ANALYSE NON STANDARD
ET CALCUL NUMÉRIQUE SUR ORDINATEUR

Emile URLACHER

0.— Introduction

Un ordinateur ne sait traiter que des nombres entiers. Cette réalité est parfois reconnue (voir pour les coordonnées des pixels d'un écran bit-map), d'autres fois elle est cachée. Ainsi les réels de l'ordinateur ne sont en fait que des entiers les représentant. Les deux représentations usuelles sont connues sous les dénominations suivantes :

— la virgule fixe,
— la virgule flottante.

Chacune d'elles a pour objectif d'éviter les inconvénients de l'autre. Pour la virgule fixe, overflow (dépassement de capacité) et underflow, sont monnaie courante et de fréquents *changements d'échelle* sont nécessaires lors des calculs intermédiaires. La virgule flottante exige, quant à elle, des temps de calcul importants pour des précisions parfois inutiles (voir le tracé d'une courbe sur un écran bit-map) et quelquefois illusoires (1). Les deux représentations reposent cependant sur la même assertion : à une *très petite* échelle ou avec une *très grande* unité les entiers simulent bien le *continu* et donc les réels.

Alors qu'elles n'ont qu'un sens heuristique en mathématiques classiques, les notions de *très petite* échelle ou de *très grande* unité peuvent être formulées, donc contrôlées, de manière à la fois simple et rigoureuse dans le cadre de l'analyse non standard (2).

Plus précisément l'analyse non standard permet de considérer un **ordinateur "idéal"** (ou idéel), modèle de l'ordinateur usuel, et de disposer de chemins balisés pour la construction d'algorithmes de calculs utilisant uniquement les entiers. Quitte à ce que certaines constantes soient bien choisies, ces algorithmes seront exécutables sur l'ordinateur usuel. Du recours exclusif aux entiers résulte un gain de temps de calcul important et, évidemment, une transportabilité accrue des programmes.

1.— Un ordinateur idéal

Soit ω un entier positif infiniment grand fixé. J'appelle ordinateur *idéal* (dans la suite, j'écrirai simplement OI) un ordinateur formel qui sait gérer (comparer,

(1) Voir l'article de R. SEROUL : "*Equations différentielles et nombres entiers*" '*L'Ouvert*' n° 48 (Septembre 1987).
(2) En annexe est donnée une très brève introduction à l'analyse non standard.

additionner, soustraire, multiplier et faire le quotient entier) tous les entiers n pour lesquels il existe k standard avec $|n| \leq k\omega$ (autrement dit les entiers de l'ordre de ω).

Remarques

1. Le dépassement de capacités est possible sur l'OI (par exemple pour ω^2).

2. Les entiers infiniment grands pouvant être distingués des entiers standard, l'OI permet de prendre en compte des *entiers longs* et ainsi d'orienter les algorithmes afin de gagner éventuellement de la mémoire dans des problèmes où ceux-ci interviennent (les entiers longs occupant évidemment beaucoup de mémoire).

Tout algorithme de calcul exécutable sur l'OI donne immédiatement un algorithme exécutable sur l'ordinateur usuel en remplaçant ω par un entier N disponible sur ce dernier et judicieusement choisi.

L'analyse non standard nous donne de bonnes raisons de penser que si l'algorithme *idéal* était *bon* alors l'algorithme *concret* ainsi obtenu le sera aussi. En fait en choisissant ω infiniment grand j'ai, en quelque sorte, répondu par avance à toute condition qui pouvait porter sur la *grandeur* de N.

Afin de pouvoir traiter tout problème numérique je vais définir une représentation des réels par des entiers.

2.— Une représentation des réels par des entiers

Je note par $[a]$ la partie entière de a. Tout réel x est représenté par le couple d'entiers (X_0, R) défini comme suit

$$X_0 = [\omega x]$$
$$R = [\omega(\omega x - X_0)].$$

Il en résulte immédiatement la

Proposition : Tout réel non infiniment grand est représenté par un couple d'entiers (X_0, R) de l'OI.

Définitions : L'entier X_0 est appelé **représentation** de x et R est appelé le reste de la représentation.
Un entier X est dit un représentant de x si $|X - X_0|$ est standard.

Ces définitions appellent les

Commentaires et remarques :
1. La représentation X_0 du réel x détermine x avec une erreur inférieure à $1/\omega$. Un représentant X de x détermine x avec une erreur de l'ordre de $1/\omega$ (i.e. $|x - (X/\omega)| \leq k/\omega$, avec k standard). A chaque réel on associe la collection de tous ses représentants (on représente un point par une tache, ce qui nous est assez familier).

Deux réels standard distincts ont des collections disjointes de représentants .

Pour calculer une valeur approchée d'un réel x_o, je construis un algorithme pouvant tourner sur l'OI et qui permet de calculer un représentant X de x_o.

