('Graphe de f')
» dessineRepere
```



et on vérifie par le calcul que la solution cherchée appartient à l'intervalle

$$[a, b] = [1, 2],$$

puisqu'on a $f(1) > 0$ et $f(2) < 0$:

```
» subs(f,x,1)
ans = 0.0880
» subs(f,x,2)
ans = -0.2030
```

2) Méthode du point fixe.
a) Pour la résolution de

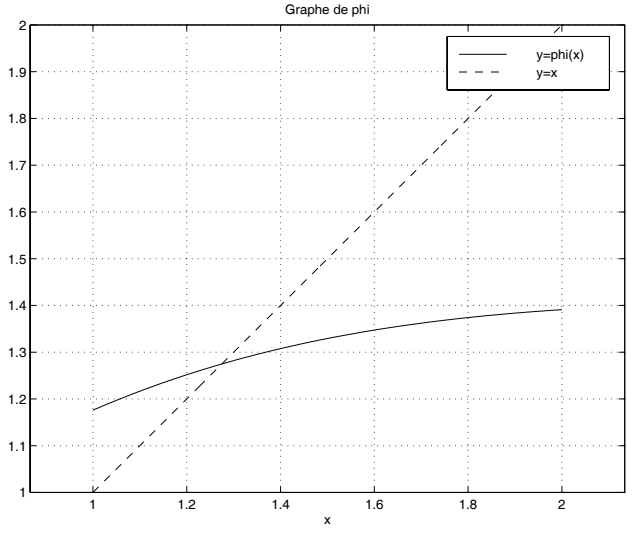
$$(*) \qquad x = (x+1)\arctan\left(\frac{x+1}{x^2+2}\right)$$

par la méthode du point fixe, on introduit la fonction φ définie par

$$\varphi(x) = (x+1)\arctan\left(\frac{x+1}{x^2+2}\right).$$

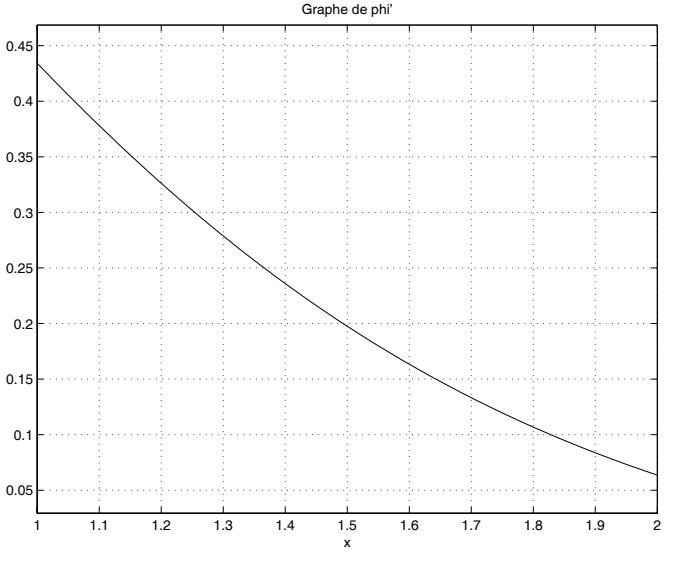
Cette fonction est continue et dérivable sur $[1,\ 2]$. On trace, sur une autre figure, sa courbe représentative (la commande *figure(n)* permet d'ouvrir une nouvelle figure, numérotée n, sans avoir à fermer la ou les figures en cours).

```
» syms x real
» phi=(x+1)*atan((x+1)/(x^2+2));
» figure(2)
» ezplot(phi,1,2); grid on ;hold on
» set(gca,'LineStyle','–')
» ezplot(x,1,2)
» axis equal ; axis auto
» legend('y=phi(x)','y=x')
» title('Graphe de phi')
```



On calcule sa dérivée φ', et on trace aussi la courbe représentative de φ' sur l'intervalle de travail.

```
» phiPrime = simplify(diff(phi));
» figure(3)
» ezplot(phiPrime,1,2)
» grid on
» title('Graphe de phi' ' ')
```



Graphiquement, on observe les propriétés :

- pour tout $x \in [1, 2]$, $\varphi(x) \in [1, 1.4] \subset [1, 2]$;
- pour tout $x \in [1, 2]$, $|\varphi'(x)| \leqslant 0.45 = M < 1$.

Pour vérifier ces propriétés par le calcul, on étudie les variations de φ', donc le signe de φ'' :

```
» phiSeconde =simplify(diff(phiPrime))
phiSeconde =
2*(3-12*x^3-4*x^4+2*x^5-10*x^2-28*x)/(x^4+5*x^2+5+2*x)^2
» solve(phiSeconde)
ans =
[ -1.90625]
[ -.105843-1.37614*i]
[ -.105843+1.37614*i]
[ .102881]
[ 4.01506]
```

Ainsi l'équation $\varphi''(x) = 0$ admet 5 solutions, dont 3 réelles (*Matlab* ne peut fournir ici que des valeurs approchées). Il n'y a pas de solution dans l'intervalle $[1, 2]$, ce qui montre, par continuité, que φ'' garde un signe constant dans cet intervalle. Par le calcul de $\varphi''(1)$, on détermine ce signe :

```
» subs(phiSeconde,1)
ans = -0.5799
```

Ainsi, φ'' est négative, donc φ' est décroissante sur l'intervalle $[1, 2]$. On calcule le minimum et le maximum de φ'.

```
» M1=subs(phiPrime,x,1)
M1 = 0.4342
» M2=subs(phiPrime,x,2)
M2 = 0.0636
```

On a donc, pour tout $x \in [1, 2]$,

$$0.0636... = M_2 \leqslant \varphi'(x) \leqslant M_1 = 0.4341...$$

d'où :

- pour tout $x \in [1, 2]$, $|\varphi'(x)| \leqslant M_1 < 1$.
- pour tout $x \in [1, 2]$, $\varphi'(x) > 0$.

φ est donc strictement croissante de $\varphi(1)$ à $\varphi(2)$. On calcule le minimum et le maximum de φ sur $[1, 2]$.

```
» minPhi=subs(phi,x,1)
minPhi = 1.1760
» maxPhi=subs(phi,x,2)
maxPhi = 1.3909
```

On a bien prouvé

$$\varphi\left([1,\ 2]\right) = [1,1760...,\ 1,3909...] \subset [1,\ 2]\,.$$

Toutes les hypothèses d'application de la méthode du point fixe sont vérifiées.

b) On calcule le nombre de termes n_0 nécessaire pour obtenir une valeur approchée de la racine à $\varepsilon = 10^{-6}$ près, en utilisant la formule

$$n_0 = E\left(\frac{\ln\varepsilon - \ln|b-a|}{\ln M}\right) + 1$$

```
» epsilon =1e-6 ;
» N0=ceil((log(epsilon)-log(2-1))/double(log(M1)))
N0 = 17
```

Partant de $x_0 = 1$, on calcule successivement $x_1, x_2, ..., x_{17}$. Ces valeurs sont placées dans un tableau $[X(1), ...X(18)]$ (attention au décalage d'indices).

```
» X(1)=1 ;
» for i=2 :N0+1, X(i)=double(subs(phi,x,X(i-1)));end
» format long
» X
X =
1.00000000000000 1.17600520709514 1.24379170581345 1.26557796182895
1.27210780086810 1.27402155079797 1.27457867309538 1.27474054108967
1.27478754376334 1.27480118996655 1.27480515165309 1.27480630177079
1.27480663566025 1.27480673259122 1.27480676073111 1.27480676890036
1.27480677127196 1.27480677196046
```

La solution , à 10^{-6} près, est donc $1,274806$.

3) Méthode de Newton

a) Pour la méthode de Newton, les hypothèses $(N1)$ à $(N4)$ ont dèjà été établies au cours de l'étude de f. La propriété $N5$ (dérivabilité de f') est elle aussi vérifiée.

b) On place dans le tableau X la suite des cinq premiers itérés de la méthode de Newton à partir de l'initialisation $X(1) = 1$:

```
» clear X ; X(1)=1 ;
» for i=1 :5,X(i+1)=X(i)-double(subs(f,x,X(i))/subs(fPrime,x,X(i)));end
» X
X =
1.00000000000000 1.26918443438080 1.27480320880089 1.27480677224065
1.27480677224209 1.27480677224209
```

On peut apprécier la vitesse de convergence en comparant la suite de valeurs obtenues, avec la solution approchée donnée directement par la fonction *solve* de *Matlab*.

```
» solve(f)
ans =1.27480677224209465805799444159 28
```

On a un chiffre significatif pour $X(1)$, deux pour $X(2)$, quatre pour$X(3)$, huit pour $X(4)$, et les quinze chiffres qu'on peut lire pour $X(5)$ sont également significatifs.

6.5. Exercices

6.5.1. *Méthode de dichotomie, de Newton et du point fixe*

On considère l'équation $f(x) = 0$, avec

$$f(x) = \cos x - xe^x, \quad x \in [0\ ,\ \frac{\pi}{2}].$$

1) Etudier les variations de f et montrer que cette équation admet une unique solution s dans $[0\ ,\ \frac{\pi}{2}]$.

2) Utilliser la méthode de dichotomie pour trouver une valeur approchée de s avec la précision 10^{-6}..

3) Vérifier que la méthode de Newton est applicable pour trouver une valeur approchée de s. En étudiant le signe de f'', indiquer un bon choix de x_0. Calculer alors les 10 premiers itérés de cette méthode.

4) On met l'équation $f(x) = 0$ sous la forme

$$x = \frac{\cos x}{e^x}.$$

a) Montrer que les hypothèses d'application de la méthode du point fixe ne sont pas vérifiées sur l'intervalle $[0\ ,\ \frac{\pi}{2}]$.

b) Montrer qu'elles le sont sur l'intervalle $[0.45,\ 0.6]$.

c) Combien de termes devrait-on calculer par la méthode du point fixe pour trouver une valeur approchée de s à 10^{-6} près ?

(solution p. 209)

6.5.2. *Méthode de Newton pour une fonction affine*

Appliquer la méthode de Newton à une fonction affine définie par $f(x) = \alpha x + \beta$, $(\alpha \neq 0)$, en prenant une initialisation x_0 quelconque.

(solution p. 212)

6.5.3. *Valeur approchée de* $\sqrt{2}$

En considérant la fonction définie par $f(x) = x^2 - 2$, utiliser la méthode de Newton pour construire une suite convergeant vers $\sqrt{2}$. En calculer les six premiers termes avec *Matlab*.

(solution p. 212)

6.5.4. *Programmation de la méthode du point fixe*

1) Ecrire une fonction

$$s = pointFixe(phi, var, x0, n)$$

qui place dans le tableau

$$[s(1)\ s(2)...s(n)]$$

les n premiers itérés de la méthode du point fixe appliquée à la fonction de la variable *var*, définie par l'expression symbolique *phi*, en partant de l'initialisation x_0 (on aura donc $s(1) = x_0$).

2) Ecrire une fonction

$$s = pointFixeGraphe(phi, var, x0, n, a, b)$$

qui, en plus des traitements de la question précédente, représente sur un graphique, dans un repère orthonormé :

- la fonction φ sur l'intervalle $[a, b]$;
- la droite d'équation $y = x$;
- la ligne polygonale $[M_0N_0M_1N_1...M_nN_n]$ présentée au paragraphe 6.2.3.1.

(solution p. 213)

6.5.5. *Programmation de la méthode de Newton*

1) Ecrire une fonction

$$s = newton(f, var, x0, n)$$

qui place dans le tableau

$$[s(1)\ s(2)...s(n)]$$

les n premiers itérés de la méthode de Newton appliquée à la fonction de la variable *var*, définie par l'expression symbolique *f*, en partant de l'initialisation $x_0 = s(1)$.

2) Ecrire une fonction

$$s = newtonGraphe(f, var, x0, n, a, b)$$

qui, en plus des traitements de la question précédente, représente sur un graphique :

- la fonction f sur l'intervalle $[a, b]$;
- les tangentes successives illustrant la méthode (voir paragraphe 6.3.4.2).

(solution p. 215)

6.6. Solutions

Exercice 6.5.1

1) Remarquons tout d'abord que la fonction f est indéfiniment dérivable. On la définit avec *Matlab*, et on calcule sa dérivée.

```
» syms x
» f = cos(x)-x*exp(x);
» fPrime = diff(f)
fPrime = -sin(x)-exp(x)-x*exp(x)
```

Sur l'intervalle $[0,\ \pi/2]$,

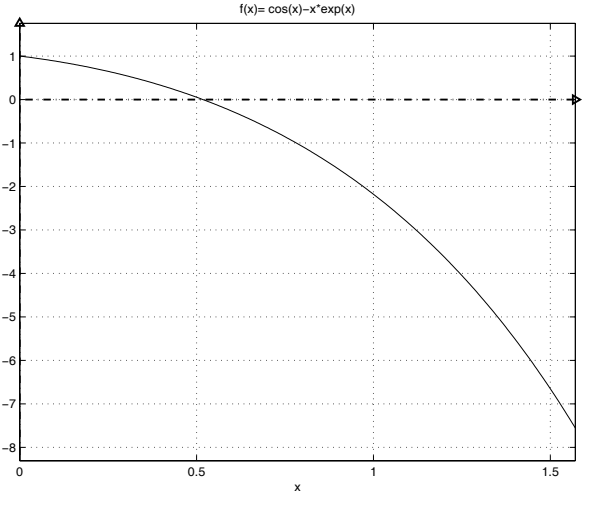
$$-\sin x \leqslant 0,\ \ -\exp(x) < 0,\ \ -x\exp(x) \leqslant 0,$$

donc $f'(x) < 0$.
On a $f(0) = 0$. On calcule une valeur approchée de $f(\pi/2)$.

```
» fDePisur2 = double(subs(f,x,pi/2))
fDePisur2 = -7.5563
```

f est continue et décroît strictement de 1 à $-7,556...$
L'équation $f(x) = 0$ admet donc une solution unique.
On construit la courbe représentative de f.

```
» clf
» ezplot(f,[0 pi/2])
» hold on ;grid on
» dessineRepere
» title('f(x)= cos(x)-x*exp(x)')
```



2) Les hypothèses de la méthode de dichotomie ont été vérifiées précédemment. On applique l'algorithme de la méthode, en comptant le nombre d'itérations pour obtenir la précision de 10^{-6}.

```
» g = 0 ; d = pi/2 ;
» n=0 ;
» while (d-g>1e-6)
    n=n+1 ;
    m = (g+d)/2 ;
    fdeM = cos(m) - m*exp(m) ;
    if fdeM>0
       g=m ;
    else
       d = m ;
    end
end
» format long
» [ g d]
ans = 0.51775671562829 0.51775746464235
» n
n =21
```

Ainsi, $s \in [0,5177567,\ 0,5177575]$.

3) D'après la première question, les hypothèses (N_1) à (N_5) sont vérifiées sur l'intervalle $[0,\ \pi/2]$. On calcule f'' :

```
» fSec=diff(fPrime)
fSec = -cos(x)-2*exp(x)-x*exp(x)
```

On a $f''(x) < 0$ sur $[0,\ \pi/2]$, et on choisit l'initialisation x_0 telle que $f(x_0)$ soit lui aussi négatif, par exemple

$$x_0 = \pi/2$$

On calcule alors $X(1) = x_0$, ..., $X(11) = x_{10}$.

```
» h = simplify(x-f/fPrime)
h = (x*sin(x)+x^2*exp(x)+cos(x))/(sin(x)+exp(x)+x*exp(x))
» X(1)= pi/2 ;
» for i = 1 :10,
   X(i+1)=double(subs(h,x,X(i)));
end
» X
X =
1.57079632679490 1.00549242610729 0.65572760463222 0.53184945218649
0.51792141093558 0.51775738621207 0.51775736368246 0.51775736368246
0.51775736368246 0.51775736368246 0.51775736368246
```

Dès le sixième terme, on a obtenu 6 chiffres significatifs, et dès le huitième, on a quatorze chiffres significatifs..

4) Pour la méthode du point fixe, on utilise l'équivalence

$$\cos x - xe^x = 0 \iff x = \frac{\cos x}{e^x},$$

pour tout $x \in [0,\ \pi/2]$.

a) On définit donc $\varphi(x) = \cos x/e^x$, et on calcule $\varphi'(x)$:

```
» phi = cos(x)./exp(x);
» phiPrime = diff(phi)
phiPrime = -sin(x)/exp(x)-cos(x)/exp(x)
```

On calcule les valeurs de $\varphi(x)$ et $\varphi'(x)$ pour

$$x = 0, \quad 0.45, \quad 0.6, \quad \pi/2.$$

```
» format short ; double(subs(phi, x, [0 0.45 0.6 pi/2]))
ans = 1.0000 0.5741 0.4529 0
» double(subs(phiPrime, x, [0 0.45 0.6 pi/2]))
ans = -1.0000 -0.8515 -0.7628 -0.2079
```

Comme $\varphi'(0) = -1$, l'hypothèse $(F3)$ n'est pas vérifiée.

b) On calcule φ''.

```
» phiSeconde = diff(phiPrime)
phiSeconde = 2*sin(x)/exp(x)
```

Sur l'intervalle $[0.45,\ 0.6]$, φ' est croissante car

$$\varphi''(x) = \frac{2\sin x}{e^x} > 0,$$

Les tableaux de valeurs précédents montrent que φ' croît de

$$\varphi'(0,45) \simeq -0,8515$$

à

$$\varphi'(0,6) \simeq -0,7628.$$

On a donc

$$M = \max_{x\in[0.45,\ 0.6]} |\varphi'(x)| = |\varphi'(0,45)| \simeq 0,8515$$

et, φ' étant négative φ décroît de 0,5741 à 0,4529. Les hypothèses $(F1)$, $(F2)$ et $(F3)$ sont donc bien vérifiées sur l'intervalle $[0.45,\ 0.6]$.

c) On évalue le nombre n de termes à calculer pour obtenir la précision de 10^{-6}.

```
» M =double(abs(subs(phiPrime, x, 0.45 )))
M = 0.8515
» n = ceil((log(1e-6)-log(0.6-0.45))/log(M))
n = 75
```

Exercice 6.5.2

On vérifie facilement que les hypothèses (N_1) à (N_5) sont satisfaites. Partant d'une initialisation x_0 quelconque, on calcule

$$x_1 = x_0 - \frac{\alpha x_0 + \beta}{\alpha} = -\frac{\beta}{\alpha}.$$

Ainsi, dès la première itération, on a obtenu la valeur exacte de la solution de l'équation

$$\alpha x + \beta = 0.$$

Exercice 6.5.3

Sur l'intervalle $[a,\ b] = [1,\ 2]$, la fonction f définie par $f(x) = x^2 - 2$ est continue et deux fois dérivable (fonction polynôme). Sa dérivée est strictement positive, donc f est strictement croissante sur cet intervalle. De plus $f(1) = -1$ et $f(2) = 3$ sont de signes contraires. Ainsi, toutes les hypothèses sont vérifiées pour appliquer la méthode de Newton. Comme

$$f''(x) = 2 > 0,$$

on choisit l'initialisation x_1 telle que $f(x_1) > 0$, par exemple

$$x_1 = 2.$$

La suite des itérés de Newton est définie par

$$\begin{cases} x_1 = 2 \\ x_{n+1} = x_n - \dfrac{x_n^2 - 2}{2x_n} = \dfrac{x_n^2 + 2}{2x_n}. \end{cases}$$

On calcule avec *Matlab* les six premiers termes de cette suite.

```
» X(1)=2 ;
» for n=1 :5,
   X(n+1)=(X(n)^2+2)/(2*X(n)) ;
end
» format long ;
» X
X =
Columns 1 through 4
2.00000000000000 1.50000000000000 1.41666666666667 1.41421568627451
Columns 5 through 6
1.41421356237469 1.41421356237309
```

On compare avec la valeur de $\sqrt{2}$, affichée au format long :

```
» s=sqrt(2)
» s = 1.41421356237310
```

Dès le sixième terme, on a obtenu 15 chiffres significatifs.

Exercice 6.5.4

1) La suite des itérés de la méthode du point fixe est définie par

$$\begin{cases} s_1 = x_0 \\ s_i = \varphi(s_{i-1}) \qquad i \geqslant 2. \end{cases}$$

Pour calculer $\varphi(s_{i-1})$, on remplace dans l'expression de φ la variable *var* par s_{i-1}.

```
function s=pointFixe(phi,var,x0,n)
s(1)=x0 ;
for i=2 :n
    s(i)=double(subs(phi,var,s(i-1))) ;
end
```

On peut tester sur l'exemple présenté au paragraphe 6.2.4.

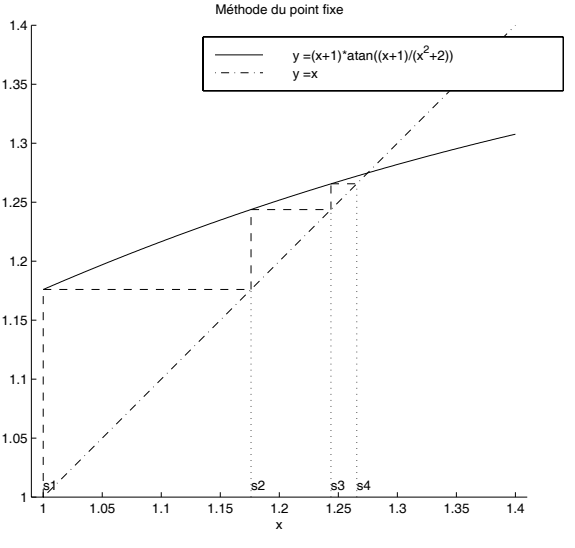
```
» pointFixe('9/10*exp(-x)','x',0,6)
ans =
0 0.9000 0.3659 0.6242 0.4821 0.5557
```

2) Pour l'illustration graphique, après avoir construit la courbe représentative de φ et la droite $y = x$, on trace, pour i de 1 à $n-1$, les segments joignant $A_i \begin{pmatrix} s_i \\ a \end{pmatrix}$

à $M_i \begin{pmatrix} s_i \\ s_i \end{pmatrix}$, puis M_i à $N_i \begin{pmatrix} s_i \\ s_{i+1} \end{pmatrix}$, et enfin N_i à $M_{i+1} \begin{pmatrix} s_{i+1} \\ s_{i+1} \end{pmatrix}$ (voir paragraphe 6.2.3.1). Au point A_i, on place le texte 's_i'.

```
function s=pointFixeGraph(phi,var,x0,n,a,b)
s=pointFixe(phi,var,x0,n);
clf ; hold on ;
ezplot(phi,[a b])    %représentation de phi
axis equal
h=(b-a)/40 ;    % h définit une marge à gauche de a et à droite de b
axis([a-h b+h a b])
plot([a b], [a b],'-.')    %représentation de y=x entre a et b
legend(strcat('y = ',char(phi)),strcat('y = ',char(var)))
for i=1 :n-1
   plot([s(i) s(i)],[a s(i)],' :')
   plot([s(i) s(i)],[s(i) s(i+1)],'–')
   plot([s(i) s(i+1)],[s(i+1) s(i+1)],'–')
   text(s(i),a+h,strcat('s',num2str(i)))    % place le texte 'si'
end
plot([s(n) s(n)],[a s(n)],' :')    % dessine le segment [An Mn]
text(s(n),a+h,strcat('s',num2str(n))) % place le texte sn
title('Méthode du point fixe')
```

Pour la fonction de l'exemple 6.4.1, on obtient le graphe suivant



grâce aux instructions

```
» syms x real
» phi=(x+1)*atan((x+1)/(x^2+2));
» s=pointFixeGraph(phi,x,1,4,1,1.4)
s =  1.0000 1.1760 1.2438 1.2656
```

Exercice 6.5.5

1) Appliquer la méthode de Newton à la fonction f revient à appliquer la méthode du point fixe à la fonction

$$x \mapsto \varphi(x) = x - \frac{f(x)}{f'(x)}.$$

D'où la fonction *Matlab*

```
function s=newton(f,var,x0,n)
    phi = simplify(var - f/diff(f,var));
    s=pointFixe(phi,var,x0,n);
```

Exemple d'utilisation (voir paragraphe 6.3.4.2) :

```
» syms x real
» f=x^3-4.53*x^2+6.0291*x-2.218039;
» s=newton(f,x,1.15,6)
s = 1.1500 1.7309 1.4776 1.5101 1.5100 1.5100
```

2) Pour la fonction *newtonGraphe*, on doit en plus représenter graphiquement f, et construire pour chaque itéré s_i de la méthode de Newton :

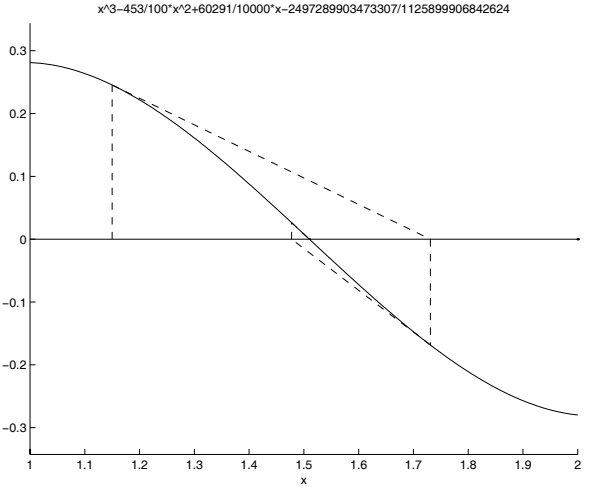
- le segment "vertical" $[N_iM_i]$ joignant le point d'abscisse s_i de l'axe (Ox) au point d'abscisse s_i de la courbe représentative de f ;
- le segment de tangente joignant ce dernier point M_i au point N_{i+1} d'abscisse s_{i+1} de l'axe (Ox).

```
function s=newtonGraph(f,var,x0,n,a,b)
s=newton(f,var,x0,n);
clf ; hold on ;
ezplot(f,[a b])
plot([a b], [0 0],'.-') % dessin axe des abscisses
for i=1 :n-1
   y= double(subs(f,'x',s(i)));    % calcul de y = f(s_i)
   plot([s(i) s(i)],[0 y],'–')    %dessin de [N_iM_i]
   plot([s(i) s(i+1)],[y 0],'–')    %dessin de [M_iN_{i+1}]
end
```

Sur l'exemple précédent :

```
» syms x real
» f=x^3-4.53*x^2+6.0291*x-2.218039;
» s=newtonGraph(f,x,1.15,6,1,2);
```

D'où le graphe



Chapitre 7

Interpolation polynomiale

7.1. Le polynôme d'interpolation d'une fonction

7.1.1. *Définitions*

On suppose connues les valeurs d'une fonction f en un nombre fini de points distincts selon le tableau suivant :

x	x_0	x_1	$\cdots$	x_n
$f(x)$	f_0	f_1	$\cdots$	f_n

Un tel tableau peut être le résultat de mesures effectuées expérimentalement.

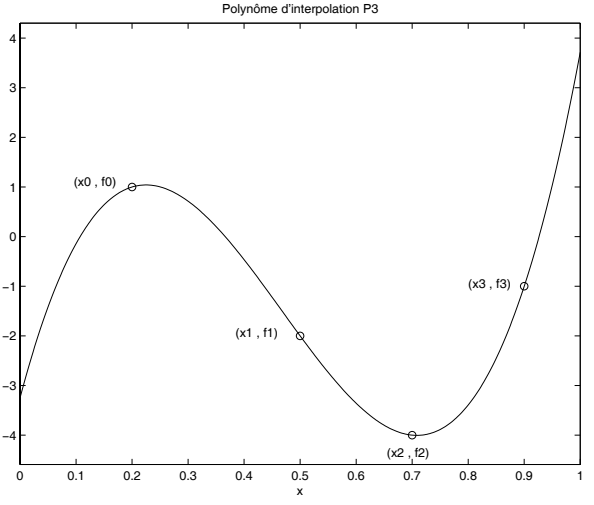
On se propose alors d'approcher f par une fonction simple de type polynomial P_n, de degré inférieur ou égal à n, et telle que

$$P_n(x_i) = f_i \qquad i = 0, ..., n.$$

On montrera que P_n existe et est unique. On l'appelle le polynôme d'interpolation de f aux points $\{x_0, x_1, ..., x_n\}$.

La représentation graphique de cette fonction P_n est une courbe passant par les $n+1$ points de coordonnées

$$\begin{pmatrix} x_0 \\ f_0 \end{pmatrix}, \begin{pmatrix} x_1 \\ f_1 \end{pmatrix}, \cdots, \begin{pmatrix} x_n \\ f_n \end{pmatrix}.$$



Lorsqu'on travaille sur un intervalle $[a,\ b]$, l'interpolation polynomiale :

– permet de donner une approximation numérique $P_n(\alpha)$ de $f(\alpha)$ pour α appartenant à $[a,\ b]$, et $\alpha \neq x_i$, $i = 0, 1, ..., n$. On dira qu'on a **interpolé**,

– sert à construire des formules explicites utiles pour le calcul approché d'intégrales.

7.1.2. *Théorème d'existence et d'unicité de* P_n

On a le résultat :

> *Il existe un unique polynôme P_n de degré inférieur ou égal à n tel que*
> $$P_n(x_i) = f_i \quad i = 0, ..., n.$$

Démontrons ce résultat dans le cas $n = 2$. Le cas général se fait d'une manière similaire. En écrivant

$$P_2(x) = a_0 + a_1 x + a_2 x^2,$$

et

$$P_2(x_i) = f_i \quad i = 0, 1, 2,$$

on obtient le système linéaire

$$\begin{cases} a_0 + a_1 x_0 + a_2 x_0^2 = f_0 \\ a_0 + a_1 x_1 + a_2 x_1^2 = f_1 \\ a_0 + a_1 x_2 + a_2 x_2^2 = f_2, \end{cases}$$

qu'on peut écrire sous forme matricielle

$$\begin{pmatrix} 1 & x_0 & x_0^2 \\ 1 & x_1 & x_1^2 \\ 1 & x_2 & x_2^2 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ a_2 \end{pmatrix} = \begin{pmatrix} f_0 \\ f_1 \\ f_2 \end{pmatrix}.$$

Le déterminant de la matrice de ce système (appelé déterminant de Vandermonde) se calcule sous *Matlab* :

```
» syms x0 x1 x2
» A=[1 x0 x0^2 ; 1 x1 x1^2 ; 1 x2 x2^2] ;
» factor (det(A))
ans= -(x2-x0)*(-x0+x1)*(x1-x2)
```

Ce déterminant est différent de zéro puisque les points x_i sont distincts. D'où l'existence et l'unicité des a_i.

7.1.2.1. *Remarque*

Dans le cas général, on montre que la résolution du système, permettant le calcul des coefficients du polynôme d'interpolation, nécessite un nombre d'opérations en $O(n^3)$. On utilisera plutôt d'autres méthodes, moins coûteuses en nombre d'opérations, dès que n devient grand. Par exemple, l'utilisation des polynômes de Lagrange, présentée ci-dessous, nécessite un nombre d'opérations en $O(n^2)$ (voir exercice 7.3.5).

7.1.3. *Polynôme de Lagrange*

7.1.3.1. *Définition*

On appelle polynôme de Lagrange d'indice $k = 0, ..., n$ associé aux points $\{x_0, x_1, ..., x_n\}$ le polynôme défini par

$$L_k(x) = \frac{(x - x_0)(x - x_1)...(x - x_{k-1})(x - x_{k+1})...(x - x_n)}{(x_k - x_0)(x_k - x_1)...(x_k - x_{k-1})(x_k - x_{k+1})...(x_k - x_n)}.$$

On retiendra que dans le polynôme de Lagrange d'indice k :

– le numérateur est le produit de tous les facteurs $(x - x_i)$, exception faite du facteur $(x - x_k)$,

– le dénominateur est le produit de tous les facteurs $(x_k - x_i)$, pour $i \neq k$.

7.1.3.2. *Exemple*

Dans le cas $n = 2$, traité précédemment, les trois polynômes de Lagrange sont

$$L_0(x) = \frac{(x - x_1)(x - x_2)}{(x_0 - x_1)(x_0 - x_2)}, \quad (k = 0)\,,$$

$$L_1(x) = \frac{(x - x_0)(x - x_2)}{(x_1 - x_0)(x_1 - x_2)}, \quad (k = 1)\,,$$

$$L_2(x) = \frac{(x - x_0)(x - x_1)}{(x_2 - x_0)(x_2 - x_1)}, \quad (k = 2)\,.$$

On va montrer que les polynômes de Lagrange constituent une base pour l'écriture du polynôme d'interpolation de f. En effet on a le résultat :

7.1.3.3. *Théorème*

Le polynôme d'interpolation de f aux points $\{x_0, x_1, ..., x_n\}$ est donné par $P_n(x) = \sum_{k=0}^{n} f_k L_k(x)$.

Cette relation découle facilement du fait que

$$\begin{cases} d^\circ P_n \leqslant n, \\ L_k(x_k) = 1 \ \text{ pour } k = 0, 1, ..., n, \\ L_k(x_j) = 0 \ \ \text{ pour } j \neq k, \end{cases}$$

d'où pour tout $k = 0, 1, ..., n$,

$$P_n(x_k) = f_0.0 + ... + f_{k-1}.0 + f_k.1 + f_{k+1}.0 + ...f_n.0 = f_k.$$

7.1.3.4. *Exemple 1*

Dans le cas de deux points

x	x_0	x_1
$f(x)$	$f_0 = f(x_0)$	$f_1 = f(x_1)$

on a les deux polynômes de Lagrange

$$L_0(x) = \frac{(x - x_1)}{(x_0 - x_1)}, \quad L_1(x) = \frac{(x - x_0)}{(x_1 - x_0)},$$

et le polynôme P_1 est la fonction affine donnée par

$$P_1(x) = \frac{x_1 - x}{x_1 - x_0} f_0 + \frac{x - x_0}{x_1 - x_0} f_1.$$

7.1.3.5. *Exemple 2*

Soit f une fonction connue aux points

$$x_0 = 0,\ x_1 = 2,\ x_2 = 4,$$

de valeurs respectives

$$f_0 = 3,\ f_1 = -1,\ f_2 = 3.$$

Calculons $P_2(x)$ pour en déduire une approximation polynomiale de $f(2,5)$.

On a

$$\begin{aligned} P_2(x) &= \sum_{k=0}^{2} f_k L_k(x) \\ &= 3\frac{(x-x_1)(x-x_2)}{(x_0-x_1)(x_0-x_2)} - 1\frac{(x-x_0)(x-x_2)}{(x_1-x_0)(x_1-x_2)} \\ &\quad +3\frac{(x-x_0)(x-x_1)}{(x_2-x_0)(x_2-x_1)} \\ &= x^2 - 4x + 3, \end{aligned}$$

d'où $P_2(2,5) = 2,5^2 - 4 \times 2,5 + 3 = -0,75$.

7.1.4. *Algorithme d'Aitken*

L'algorithme d'Aitken utilise une formule de récurrence permettant de calculer le polynôme d'interpolation d'une fonction connue en $n+1$ points, à partir de deux polynômes d'interpolation déterminés à partir de n de ces points.

Etant donné le tableau de valeurs d'interpolation

x_0	x_1	$\cdots$	x_n
f_0	f_1	$\cdots$	f_n

on notera par commodité

$$P_n(x) = P_{\{x_0, x_1, \ldots, x_n\}}(x).$$

Dans le cas d'un seul point d'interpolation $\{x_i\}$, on a

$$P_{\{x_i\}}(x) = f_i.$$

On vérifie que, à partir des polynômes d'interpolation

$$P_{\{x_0,x_1,\ldots,x_{n-1}\}}(x),\ P_{\{x_1,\ldots,x_n\}}(x)$$

construits respectivement sur les n points

$$\{x_0,x_1,\ldots,x_{n-1}\},\ \{x_1,\ldots,x_n\},$$

on obtient le polynôme $P_{\{x_0,x_1,\ldots,x_n\}}(x)$ construit sur $n+1$ points par

$$P_{\{x_0,x_1,\ldots,x_n\}}(x)=\frac{(x_n-x)P_{\{x_0,x_1,\ldots,x_{n-1}\}}(x)-(x_0-x)P_{\{x_1,\ldots,x_n\}}(x)}{x_n-x_0}.$$

7.1.4.1. *Exemple*

A partir du tableau de valeurs

x_i	0	2	4
f_i	3	-1	3

calculons $P_2(2,5)$ par l'algorithme d'Aitken.

On a

$$\begin{aligned}P_{\{x_0,x_1,x_2\}}(2,5) &= \frac{(x_2-2,5)P_{\{x_0,x_1\}}(2,5)-(x_0-2,5)P_{\{x_1,x_2\}}(2,5)}{x_2-x_0}\\ &= \frac{(1,5)\,P_{\{x_0,x_1\}}(2,5)+(2,5)P_{\{x_1,x_2\}}(2,5)}{4}\end{aligned}$$

où $P_{\{x_0,x_1\}}$ et $P_{\{x_1,x_2\}}$ sont les polynômes d'interpolation sur $\{x_0,\ x_1\}$ et $\{x_1,\ x_2\}$ respectivement. On réitère ensuite pour obtenir

$$\begin{aligned}P_{\{x_0,x_1\}}(2,5) &= \frac{(x_1-2,5)P_{\{x_0\}}(2,5)-(x_0-2,5)P_{\{x_1\}}(2,5)}{x_1-x_0}\\ &= \frac{-0,5\times 3-(-2,5)\times(-1)}{2}\\ &= -2.\end{aligned}$$

On obtient de même

$$P_{\{x_1,x_2\}}(2,5)=0.$$

D'où

$$P_{\{x_0,x_1,x_2\}}(2,5) = \frac{(1,5)\times(-2)}{4} = -0,75.$$

On peut disposer les résultats dans le tableau

$P_{\{x_0\}} = f_0 = 3$		
$P_{\{x_1\}} = f_1 = -1$	$P_{\{x_0,x_1\}}(2,5) = -2$	
$P_{\{x_2\}} = f_2 = 3$	$P_{\{x_1,x_2\}}(2,5) = 0$	$P_{\{x_0,x_1,x_2\}}(2,5) = -0,75$

Ces résultats s'obtiennent sous *Matlab* de la manière suivante

```
» x0=0 ; x1=2 ; x2=4 ;
» f0=3 ; f1=-1 ; f2=3 ;
» Px0=f0 ; Px1=f1 ; Px2=f2 ;
» x=2.5 ;
» Px0x1=((x1-x)*Px0-(x0-x)*Px1)/(x1-x0)
Px0x1=      -2
» Px1x2=((x2-x)*Px1-(x1-x)*Px2)/(x2-x1)
Px1x2=      0
» Px0x1x2=((x2-x)*Px0x1-(x0-x)*Px1x2)/(x2-x0)
Px0x1x2=      -0.75
```

7.1.5. *Gestion d'erreur*

7.1.5.1. *Résultat préliminaire*

On va estimer l'erreur mathématique

$$|f(x) - P_n(x)|,$$

dans le cas où la fonction est supposée $n+1$ fois dérivable sur l'intervalle de travail $[a,b]$ et où sa dérivée d'ordre $(n+1)$ est bornée.

Posons, pour $x_0, x_1, ..., x_n$ et x fixés

$$\begin{aligned} w(t) &= f(t) - P_n(t) - \\ &\quad (t-x_0)(t-x_1)...(t-x_n)\frac{f(x)-P_n(x)}{(x-x_0)(x-x_1)...(x-x_n)}. \end{aligned}$$

Alors la dérivée d'ordre $(n+1)$ de w est

$$w^{(n+1)}(t) = f^{(n+1)}(t) - (n+1)!\frac{f(x)-P_n(x)}{(x-x_0)(x-x_1)...(x-x_n)},$$

d'autre part on a

$$w(x) = w(x_0) = w(x_1) = ... = w(x_n) = 0,$$

et donc, par application du théorème de Rolle à la fonction w dans les intervalles

$$[x_0, x], [x_0, x_1], ..., [x_{n-1}, x_n],$$

il existe des points $(c'_i)_{i=0,1,...,n}$, tels que

$$w'(c'_i) = 0.$$

On réitère le théorème de Rolle pour la fonction w', puisque

$$w'(c'_0) = w'(c'_1) = w'(c'_2) = ... = w'(c'_n) = 0,$$

pour avoir cette fois n points $(c''_i)_{i=0,1,...,n-1}$ tels que

$$w''(c''_0) = w''(c''_1) = w''(c''_2) = ... = w''(c''_{n-1}) = 0,$$

puis, de proche en proche, il existe un point $c_x \in]a, b[$ tel que

$$w^{(n+1)}(c_x) = 0.$$

On en déduit que

$$f(x) - P_n(x) = \frac{1}{(n+1)!}(x - x_0)(x - x_1)...(x - x_n)f^{(n+1)}(c_x).$$

D'où le théorème suivant :

7.1.5.2. *Théorème*

Soit $f : [a, b] \to \mathbb{R}$ une fonction $(n + 1)$ fois dérivable, telle que $\left|f^{(n+1)}\right|$ soit bornée par une constante $M > 0$. Si P_n désigne le polynôme d'interpolation de f sur la subdivision $\{x_0, x_1, ..., x_n\}$ de $[a, b]$, alors on a pour tout $x \in [a, b]$,

$$|f(x) - P_n(x)| \leqslant \frac{M}{(n+1)!} \left|(x - x_0)(x - x_1)...(x - x_n)\right|.$$

7.1.5.3. *Exemple*

En utilisant cette dernière inégalité, dans le cas où $f(x) = \ln(x)$, on peut donner une majoration de l'erreur commise en calculant une valeur approchée de $\ln(529, 62)$ par une interpolation utilisant les valeurs de $\ln(529)$ et $\ln(530)$:

$$|\ln(529, 62) - P_1(529, 62)| \leqslant \frac{M}{(1+1)!} \left|(529, 62 - 529, 62)(529, 62 - 530)\right|,$$

avec

$$M = \max_{x\in[529,\,530]} |f''(x)| = \max_{x\in[529,\,530]} \left|-\frac{1}{x^2}\right| = \left(\frac{1}{529}\right)^2.$$

On obtient

$$|\ln(529,62) - P_1(529,62)| \leqslant \frac{1}{2\times 529^2}\,(0,62\times 0,48) \simeq 5,32.10^{-7}.$$

A titre de vérification, on calcule

$$\begin{aligned} P_1(529,62) &= \frac{529,62-530}{529-530}\ln(529) + \frac{529,62-529}{530-529}\ln(530) \\ &\simeq 6,27215934816478 \end{aligned}$$

puis

$$\ln(529,62) \simeq 6.27215976826020.$$

L'erreur observée, liée à cette méthode d'interpolation est donc

$$|P_1(529,62) - \ln(529,62)| \simeq 4,20.10^{-7}.$$

7.2. Approche polynomiale de la dérivation

7.2.1. *Approche classique*

On se donne une fonction $f : \mathbb{R} \to \mathbb{R}$ et un point $a \in \mathbb{R}$. On sait alors que, lorsque f est dérivable au point a, on a

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

Cela suggère que pour h assez petit, on peut approcher $f'(a)$ par le rapport

$$\frac{f(a+h)-f(a)}{h}.$$

On peut aussi penser à utiliser d'autres quotients différentiels tels que

$$Q_a(h) = \frac{f(a+h)-f(a-h)}{2h}.$$

On pourra montrer que, si f est dérivable en a,

$$\lim_{h\to 0} Q_a(h) = f'(a).$$

7.2.2. *Approche polynomiale*

Dans le cas d'une fonction f connue en $(n+1)$ points équidistants $\{x_0, x_1, ..., x_n\}$ de $[a, b]$, avec

$$x_i = x_0 + ih, \quad i = 0, 1, ..., n$$

où h est le pas, on peut approcher $f'(x_i)$ par $P'_n(x_i)$.

7.2.2.1. *Exemple dans le cas de deux points*

Partant du tableau :

x_0	$x_1 = x_0 + h$
$f_0 = f(x_0)$	$f_1 = f(x_1)$

le polynôme P_1 est

$$P_1(x) = \frac{x_1 - x}{x_1 - x_0} f_0 + \frac{x - x_0}{x_1 - x_0} f_1,$$

d'où

$$\begin{aligned} P'_1(x) &= \frac{f_0}{x_0 - x_1} + \frac{f_1}{x_1 - x_0} \\ &= \frac{f_1 - f_0}{h} \\ &= \frac{f(x_0 + h) - f(x_0)}{h} \\ &= P'_1(x_0) = P'_1(x_1). \end{aligned}$$

On retrouve les résultats de la première approche.

7.2.2.2. *Exemple dans le cas de trois points*

Partant du tableau

$x_{-1} = x_0 - h$	x_0	$x_1 = x_0 + h$
$f_{-1} = f(x_0 - h)$	$f_0(x_0)$	$f_1 = f(x_0 + h)$

et, après calcul de $P_2(x)$, on trouve

$$\begin{aligned} P'_2(x) &= \frac{f_{-1}}{(x_{-1} - x_0)(x_{-1} - x_1)}[2x - (x_0 + x_1)] \\ &\quad + \frac{f_0}{(x_0 - x_{-1})(x_0 - x_1)}[2x - (x_{-1} + x_1)] \\ &\quad + \frac{f_1}{(x_1 - x_{-1})(x_1 - x_0)}[2x - (x_0 + x_{-1})], \end{aligned}$$

d'où

$$P_2'(x_{-1}) = \frac{3f_{-1} - 4f_0 + f_1}{2h}$$
$$P_2'(x_0) = \frac{f_1 - f_{-1}}{2h}$$
$$P_2'(x_1) = \frac{f_{-1} - 4f_0 + 3f_1}{2h},$$

valeurs qui approchent respectivement $f'(x_{-1})$, $f'(x_0)$ et $f'(x_1)$.

On remarquera que, pour $P_2'(x_0)$, on retrouve le quotient différentiel $Q_{x_0}(h)$ donné à la fin du paragraphe 7.2.1.

7.2.3. *Gestion d'erreur mathématique*

Par dérivation de la formule

$$f(x) - P_n(x) = \frac{1}{(n+1)!}(x - x_0)(x - x_1)...(x - x_n)f^{(n+1)}(c_x),$$

vue au paragraphe 7.1.5.1, on déduit que pour une fonction f indéfiniment dérivable, admettant une dérivée d'ordre $(n+1)$ bornée, et pour une subdivision de points équidistants

$$(x_i = x_0 + ih)_{i=0,1,...,n}$$

on a la majoration d'erreur

$$|f'(x_i) - P_n'(x_i)| \leqslant \frac{M}{(n+1)!} \prod_{j \neq i,} |x_i - x_j|.$$

A titre illustratif, dans le cas de trois points équidistants, on a, au point x_0

$$|f'(x_0) - P_2'(x_0)| \leqslant \frac{M}{6} |(x_0 - x_{-1})(x_0 - x_1| \leqslant \frac{Mh^2}{6},$$

où

$$M = \max_{x \in [a,b]} |f'''(x)|.$$

7.2.4. *Etude complète d'erreur*

On va montrer, que le cumul de l'erreur mathématique et des erreurs d'arrondi fait qu'on ne peut pas prendre le pas h trop petit si on veut optimiser l'erreur totale commise lorsqu'on confond $f'(x_0)$ avec $P_2'(x_0)$.

7.2.4.1. *Première partie*

Etant donné le tableau d'interpolation basé sur les trois points équidistants

$x_{-1} = x_0 - h$	x_0	$x_1 = x_0 + h$
$y_{-1} = f(x_0 - h)$	$y_0 = f(x_0)$	$y_1 = f(x_0 + h)$

on retrouve, grâce au calcul symbolique sous *Matlab*, les approximations polynomiales des dérivées de f aux points d'interpolation.

– On calcule d'abord les trois polynômes de Lagrange.

```
» syms x x0 h fMoins1 f0 f1
» xMoins1=x0-h;
» x1=x0+h;
» LMoins1=(x-x0)*(x-x1)/((xMoins1-x0)*(xMoins1-x1))
LMoins1= 1/2*(x-x0)*(x-x0-h)/h^2
» L0=(x-xMoins1)*(x-x1)/((x0-xMoins1)*(x0-x1))
L0= -(x-x0+h)*(x-x0-h)/h^2
» L1=(x-xMoins1)*(x-x0)/((x1-xMoins1)*(x1-x0))
L1= 1/2*(x-x0+h)*(x-x0)/h^2
```

– On calcule ensuite $P_2(x)$, $P_2'(x)$ et $P_2'(x_0)$.

```
» P=fMoins1*LMoins1+f0*L0+f1*L1;
» Pprime=diff(P,x);
» Pprime=simple(Pprime);
» PprimeX0=simple(subs(Pprime,x,x0))
PprimeX0= 1/2/h*(-fMoins1+f1)
```

On retrouve ainsi $P_2'(x_0) = \dfrac{f_1 - f_{-1}}{2h}$.

7.2.4.2. *Deuxième partie*

On analyse l'erreur mathématique liée à l'interpolation et la dérivation numérique.

On utilise la formule

$$f(x) - P_2(x) = \frac{1}{6}(x - x_{-1})(x - x_0)(x - x_1)f'''(c_x).$$

Comme on ne connaît pas explicitement c_x, on déclare la variable symbolique *fTierceCx* et sa dérivée *fTierceCxprime* (on admet l'existence de cette dernière).

```
» syms fTierceCx fTierceCxprime
» FmoinsP=(x-xMoins1)*(x-x0)*(x-x1)/6*fTierceCx
FmoinsP= 1/6*(x-x0+h)*(x-x0)*(x-x0-h)*fTierceCx
```

Dans le calcul de $(f(x) - P_2(x))'$, on doit tenir compte de la dérivée de $f'''(c_x)$.

```
» FmoinsPprime=diff(FmoinsP,x) +...
      fTierceCxprime*(x-xMoins1)*(x-x0)*(x-x1);
» FmoinsPprimeX0=simple(subs(FmoinsPprime,x,x0))
FmoinsPprimeX0= -1/6*h^2*fTierceCx
```

On retrouve ainsi l'estimation de l'erreur mathématique

$$|f'(x_0) - P_2'(x_0)| = \left| -\frac{f'''(c_{x_0})h^2}{6} \right| \leqslant \frac{Mh^2}{6}.$$

7.2.4.3. *Troisième partie*

Sachant que sur les données f_{-1}, f_0, f_1, on commet une erreur de mesure (ou d'arrondi) d'au plus ε, cherchons le pas h optimum pour que l'erreur globale commise sur $f'(x_0)$ en la confondant avec $P_2'(x_0)$ soit minimale.

On rappelle que

$$P_2'(x_0) = \frac{f_1 - f_{-1}}{2h}.$$

La valeur calculée de $P_2'(x_0)$ (due aux erreurs sur les données) est

$$P_{2app}'(x_0) = \frac{f_{1app} - f_{-1app}}{2h}.$$

L'erreur de mesure est estimée par

$$\begin{aligned} \left|P_{2app}'(x_0) - P_2'(x_0)\right| &= \left| \frac{f_{1app} - f_{-1app}}{2h} - \frac{f_1 - f_{-1}}{2h} \right| \\ &= \left| \frac{f_{1app} - f_1}{2h} - \frac{f_{-1} - f_{-1app}}{2h} \right| \\ &\leqslant \frac{\varepsilon}{2h} + \frac{\varepsilon}{2h} = \frac{\varepsilon}{h}. \end{aligned}$$

On en déduit l'estimation de l'erreur totale

$$\begin{aligned} \left|f'(x_0) - P_{2app}'(x_0)\right| &\leqslant |f'(x_0) - P_2'(x_0)| + \left|P_2'(x_0) - P_{2app}'(x_0)\right| \\ &\leqslant \frac{Mh^2}{6} + \frac{\varepsilon}{h}. \end{aligned}$$

Il suffit de chercher les valeurs de h réelles positives pour lesquelles la fonction

$$f(h) = \frac{Mh^2}{6} + \frac{\varepsilon}{h}$$

est minimum.

```
» syms epsi M
» f=M*h^2/6+epsi/h;
» fPrime=diff(f,h)
fPrime =  1/3*M*h-epsi/h^2
» S =solve(fPrime,h);
» h0 = S(1) %seule la première solution est réelle
h0 = 1/M*3^(1/3)*(epsi*M^2)^(1/3)
```

Pour vérifier qu'au point

$$h_0 = \frac{1}{M}\sqrt[3]{3\varepsilon M^2} = \sqrt[3]{\frac{3\varepsilon}{M}},$$

la fonction f présente un minimum, on écrit $f'(h)$ sous forme de quotient N/D.

```
» [N,D]=numden(fPrime)
N =M*h^3-3*epsi
D =3*h^2
```

Le dénominateur $D = 3h^2$ est strictement positif, et le numérateur N est positif si

$$h^3 > \frac{3\varepsilon}{M},$$

soit

$$h > h_0.$$

L'étude du signe de la dérivée montre que f est décroissante sur $]0,\ h_0]$, et croissante sur $[h_0,\ +\infty[$.

7.2.4.4. *Quatrième partie*

Illustrons les calculs précédents numériquement sur la fonction

$$f : x \longmapsto f(x) = x^5/20,$$

au voisinage de $x_0 = 1$.

L'erreur d'arrondi est estimée par

$$\varepsilon = 2^{-52}.$$

On peut penser a priori que le pas h optimum pour approcher $f'(x_0)$ par

$$P'_{2app}(x_0) = \frac{f_{1app} - f_{-1app}}{2h}$$

est proche de $\varepsilon = 2^{-52} \simeq 10^{-16}$.

Le calcul précédent montre qu' il n'en est rien, puisqu'on a ici $f'''(x) = 3x^2$, et au voisinage de 1, on peut assimiler M à 3. D'où

```
» h0=double(subs(h0,{M,epsi},{3,2^(-52)}))
h0 = 6.0555e-006
```

Le pas optimum est donc $h_0 \simeq 6.10^{-6}$.

Examinons de plus près l'erreur observée

$$\left| f'(x_0) - P'_{2app}(x_0) \right|,$$

lorsque h prend les valeurs successives :

$$10^{-1}, 10^{-2}, ..., 10^{-14}.$$

– On calcule $f'(x_0)$.

```
» syms x
» f= x^5/20 ;
» x0 = 1 ;
» f0 = double(subs(f,x,x0))
f0 = 0.0500
» derExact = double(subs(diff(f),x,x0))
derExact = 0.2500
```

– On calcule pour $h = 10^{-1}, 10^{-2}, ..., 10^{-14}$ (première colonne du tableau $derApp$) les valeurs correspondantes de $P'_{2app}(x_0)$ (deuxième colonne) et enfin l'erreur observée $\left| f'(x_0) - P'_{2app}(x_0) \right|$ (en troisième colonne).

```
» for n = 1 :14
    h=10^(-n) ;
    derApp(n,1) = h ;
    fMoins1= double(subs(f,x,x0-h)) ;
    f1 = double(subs(f,x,x0+h)) ;
    derApp(n,2) = (f1-fMoins1)/(2*h) ;
end
» derApp( :,3)=abs(derExact-derApp( :,2)) ;
```

```
» format long
» derApp
derApp =
0.10000000000000   0.25500500000000   0.00500500000000
0.01000000000000   0.25005000050000   0.00005000050000
0.00100000000000   0.25000050000005   0.00000050000005
0.00010000000000   0.25000000500000   0.00000000500000
0.00001000000000   0.25000000005125   0.00000000005125
0.00000100000000   0.24999999999331   0.00000000000669
0.00000010000000   0.24999999997249   0.00000000002751
0.00000001000000   0.24999999952147   0.00000000047853
0.00000000100000   0.25000000680730   0.00000000680730
0.00000000010000   0.24999998599062   0.00000001400938
0.00000000001000   0.25000002068509   0.00000002068509
0.00000000000100   0.25000834735778   0.00000834735778
0.00000000000010   0.24993895841874   0.00006104158126
0.00000000000001   0.24945323584546   0.00054676415454
```

On constate que, au-delà de la sixième ligne de ce tableau (correspondant à $h = 10^{-6}$), l'erreur observée augmente. Pour $h = 10^{-14}$, l'erreur est supérieure à celle observée pour 10^{-2} !

7.3. Exercices

7.3.1. *Calcul d'un polynôme d'interpolation*

Soit f une fonction connue selon le tableau suivant

$x_0 = 0$	$x_1 = \pi/6$	$x_2 = \pi/4$
$f_0 = 0$	$f_1 = 0.5$	$f_2 = \sqrt{2}/2$

1) Donner les trois polynômes de Lagrange $L0$, $L1$, $L2$.

2) Donner le polynôme d'interpolation $P2$ en expression symbolique polynomiale et calculer une approximation de $f(\pi/5)$.

3) En supposant que $f(x) = \sin(x)$, donner l'erreur absolue commise en confondant $f(\pi/5)$ avec $P_2(\pi/5)$ et justifier la précision obtenue.

(solution p. 237)

7.3.2. *Polynôme de Lagrange et programmation*

1) On suppose que les points $(x_0, x_1, ..., x_n)$ sont donnés dans un tableau

$$X = [X(1), X(2), ..., X(n+1)],$$

et que les valeurs $(f_0, f_1, ..., f_n)$ sont données dans un second tableau

$$Y = [Y(1), Y(2), ..., Y(n+1)].$$

Ecrire la fonction

$$Px = interpolLagrange(x, X, Y)$$

qui calcule la valeur du polynôme d'interpolation au point x, pour X et Y donnés. On pourra utiliser une fonction auxiliaire

$$Lkx = Lagrange(k, x, X)$$

qui calcule $L_k(x)$, pour chaque polynôme de Lagrange L_k.

2) Utiliser cette fonction $interpolLagrange$ pour retrouver le polynôme P_1 de l'exemple 7.1.3.4.

3) Utilser cette fonction $interpolLagrange$ pour calculer $P_2\ (\pi/5)$, où P_2 est le polynôme d'interpolation associé aux valeurs données dans l'exercice 7.3.1

x_i	0	$\pi/6$	$\pi/4$
f_i	$\sin 0$	$\sin\,(\pi/6)$	$\sin\,(\pi/4)$

(solution p. 237)

7.3.3. *Effet de Runge*

On considère la fonction f définie sur $[-5,\ 5]$ par

$$f(x) = \frac{1}{1+x^2}.$$

Pour n entier donné, on note $P_n(x)$ le polynôme d'interpolation de Lagrange associé aux valeurs

$$(x_0, x_1, ..., x_n) = (-5, -5+2/n, -5+4/n, ..., 5)$$

et

$$(f_0, f_1, ..., f_n) = (f\,(x_0)\,, f\,(x_1)\,, ..., f\,(x_n))\,.$$

Pour comparer f et P_n, on représente ces deux fonctions sur un même graphique. On effectuera cette comparaison successivement pour

$$n = 5,\ n = 10,\ n = 20$$

(on pourra utiliser la fonction $interpolLagrange$ construite à l'exercice 7.3.2).

(solution p. 239)

7.3.4. *Méthode d'Aitken et programmation*

Ecrire la fonction récursive

$$Px = aitken(x, X, Y)$$

qui calcule, par la méthode d'Aitken, la valeur du polynôme d'interpolation au point x, pour

$$X = [X(1), X(2), ..., X(n+1)]$$

et

$$Y = [Y(1), Y(2), ..., Y(n+1)]$$

donnés (on reprend les notations de l'exercice 7.3.2).

(solution p. 241)

7.3.5. *Complexité de calcul de polynôme d'interpolation*

On s'intéresse au nombre d'opérations élémentaires (additions, soustractions, multiplications, divisions) nécessaires au calcul de

$$P_n(x) = \sum_{i=0}^{n} f_i L_i(x),$$

pour x, $\{x_0, x_1, ..., x_n\}$, $\{f_0, f_1, ..., f_n\}$ donnés.

1) Donner le nombre d'opérations effectuées pour calculer chaque polynôme de Lagrange

$$L_k(x) = \frac{(x-x_0)(x-x_1)...(x-x_{k-1})(x-x_{k+1})...(x-x_n)}{(x_k-x_0)(x_k-x_1)...(x_k-x_{k-1})(x_k-x_{k-1})...(x_k-x_n)}.$$

2) En déduire le nombre total d'opérations pour calculer $P_n(x)$.

(On pourra analyser les fonctions $Lagrange$ et $interpolLagrange$ de l'exercice 7.3.2).

(solution p. 242)

7.3.6. *Formule barycentrique de Lagrange*

Le but de cet exercice est d'obtenir une expression permettant d'évaluer le polynôme d'interpolation en effectuant moins de calculs qu'avec les polynômes de Lagrange :

$$L_k(x) = \frac{(x-x_0)(x-x_1)...(x-x_{k-1})(x-x_{k+1})...(x-x_n)}{(x_k-x_0)(x_k-x_1)...(x_k-x_{k-1})(x_k-x_{k+1})...(x_k-x_n)}.$$

1) Obtention de la formule barycentrique de Lagrange (les relations ci-dessous peuvent s'obtenir par des raisonnements et calculs simples ; on pourra aussi les vérifier avec *Matlab* pour une valeur raisonnable de n, par exemple $n = 3$).

a) Montrer que

$$\sum_{k=0}^{n} L_k(x) = 1$$

(considérer le polynôme identiquement égal à 1 et son polynôme d'interpolation associé aux points$\{x_0, x_1, ..., x_n\}$).

b) Pour $\{f_0, f_1, ..., f_n\}$ quelconques, on note

$$P_n(x) = \sum_{i=0}^{n} f_i L_i(x),$$

le polynôme d'interpolation associé. On définit

$$Q(x) = (x-x_0)(x-x_1)...(x-x_n) = \prod_{i=0}^{n} (x-x_i),$$

et, pour tout $k \in \{0, ...n\}$,

$$a_k = \prod_{i \neq k} (x_k - x_i).$$

Vérifier que, pour $x \neq x_k$,

$$L_k(x) = \frac{Q(x)}{(x-x_k)a_k}.$$

c) Déduire des deux questions précédentes, que, pour $x \notin \{x_0, x_1, ..., x_n\}$,

$$Q(x) = \frac{1}{\sum_{k=0}^{n} \frac{1}{(x-x_k)a_k}},$$

et enfin

$$P_n(x) = \frac{\sum_{k=0}^{n} \frac{f_k}{(x-x_k)a_k}}{\sum_{k=0}^{n} \frac{1}{(x-x_k)a_k}}$$

(formule barycentrique de Lagrange).

2) a) Ecrire une fonction *Matlab*

$$A = coeff(X)$$

qui calcule $A = [a_0, a_1, ..., a_n]$ en fonction de $X = [x_0, x_1, ..., x_n]$.

b) Ecrire une fonction *Matlab*

$$Pnx = lagrangeBar(x, X, Y, A)$$

qui calcule $P_n(x)$ en utilisant le plus judicieusement possible la formule barycentrique de Lagrange(on note $Y = [f_0, f_1, ..., f_n]$).

3) Déterminer le nombre d'opérations élémentaires correspondant au calcul de $coeff(X)$, puis au calcul de $lagrangeBar(x, X, Y, A)$. On notera que le calcul de A est à faire une seule fois (pour X donné), et que seul le calcul de Pnx est à refaire pour chaque nouvelle valeur de x.

(solution p. 242)

7.3.7. *Complexité de calcul par la méthode d'Aitken*

1) Donner le nombre d'opérations élémentaires nécessaires pour calculer Px, valeur du polynôme d'interpolation au point x, pour

$$X = [X(1), X(2), ..., X(n+1)], Y = [Y(1), Y(2), ..., Y(n+1)]$$

donnés, par la version récursive de l'algorithme d'Aitken (voir exercice 7.3.4).

2) Ecrire une version itérative de cet algorithme

$$T = aitkenIte(x, X, Y)$$

construisant un tableau triangulaire à deux dimensions T qui sera une généralisation de celui construit dans l'exemple 7.1.4.1 :

- dans la première colonne, on aura, pour $i = 1, ..., n+1$

$$T(i, 1) = P_{\{X(i)\}} = Y(i),$$

- dans la deuxième colonne, on aura, pour $i = 2, ..., n+1$

$$T(i, 2) = P_{\{X(i-1)X(i)\}},$$

- dans la $j^{ième}$ colonne, on aura, pour $i = j, ..., n+1$

$$T(i, j) = P_{\{X(i-j+1)...X(i)\}}.$$

3) Donner le nombre d'opérations nécessaires pour obtenir ce tableau T.

(solution p. 246)

7.4. Solutions

Exercice 7.3.1

1) Pour calculer le polynôme P_2, on déclare les tableaux de valeurs symboliques X et F, puis on calcule les polynômes de Lagrange $L_1(x)$, $L_2(x)$, $L_3(x)$. (Attention au décalage des indices).

```
» X=sym([0 pi/6 pi/4]) ;
» F=sym([0 1/2 sqrt(2)/2]) ;
» syms x
» L1=(x-X(2))*(x-X(3))/((X(1)-X(2))*(X(1)-X(3)))
L1= 24*(x-1/6*pi)*(x-1/4*pi)/pi^2
» L2=(x-X(1))*(x-X(3))/((X(2)-X(1))*(X(2)-X(3)))
L2= -72*x*(x-1/4*pi)/pi^2
» L3=(x-X(1))*(x-X(2))/((X(3)-X(1))*(X(3)-X(2)))
L3= 48*x*(x-1/6*pi)/pi^2
```

2) On peut alors calculer $P_2(x)$, puis $P_2\left(\pi/5\right)$.

```
» P2=F(1)*L1+F(2)*L2+F(3)*L3
P2=
-36*x*(x-1/4*pi)/pi^2+24*2^(1/2)*x*(x-1/6*pi)/pi^2
» P2DepiSur5=subs(P2,sym('pi/5'))
P2DepiSur5 = 9/25+4/25*2^(1/2)
» P2DepiSur5App=double(P2DepiSur5)
P2DepiSur5App= 0.5863
```

3) On calcule également $f\left(\pi/5\right) = \sin\left(\pi/5\right)$ et l'erreur absolue commise en remplaçant $f\left(\pi/5\right)$ par $P_2\left(\pi/5\right)$.

```
» sin(pi/5)
ans= 0.5878
» err= abs( P2DepiSur5App-sin(pi/5))
err = 0.0015
```

Le fait que l'erreur obtenue est petite s'explique par le fait que le graphe de la fonction *sinus* sur l'intervalle $[0,\ \pi/4]$ est proche d'une parabole.

Exercice 7.3.2

1) On définit une fonction *Lagrange* qui calcule de manière itérative

$$L_k(x) = \frac{\prod\limits_{i=1\ldots n+1,\ i\neq k} (x - X(i))}{\prod\limits_{i=1\ldots n+1,\ i\neq k} (X(k) - X(i))}.$$

La valeur de n est donnée par le nombre d'éléments de X, diminué de 1.

```
function Lkx=Lagrange(k,x,X)
n=length(X)-1 ;
Lkx=1 ;
for i=1 :k-1
   Lkx=Lkx*(x-X(i))/(X(k)-X(i)) ;
end
for i=k+1 :n+1
   Lkx=Lkx*(x-X(i))/(X(k)-X(i)) ;
end
```

On utilise ensuite la relation

$$P_n(x) = \sum_{k=1}^{n+1} Y(k).L_k(x)$$

```
function Px=interpolLagrange(x,X,Y)
n=length(X)-1 ;
Px=0 ;
for k=1 :n+1
   Px=Px+Y(k)*Lagrange(k,x,X) ;
end
```

2) On applique la fonction précédente aux tableaux de valeurs symboliques

$$X = [x_0, x_1], Y = [f_0, f_1]$$

```
» syms x0 x1 f0 f1 x ;
» X=[x0 x1] ; Y=[f0 f1] ;
» syms x
» Px=interpolLagrange(x,X,Y)
Px = f0*(x-x1)/(x0-x1)+f1*(x-x0)/(x1-x0)
```

On retrouve l'expression

$$P_1(x) = \frac{x_1 - x}{x_1 - x_0} f_0 + \frac{x - x_0}{x_1 - x_0} f_1$$

de l'exemple 7.1.3.4.

3) On définit les tableaux X et Y, et on calcule $P_2\left(\pi/5\right)$.

```
» X=[0 pi/6 pi/4] ;
» Y=sin(X) ;
» PdePiSur5= interpolLagrange(pi/5,X,Y)
PdePiSur5 = 0.5863
```

Exercice 7.3.3

On définit la fonction f et on construit sa représentation graphique sur l'intervalle

$$[a,\ b] = [-5,\ 5].$$

```
» syms x real
» f=1/(1+x^2) ;
» a=-5 ;b=5 ;
» ezplot(f,a,b)
» hold on ;grid on
```

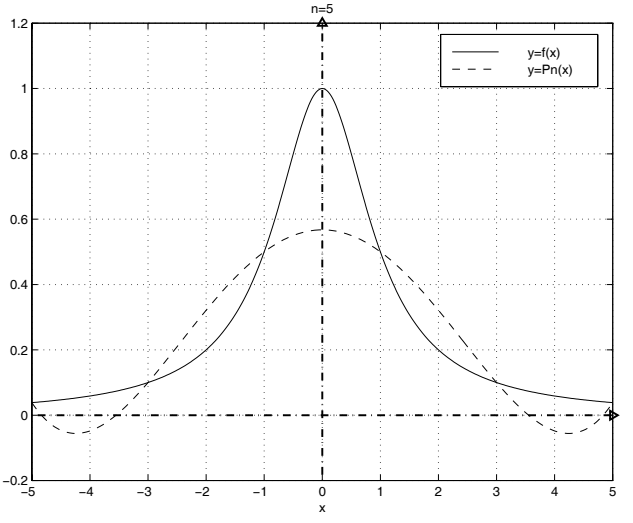
Pour $n = 5$, on définit les tableaux $X = [x_0, x_1, ...x_n]$ et

$$F = [f(x_0), f(x_1), ...f(x_n)]\,.$$

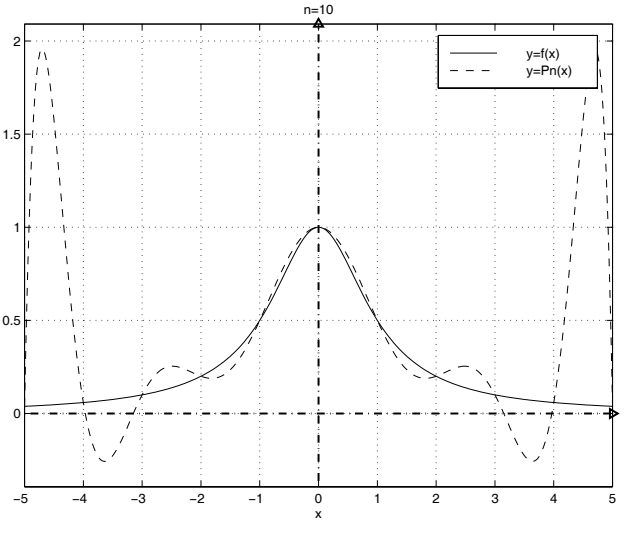
```
» n=5
» X=[a :(b-a)/n :b]
X = -5 -3 -1 1 3 5
» F=1./(1+X.^2)
F = 0.0385 0.1000 0.5000 0.5000 0.1000 0.0385
```

On définit ensuite le polynôme d'interpolation P_n, et on le représente graphiquement

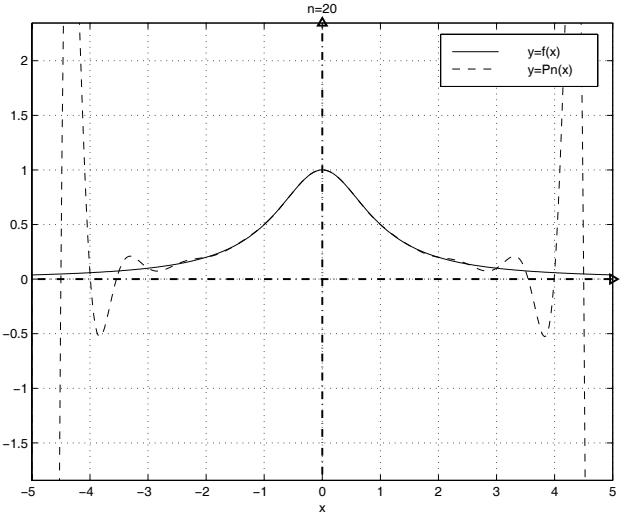
```
» Pn=interpolLagrange(x,X,F);
» set(gca,'LineStyle','- -')
» ezplot(Pn,a,b)
» legend ('y=f(x)','y=Pn(x)')
» axis auto
» title('n=5')
» dessineRepere
```



On effectue le même traitement pour $n = 10$.



On constate que la courbe représentative de P_{10} est plus proche de celle de f, sauf aux extrémités. Ce phénomène est encore accentué pour $n = 20$:



on l'appelle effet de Runge.

Exercice 7.3.4

On utilise les relations :

– pour $n \geqslant 1$:

$$P_{\{X(1),X(2),\ldots,X(n+1)\}}(x) = \frac{(X(n+1)-x)P_{\{X(1),X(2),\ldots,X(n)\}}(x) - (X(1)-x)P_{\{X(2),\ldots,X(n+1)\}}(x)}{X(n+1)-X(1)}$$

– pour $n = 0$:

$$P_{\{X(1)\}}(x) = Y(1).$$

D'où :

```
function y = aitkenRec(x,X,Y)
n=length(X)-1 ;    % Le tableau est [X(1), ...,X(n+1)]
if n==0
   y=Y(1) ;
else
   Q=aitkenRec(x,X(2 :n+1),Y(2 :n+1)) ;
   R=aitkenRec(x,X(1 :n),Y(1 :n)) ;
   y=((X(n+1)-x)*R-(X(1)-x)*Q)/(X(n+1)-X(1)) ;
end
```

On peut tester l'utilisation de cette fonction sur l'exemple 7.1.4.1.

```
» X=[0 2 4] ;
» Y=[3 -1 3] ;
» aitkenRec(2.5,X,Y)
ans = -0.7500
```

Exercice 7.3.5

On reprend les notations de l'exercice 7.3.2.

1) Pour calculer $L_k(x)$, on doit répéter n fois l'instruction

$$L_k \longleftarrow L_k \times (x - X(i))/(X(k) - X(i)),$$

soit $n \times 4$ opérations élémentaires.

2) Pour calculer $P_n(x)$, on doit répéter $n + 1$ fois l'instruction

$$Px \longleftarrow Px + Y(k) \times Lagrange(k, x, X).$$

En tenant compte de la première question, cela fait $4n + 2$ opérations élémentaires à répéter $n + 1$ fois. Soit au total

$$(n + 1)(4n + 2) = 4n^2 + 6n + 2.$$

Remarque

On a vérifié que le nombre d'opérations nécessaires pour calculer $P_n(x)$ est en $O(n^2)$ (voir remarque 7.1.2.1).

Exercice 7.3.6

1) a) Le polynôme constant égal à 1 peut être considéré comme le polynôme d'interpolation associé aux valeurs $\{x_0, x_1, ..., x_n\}$ et

$$\{f_0, f_1, ..., f_n\} = \{1, 1, ..., 1\}.$$

En appliquant la formule de Lagrange, on a donc

$$1 = \sum_{k=0}^{n} 1.L_k(x).$$

Vérification avec *Matlab* dans le cas $n = 3$: on utilise les fonctions *Lagrange* et *interpolLagrange de l'exercice 7.3.2* (solution p. 237).

```
» syms x0 x1 x2 x3 x
» X=[x0 x1 x2 x3] ;
» n=3 ;
» for i =1 :n+1,
   Lk(i)=Lagrange(i,x,X) ;
» end
» simplify(sum(Lk))
ans = 1
```

On calcule aussi le polynôme d'interpolation associé aux valeurs

$$\{f_0, f_1, ..., f_n\} = \{1, 1, ..., 1\}$$

```
» Y=[1 1 1 1 ] ;
» Px=interpolLagrange(x,X,Y) ;
» simplify(Px)
ans = 1
```

b) Il suffit de remarquer que

$$a_k = \prod_{i \neq k} (x_k - x_i)$$

est le dénominateur de $L_k(x)$, et que, pour $x \notin \{x_0, x_1, ..., x_n\}$, le numérateur de $L_k(x)$ peut s'écrire

$$\frac{Q(x)}{(x - x_k)}.$$

On effectue la vérification avec *Matlab* : on calcule $Q(x)$ en utilisant la fonction *prod.*

```
» Qx=prod(x-X)
Qx = (x-x0)*(x-x1)*(x-x2)*(x-x3)
```

On calcule de même $A(k)$ en utilisant un tableau intermédiaire Tab dans lequel ne figure pas $X(k)$.

```
» for k=1 :n+1
    Tab=[X(1 :k-1) X(k+1 :n+1)] ;
    A(k)=prod(X(k)-Tab) ;
end
» A(1)
ans =(x0-x1)*(x0-x2)*(x0-x3)
```

On calcule

$$T = \left[\frac{Q(x)}{(x-x_0)a_0}, \cdots \frac{Q(x)}{(x-x_n)a_n} \right]$$

en utilisant les opérations globales sur les tableaux (./, .*)

```
» T=Qx./((x-X).*A);
» T(1)
ans =(x-x1)*(x-x2)*(x-x3)/(x0-x1)/(x0-x2)/(x0-x3)
```

On vérifie que les éléments de T sont égaux à ceux de $L_{k.}$

```
» T-Lk
ans =[ 0, 0, 0, 0]
```

c) Des deux questions précédentes, on déduit

$$1 = \sum_{k=0}^{n} L_k(x) = \sum_{k=0}^{n} \frac{Q(x)}{(x-x_k)a_k} = Q(x) \sum_{k=0}^{n} \frac{1}{(x-x_k)a_k},$$

d'où

$$Q(x) = \frac{1}{\sum\limits_{k=0}^{n} \frac{1}{(x-x_k)a_k}}.$$

Et enfin

$$\begin{aligned} P_n(x) &= \sum_{k=0}^{n} f_k L_k(x) \\ &= Q(x) \sum_{k=0}^{n} f_k \frac{1}{(x-x_k)a_k} \\ &= \frac{\sum\limits_{k=0}^{n} \frac{f_k}{(x-x_k)a_k}}{\sum\limits_{k=0}^{n} \frac{1}{(x-x_k)a_k}}. \end{aligned}$$

Vérifions la première égalité avec *Matlab*, en comparant

$$S = \frac{1}{\sum\limits_{k=0}^{n} \frac{1}{(x-x_k)a_k}}$$

avec $Q(x)$.

```
» S=1/sum(1./((x-X).*A));
» S=simplify(S)
S = (-x+x3)*(-x+x2)*(x-x1)*(x-x0)
» simplify(S-Qx)
ans =0
```

2) a) La fonction $A = coeff(X)$ place dans le tableau A les coefficients $[a_0, a_1, ..., a_n]$.

```
function A=coeff(X)
n=length(X)-1;
for k=1 :n+1
   A(k)=coeffk(X,k,n);
end
```

Elle utilise une fonction auxiliaire $Ak = coeffk(X, k, n)$ qui calcule un élément

$$A(k) = \prod_{i \neq k} (X(k) - X(i)) .$$

```
function Ak=coeffk(X,k,n)
Ak=1;
for i=1 :k-1
   Ak=Ak*(X(k)-X(i));
end
for i=k+1 :n+1
   Ak=Ak*(X(k)-X(i));
end
```

b) On calcule en même temps

$$N = \sum_{k=0}^{n} \frac{Y(k)}{(x - X(k))A(k)}$$

et

$$D = \sum_{k=0}^{n} \frac{1}{(x - X(k))A(k)},$$

ce qui permet de ne calculer qu'une fois

$$quot = \frac{1}{(x - X(k))A(k)}.$$

Le résultat final est évidemment N/D.

```
function Pnx=lagrangeBar(x,X,Y,A)
n=length(X)-1;
N=0; D=0;
for k=1 :n+1
   quot=1/(A(k)*(x-X(k)));
   N=N+Y(k)*quot;
   D=D+quot;
end
Pnx=N/D;
```

3) Pour calculer chacun des coefficients $A(k)$, on doit répéter n fois l'instruction

$$Ak \longleftarrow Ak \times (X(k) - X(i)),$$

soit $2 \times n$ opérations élémentaires. Le nombre d'opérations pour obtenir l'ensemble du tableau A est donc

$$(n+1).2n = 2n^2 + 2n.$$

Pour calculer $lagrangeBar(x, X, Y, A)$, on répète $n+1$ fois le calcul de $quot$ (3 opérations), la mise à jour de N (2 opérations) et celle de D (1 opération). Le nombre total d'opérations est donc

$$(n+1)\,(3+2+1) = 6n + 6.$$

Le principal avantage de cette formule barycentrique est que le calcul de A est à faire une seule fois (pour X donné), et que chaque nouveau calcul de Pnx ne nécessite que $O(n)$ opérations.

Exercice 7.3.7

1) On calcule ce nombre u_n d'opérations par récurrence sur n en reprenant la version récursive de l'algorithme d'Aitken, dont la solution est donnée p 7.4.

- Si $n = 0$, on doit effectuer l'instruction

$$y \longleftarrow Y(1)$$

qui ne nécessite aucune opération arithmétique élémentaire.

- pour $n > 0$, on doit effectuer la suite d'instructions

$$\begin{array}{l} Q \longleftarrow aitkenRec(x, X(2:n+1), Y(2:n+1)) \\ R \longleftarrow aitkenRec(x, X(1:n), Y(1:n)) \\ y \longleftarrow ((X(n+1)-x) * R - (X(1)-x) * Q)/(X(n+1) - X(1)) \end{array}$$

d'où

$$u_n = 2u_{n-1} + 7.$$

On a alors

$$u_0 = 0, \quad u_1 = 7, \quad u_2 = 15.$$

On pourra montrer par récurrence que

$$u_n = 7(2^n - 1).$$

Ce nombre croît très vite avec n.

2) On construit le tableau T en utilisant les relations

$$T(i,1) = P_{\{X(i)\}} = Y(i),$$

puis

$$\begin{aligned} & T(i,j) \\ = \; & P_{\{X(i-j+1),\ldots,X(i)\}} \\ = \; & \frac{(X(i)-x)\,P_{\{X(i-j+1),\ldots,X(i-1)\}}}{(X(i)-X(i-j+1))} \\ & -\frac{(X(i-j+1)-x)\,P_{\{X(i-j+2),\ldots,X(i)\}}}{(X(i)-X(i-j+1))} \\ = \; & \frac{(X(i)-x)\,T(i-1,j-1)-(X(i-j+1)-x)\,T(i,j-1)}{(X(i)-X(i-j+1))} \end{aligned}$$

```
function T= aitkenIte(x,X,Y)
n=length(X)-1 ;
for i=1 :n+1
   T(i,1)=Y(i) ;
end
for j=2 :n+1
   for i=j :n+1
      num=(X(i)-x)*T(i-1,j-1)-(X(i-j+1)-x)*T(i,j-1) ;
      T(i,j)=num/(X(i)-X(i-j+1)) ;
   end
end
```

On teste avec les données de l'exemple 7.1.4.1, et on retrouve le tableau triangulaire inférieur, complété par des 0.

```
» X=[0 2 4] ;
» Y=[3 -1 3] ;
» T=aitkenIte(5,X,Y)
T =
    3 0 0
   -1 -7 0
    3 5 8
```

3) La première colonne du tableau s'obtient sans effectuer d'opérations arithmétiques. Tous les autres termes du tableau se calculent à l'aide de 7 opérations. Le tableau entier s'obtient donc colonne après colonne par un nombre d'opérations égal à

$$7n + 7(n-1) + \ldots + 7 = 7\frac{n(n+1)}{2}.$$

Le nombre d'opérations est en $O(n^2)$, comme pour la méthode de Lagrange.
Cet algorithme itératif est surtout intéressant si on ajoute un point d'interpolation supplémentaire (x_{n+1}, y_{n+1}). Il n'y a pas lieu de recalculer le tableau T, mais seulement d'y ajouter une $(n+1)^{ième}$ ligne, soit $7(n+1)$ nouvelles opérations.

Chapitre 8

Intégration numérique

8.1. Description de la méthode

On donne une fonction numérique f définie et intégrable sur $[a, b]$ supposée connue (grâce aux mesures) en $(n+1)$ points selon le tableau :

x	x_0	x_1	$\cdots$	x_n
$f(x)$	$f_0 = f(x_0)$	$f_1 = f(x_1)$	$\cdots$	$f_n = f(x_n).$

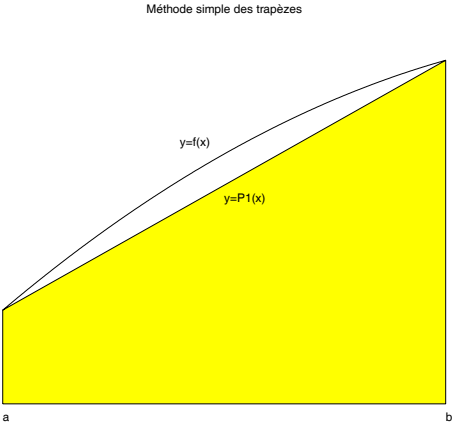
Pour approcher le nombre exact $I = \int_a^b f(t)dt$, l'idée est la suivante :

– on remplace $f(x)$ par son polynôme d'interpolation $P_n(x)$,

– on calcule $I_n = \int_a^b P_n(t)dt$,

– on estime l'erreur $|I - I_n|$.

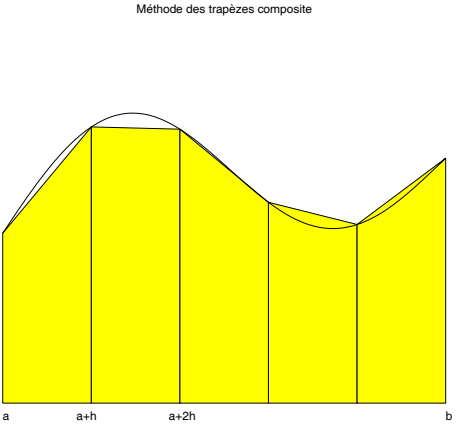
Une telle méthode est appelée souvent méthode de quadrature. On décrira celles qui font appel à l'interpolation polynomiale de degré 0, 1 et 2. On obtient respectivement les formules simples des rectangles, des trapèzes et de Simpson.



En pratique, on subdivise l'intervalle $[a,\ b]$ en plusieurs petits intervalles :

$$[a + (i-1)h,\ a + ih], \quad i = 1, ..., n$$

de longueur $h = (b-a)/n$, auxquels on applique la formule simple puis par sommation, on déduit les formules dites composites.



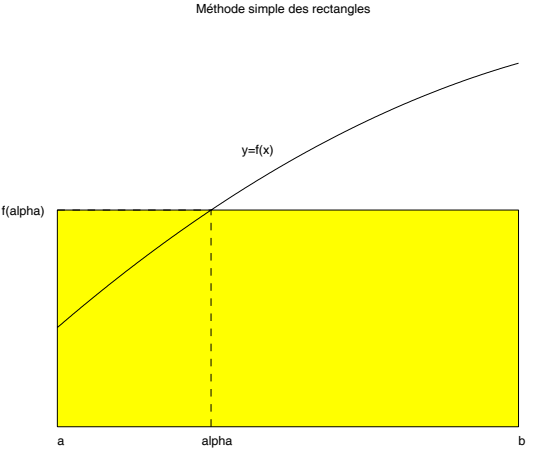
Le choix de la méthode dépend généralement du type de fonctions qu'on doit intégrer. On peut éventuellement combiner les trois méthodes (rectangles, trapèzes, Simpson) sur des intervalles bien choisis.

8.2. Méthode des rectangles

8.2.1. *Formules simples*

Supposons que f est connue en un point $\alpha \in [a,b]$. Le polynôme d'interpolation de f est la constante $P_0(x) = f(\alpha)$. Le nombre I est donc approché par

$$\left\| I_0 = \int_a^b P_0(t)dt = \int_a^b f(\alpha)dt = (b-a)f(\alpha). \right.$$



Dans le cas particulier où le point α est le milieu de $[a,b]$

$$\alpha = \frac{a+b}{2},$$

on obtient

$$I_0 = (b-a)f\left(\frac{a+b}{2}\right).$$

C'est la formule simple des rectangles point-milieu.

8.2.2. *Formules composites*

On généralise les formules précédentes à $(n+1)$ points équidistants

$$x_0 = a,\ x_1 = a+h, ...,\ x_i = a+ih, ..., x_n = b,$$

où

$$h = \frac{b-a}{n}.$$

En appliquant le même principe d'interpolation de degré 0 sur chaque intervalle

$$[x_i, x_{i+1}],$$

on approche $f(x)$ par $P_0(x) = f(\alpha_i)$, où

$$\alpha_i \in [x_i, x_{i+1}],$$

et on obtient la formule composite des rectangles :

$$\left\| I_R = \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} f(\alpha_i)dt = \frac{b-a}{n} \sum_{i=0}^{n-1} f(\alpha_i), \right.$$

qui devient dans le cas des rectangles point-milieu

$$\left\| I_M = \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} f(x_i)dt = \frac{b-a}{n} \sum_{i=0}^{n-1} f\left(\frac{x_i+x_{i+1}}{2}\right). \right.$$

8.3. Méthode des trapèzes

8.3.1. *Formule simple*

On utilise l'interpolation à deux points. Si on suppose connue f aux deux points a et b, on a vu que

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

Par intégration, on obtient

$$\left\| I_1 = \int_a^b P_1(t)dt = \frac{(b-a)(f(a)+f(b))}{2}. \right.$$

8.3.2. *Formule composite*

En généralisant le même principe aux $(n+1)$ points équidistants précédents, on obtient

$$\left\| \begin{aligned} I_T &= \sum_{i=0}^{n-1} \int_{x_i}^{x_{i+1}} P_1(t)dt = \frac{b-a}{n} \sum_{i=0}^{n-1} \frac{(f(x_i)+f(x_{i+1}))}{2} \\ &= \frac{b-a}{n} \left[\frac{1}{2}(f(a) + \frac{1}{2} f(b)) + \sum_{i=1}^{n-1} f(x_i) \right]. \end{aligned} \right.$$

On utilise usuellement la dernière formule plus économique en nombre d'opérations arithmétiques.

8.4. Méthode de Simpson

8.4.1. *Formule simple*

On suppose connue f aux trois points équidistants de $[a,b]$

$$x_0 = a,\ \ x_1 = \frac{a+b}{2} = a+h,\ \ x_2 = a+2h,$$

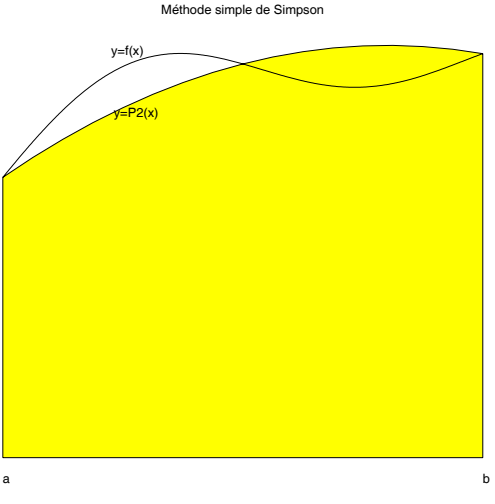
le pas est $h = (b-a)/2$. Alors

$$\begin{aligned} P_2(x) \quad = \quad & \frac{\left(x - \dfrac{a+b}{2}\right)(x-b)}{2h^2} f(a) \\ & + \frac{(x-a)(x-b)}{h^2} f\left(\frac{a+b}{2}\right) \\ & + \frac{(x-a)\left(x - \dfrac{a+b}{2}\right)}{h^2} f(b), \end{aligned}$$

et après intégration, on obtient

$$\left\| I_2 = \int_a^b P_2(t)dt = \frac{b-a}{6}\left[f(a) + 4f\left(\frac{a+b}{2}\right) + f(b)\right]. \right.$$

C'est la formule simple dite de Simpson.



8.4.2. *Formule composite*

On réitère la formule précédente en partageant l'intervalle $[a, b]$ en

$$s = \frac{n}{2}$$

intervalles $[x_{2i}, x_{2i+2}]$, centrés en x_{2i+1}, de longueur

$$2h = \frac{b-a}{s} = \frac{2(b-a)}{n}$$

pour $i = 0, 1, 2, ..., s-1$; on obtient, d'abord sur $[x_{2i}, x_{2i+2}]$,

$$\int_{x_{2i}}^{x_{2i+2}} P_2(t)dt = \frac{h}{3}\left[f(x_{2i}) + 4f\left(x_{2i+1}\right) + f(x_{2i+2})\right],$$

et ensuite, la formule composite de Simpson

$$\left\| \begin{array}{l} I_S = \sum_{i=0}^{s-1} \int_{x_{2i}}^{x_{2i+2}} P_2(t)dtt \\ = \frac{h}{3}\left[f(a) + f(b) + 2\sum_{i=1}^{s-1} f(a+2ih) + 4\sum_{i=0}^{s-1} f(a+(2i+1)h)\right]. \end{array} \right.$$

8.5. Gestion d'erreur

On se bornera à l'étude de l'erreur mathématique commise dans la méthode des trapèzes et celle de Simpson.

8.5.1. *Erreur dans la méthode des trapèzes*

En formule simple on a vu que la quantité

$$I_1 = \frac{(b-a)(f(a)+f(b))}{2}$$

approche

$$I = \int_a^b f(t)dt.$$

En posant $b = a + h$, il vient

$$\varphi(h) = I - I_1 = \int_a^{a+h} f(t)dt - \frac{h(f(a)+f(a+h))}{2}.$$

En supposant ensuite que f admet des dérivées successives continues jusqu'à l'ordre 2, on montre, grâce à un développement de Taylor de $\varphi(h)$ à l'ordre 2 qu'on a

$$|I - I_1| \leqslant \frac{h^3}{12} \max_{x\in[a,b]} |f''(x)| .$$

En formule composite, on déduit que

$$\left\| |I - I_T| \leqslant \frac{(b-a)^3}{12n^2} \max_{x\in[a,b]} |f''(x)| .$$

8.5.2. *Erreur dans la méthode de Simpson*

Utilisant la méthode précédente on trouve l'estimation de l'erreur suivante, en composite, pour des fonctions f admettant des dérivées successives continues jusqu'à l'ordre 4

$$\left\| |I - I_S| \leqslant \frac{(b-a)^5}{180n^4} \max_{x\in[a,b]} \left|f^{(4)}(x)\right| .$$

Lorsqu'on connait une majoration M de $\left|f^{(4)}(x)\right|$, le pas choisi h qui permet d'avoir au plus une erreur ε vérifie nécéssairement

$$\frac{(b-a)^5}{180n^4} M \leqslant \varepsilon,$$

d'où

$$\frac{(b-a)}{180} h^4 . M \leqslant \varepsilon,$$

ou bien

$$h \leqslant \sqrt[4]{\frac{180\varepsilon}{(b-a)M}}.$$

Le pas h est en $O\left(\varepsilon^{1/4}\right)$.

Si on veut éviter d'utiliser une majoration de $\left|f^{(4)}(x)\right|$, on effectue les approximations I_S^h et $I_S^{h/2}$ de I de pas respectifs h et $h/2$. La deuxième est peu coûteuse, puisqu'elle utilise certaines valeurs déjà calculées. On réitèrera les calculs, en remplaçant h par $h/2$, et en effectuant le test d'arrêt

$$\left|I_S^h - I_S^{h/2}\right| < \varepsilon.$$

8.6. Exercices

8.6.1. *Utilisations des méthodes des trapèzes et de Simpson*

1) Créer le fichier *f1.m* correspondant à la fonction

$$x \mapsto f_1(x) = \exp(-x^2).$$

2) Donner son graphe sur l'intervalle $[0, 1]$ et dire quelle est la méthode d'intégration numérique adaptée pour le calcul approché de

$$I = \int_0^1 e^{-x^2} dx.$$

3) Calculer les valeurs approchées I_t et I_s de I par la méthode des trapèzes et de Simpson lorsqu'on utilise la subdivision $(0, 1/4, 2/4, 3/4, 1)$.

4) Comparer avec la valeur donnée par la fonction *int* de *Matlab*.

(solution p. 257)

8.6.2. *Programmation*

1) Ecrire une fonction

$$It = ITrapezes(fonc, a, b, n),$$

qui permet de calculer une valeur approchée de

$$\int_a^b fonc(x)dx,$$

en utilisant la formule composite des trapèzes, avec le pas

$$h = (b - a)/n.$$

On pourra écrire deux versions :

a) l'une utilisant un traitement itératif naturel ;

b) l'autre utilisant les opérations globales sur les tableaux

$$X = [a\ a + h...a + nh],\ Y = fonc(X).$$

2) Ecrire de même une fonction

$$Is = ISimpson(fonc, a, b, n).$$

(solution p. 259)

8.6.3. *Calculs approchés d'intégrales et gestion d'erreur*

Soit la fonction

$$x \mapsto f(x) = \exp(-x).\cos(x).$$

1) Donner le graphe de cette fonction et une majoration de $|f''(x)|$ et de $\left|f^{(4)}(x)\right|$ sur l'intervalle $[0, 2\pi]$.

2) Combien d'intervalles doit-on prendre pour obtenir, par la méthode composite des trapèzes, une valeur approchée à 10^{-3} près de

$$I = \int_0^{2\pi} e^{-x}\cos(x)dx.$$

3) Même question avec la méthode de Simpson.

4) Calculer It et Is pour les subdivisions respectives obtenues.

5) Vérifier le calcul de It en utilisant les fonctions prédéfinies de *Matlab* **trapz** et **quad**.

6) Comparer les résultats précédents avec la valeur exacte donnée par *Matlab*.

(solution p. 261)

8.7. Solutions

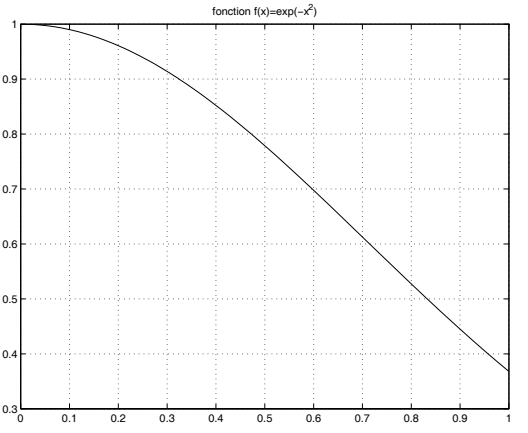
Exercice 8.6.1

1) On définit le fichier texte f1.m

```
% fonction calculant y=exp(-x^2)
function y=f1(x)
   y=exp(-x.^2);
```

2) On utilise la commande *fplot* pour obtenir le graphe de la fonction ainsi définie.

```
» fplot('f1',[0 1])
» grid on
» title('fonction f(x)=exp(-x^2)')
```



L'allure de la courbe suggère l'utilisation de la méthode des trapèzes ou de Simpson.

3) On utilise les formules composites des trapèzes et de Simpson avec la subdivision $(0, 1/4, 2/4, 3/4, 1)$. Ici, le pas est

$$h = (b - a)/4 = 1/4.$$

```
» It=((f1(0)+f1(1))/8)+((f1(1/4) +...
        f1(2/4)+f1(3/4))/4)
I =0.7430
» Is=((f1(0)+f1(1))/12)+((2*f1(2/4) + ...
        4*f1(1/4)+4*f1(3/4))/12)
Is = 0.7469
```

4) Sous *Matlab* on a

```
» syms x
» Iexact =int(f1(x),0,1)
Iexact =1/2*erf(1)*pi^(1/2)
» double(Iexact)
ans = 0.7468
```

La fonction f_1 n'a pas de primitive explicite. *Matlab* utilise la fonction :

$$x \longmapsto \operatorname{erf}(x) = \int_0^x \exp(-t^2)dt.$$

On constate que *Is* est plus proche de *Iexact* que *It.*

Exercice 8.6.2

1) a) On utilise les formules

$$\begin{cases} S = \dfrac{(fonc(a) + fonc(b))}{2} + \displaystyle\sum_{i=1}^{n-1} fonc(a + ih), \\ It = hS. \end{cases}$$

Lorsqu'on déclare

$$function\ It = ITrapezes(fonc, a, b, n),$$

le paramètre $fonc$ représente le nom du fichier contenant la fonction mathématique $fonc$. On utilise alors *feval*$(fonc, x)$ pour calculer $fonc(x)$. D'où le programme itératif :

```
function It =ITrapezesA(fonc,a,b,n)
h = (b-a)/n ;
S=(feval(fonc,a)+feval(fonc,b))/2 ;
for i=1 :n-1
    S=S+feval(fonc,a+i*h) ;
end
It=S*h ;
```

On le teste, sur la fonction *f1* de l'exercice 8.6.1 (c'est le nom de cette fonction, sous forme de chaîne de caractères, qui doit être passé en paramètre).

```
» ITrapezesA('f1',0,1,4)
ans =0.7430
```

b) Pour créer le tableau

$$X = [a\ a + h...a + nh],$$

dont la dernière valeur $a + nh$ est exactement b, et pour éviter le cas où l'arrondi de $a + nh$ est strictement supérieur à b, on déclare

$$X = a : h : b + h/2.$$

On a

```
function It =ITrapezesB(fonc,a,b,n)
h = (b-a)/n ;
X =a :h :b+h/2 ;
Y= feval(fonc,X) ;
It= (h/2)*(Y(1)+Y(n+1))+h*sum(Y(2 :n)) ;
```

Ici $sum(Y(2 : n))$ calcule

$$\sum_{i=1}^{n-1} fonc(a + ih).$$

On teste sur la fonction $f1$

```
» ITrapezesB('f1',0,1,4)
ans =0.7430
```

L'utilisation des opérations globales sur les tableaux de valeurs conduit à une syntaxe moins naturelle, mais permet de minimiser les temps de calcul, lorsque le nombre n d'intervalles devient grand :

```
» tic ;It=ITrapezesA('f1',0,1,15000);toc
elapsedtime = 0.9237
» tic ;It=ITrapezesB('f1',0,1,15000);toc
elapsedtime = 0.0031
```

2) On définit les tableaux

$$X_0 = [a\ b],$$
$$X_1 = [a+2h\ a+4h...b-2h],$$
$$X_2 = [a+h\ a+3h...b-h],$$

en tenant compte comme précédemment des erreurs éventuelles d'arrondi. On calculera *sum(Y0), sum(Y1)* et *sum(Y2)* représentant les trois sommes respectives de la formule de Simpson :

$$I_s = \frac{h}{3}\left[f(a)+f(b)+2\sum_{i=1}^{s-1} f(a+2ih)+4\sum_{i=0}^{s-1} f(a+(2i+1)h)\right].$$

D'où

```
function Is =ISimpson(fonc,a,b,n)
h = (b-a)/n ;
s=n/2 ;
X0=[a b] ;
X1= a+2*h :2*h :b-h ;
X2=a+h :2*h :b ;
Y0=feval(fonc,X0) ;
Y1=feval(fonc,X1) ;
Y2=feval(fonc,X2) ;
Is=h/3*(sum(Y0)+2*sum(Y1)+4*sum(Y2)) ;
```

Le test sur $f1$ donne

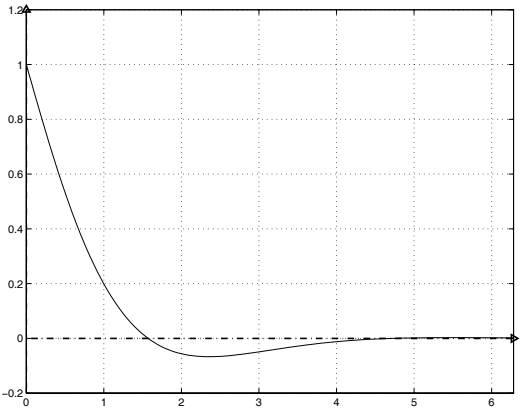
```
» ISimpson('f1',0,1,4)
ans =0.7469
```

Exercice 8.6.3

1) On déclare la fonction et on trace son graphe

```
function y=f2(x)
y=exp(-x).*cos(x);
```

```
» fplot('f2',[0 2*pi])
» grid on ; dessineRepere
```



On calcule les dérivées seconde et quatrième de $f2$

```
» syms x real
» f2seconde=diff(diff(f2(x)))
f2seconde = 2*exp(-x)*sin(x)
» f2quatrieme=diff(diff(f2seconde))
f2quatrieme = -4*exp(-x)*cos(x)
```

On en déduit que ces dérivées sont majorées respectivement par 2 et 4 sur l'intervalle $[0,\,2\pi]$.

2) Pour trouver les subdivisions demandées on utilise les estimations d'erreur

$$|I - I_T| \leqslant \frac{(b-a)^3}{12n^2} \max_{x\in[a,b]} |f''(x)|\,,$$

$$|I - I_S| \leqslant \frac{(b-a)^5}{180n^4} \max_{x\in[a,b]} \left|f^{(4)}(x)\right|.$$

Il vient, pour la méthode des trapèzes

```
» a=0 ;b=2*pi ;epsilon=1e-3 ;
» nT=ceil(((b-a)^3/(6*epsilon))^(1/2))
nT = 204
```

3) De même, pour celle de Simpson

```
» nS=ceil(((b-a)^5*4/(180*epsilon))^(1/4))
nS = 22
```

Il faut donc prendre 204 intervalles (et 205 points) pour la méthode composite des trapèzes, alors qu'il suffit d'utiliser $n+1=23$ points, et $s=n/2=11$ intervalles pour obtenir la même précision de 10^{-3} par la méthode de Simpson.

4)

```
» It=ITrapezesB('f2',0,2*pi,204)
It = 0.4991
» Is = ISimpson('f2',0,2*pi,22)
Is = 0.4990
```

5) La fonction prédéfinie de *Matlab,* **trapz***(X,Y)*, applique la méthode composite des trapèzes. On lui passe en paramètres

$$X=[x_0,x_1,...,x_n]$$

et

$$Y=[f(x_0),f(x_1),...,f(x_n)].$$

On obtient ici

```
» n=204 ;
» X1=0 :2*pi/n :2*pi+pi/n ;
» X=X1' ;
» Y=f2(X) ;
» ItMatlab=trapz(X,Y)
ItMatlab = 0.4991
```

$quad('f',a,b,eps)$ calcule, en utilisant la méthode de Simpson, une valeur approchée à eps près de

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

```
» quad('f2',0,2*pi,1e-3,1)
ans = 0.4991
```

6) La fonction prédéfinie *int* de *Matlab* donne, lorsque cela est possible, la valeur exacte de l'intégrale.

```
» Iexact=int(f2(x),0,2*pi)
Iexact = -1/2*exp(-2*pi)+1/2
» double(Iexact)
ans = 0.4991
```

Bibliographie

[BAR 02] A. BARRAUD, "Outils d'analyse numérique pour l'automatique", Hermes, Paris 2002.

[BRE 88] C. BRÉZINSKI, "Algorithmique numérique", Ellipses, Paris, 1988.

[CAL 77] B. CALVO, J. DOYEN, A. CALVO, F. BOSCHET, "Cours d'analyse III, développements limités, courbes, équations différentielles", Collection U, Armand Colin, Paris, 1977.

[DAU 97] M. DAUMAS, J-M. MULLER, "Qualité des calculs sur ordinateur", *Informatique*, Masson, Paris, 1997.

[DEM 79] B. DEMIDOVITCH, I. MARON, "Mathématiques", Mir, Moscou, 1979.

[LAR 96] C. LARCHER, M. PRIENTE, J.-C. ROY, "L'essentiel du cours, 300 exercices commentés et résolus", Techniplus, 1996.

[LEL 72] J. LELONG-FERRAND, J.M. ARNAUDIES, "Cours de Mathématiques, tome 2 : analyse", Dunod, Paris, 1972.

[MON 90] J.M. MONIER, "Analyse, tome I, 800 exercices résolus et 18 sujets d'étude, 1er cycle universitaire", Dunod, Paris, 1990.

[SAI 89] J. H. SAIAC, "L'informatique appliquée au calcul scientifique", Dunod, Paris, 1989

[THE] R. THÉODOR, "Initiation à l'analyse numérique", *CNAM, cours A*, Masson,3ième édition, 1992.

[WIL 65] J.H. WILKINSON, "The algebraic eigenvalue problem", *monographs on numerical analysis*, Oxford science publications, 1965.

Index