2. Le nombre de boucles d'un algorithme exécutable sur l'OI peut être de l'ordre de ω. Ainsi en partant avec une erreur initiale de l'ordre de $1/\omega$, sachant que les erreurs s'accumulent, on peut aboutir à un résultat non satisfaisant (c'est-à-dire à un entier qui n'est pas un représentant du réel à approcher). D'où résulte la nécessité de tenir compte du reste R. En effet (X_0, R) détermine x à $1/\omega^2$ près.

3.— Une représentation des fonctions

A la représentation des réels ainsi définie correspond naturellement une représentation des fonctions.

Ainsi, soit

$$f : [a, b] \longrightarrow \mathbb{R}$$

une fonction réelle.

Je note $\{l, l+1, \ldots, l+p\}$ l'intervalle de $\mathbb{Z}$ constitué par les représentations des éléments de $[a, b]$.

Définition : On dit qu'une application

$$F : \{l, l+1, \ldots, l+p\} \longrightarrow \mathbb{Z}$$

(i) est une **représentation** de f si
$\forall k \in \{l, l+1, \ldots l+p\}$, $F_k = F(k) = [\omega f(\frac{k}{\omega})]$
(ii) est un **représentant** de f si $\forall k \in \{l, l+1, \ldots, l+p\}$,
F_k est un représentant de $\omega f(\frac{k}{\omega})$.

Il correspond évidemment à la représentation de f une fonction **reste**

$$R : \{l, l+1, \ldots, l+p\} \longrightarrow \{0, 1, \ldots, \omega - 1\}$$

définie par

$$R_k = \left[\omega(\omega f(\frac{k}{\omega}) - F_k)\right], \qquad k \in \{l, l+1, \ldots l+p\}.$$

Et je peux affirmer

Proposition : Si f est une fonction standard, sa représentation, son reste et tout représentant est défini sur des entiers de l'OI et à valeurs dans ceux-ci.

Il est facile de vérifier que

Remarque : Si la fonction f est standard de classe C^1, si F est un représentant de f et X un représentant de x alors F_X est un représentant de $f(x)$.

Dans de nombreux problèmes numériques, tels le tracé de courbes ou l'intégration numérique d'équations différentielles, on est amené à tabuler des fonctions. Ceci se traduit dans notre cas par la détermination de représentants de ces fonctions.

La notion de code, que je vais définir, se révèle jouer un rôle essentiel dans les calculs ainsi requis.

Avec les notations qui précèdent

Définition : On dit qu'une application

$$C : \{l, l+1, \ldots, l+p\} \longrightarrow \mathbb{Z}$$

est un code de f s'il existe un représentant F de f tel que

$$\forall k \in \{l, l+1, \ldots, l+p-1\} \ , \ C_k = F_{k+1} - F_k.$$

On dira encore que C est le code correspondant au représentant F de f.

Remarques

1. Un représentant F d'une fonction f est parfaitement déterminé par le code lui correspondant et une de ses valeurs F_k. Ainsi pour le tracé du graphe de f, il suffit de connaître une telle valeur et un tel code.

2. Si C est le code correspondant à la représentation d'une fonction standard dérivable, alors

$$\forall k \in \{l, l+1, \ldots, l+p-1\} \ , \ C_k = \left[f'(\frac{k}{\omega})\right] + \alpha_k$$

où

$$\alpha_k \in \{-1, 0, 1\}.$$

Par conséquent quel que soit k, C_k est standard, donc un entier *petit* de l'ordinateur modèle.

En appelant C code d'ordre 1 de f, je peux définir D code d'ordre 2 de f par

$$\forall k \in \{l, l+1, \ldots, l+p-2\} \qquad D_k = C_{k+1} - C_k$$

et de même des codes d'ordre 3, 4, etc...

A toutes les fonctions code est associée la même fonction reste, notamment celle associée au représentant primitif.

4.— Application

Je me propose de construire un algorithme permettant de tabuler la fonction $f(x) = x^2$ sur l'intervalle $[0, 1]$, autrement dit de calculer $F_k = \left[\frac{k^2}{\omega}\right]$ pour $k = 0, 1, 2, \ldots, \omega$.

Par définition

$$\frac{k^2}{\omega} = F_k + \frac{R_k}{\omega}$$

ou encore

$$k^2 = F_k \times \omega + R_k \qquad (1)$$

où

$$\begin{cases} F_k & = k^2 \text{ div } \omega \\ R_k & = k^2 \bmod \omega \end{cases}$$

De même

$$\begin{cases} F_{k+1} & = (k+1)^2 \text{ div } \omega \\ R_{k+1} & = (k+1)^2 \bmod \omega \end{cases}$$

et d'après (1)

$$\begin{cases} F_{k+1} & = F_k + (2k+1+R_k) \text{ div } \omega \\ R_{k+1} & = (2k+1+R_k) \bmod \omega \end{cases}$$

ou encore en passant au code d'ordre 1

$$\begin{cases} C_k & = (2k+1+R_k) \text{ div } \omega \\ R_{k+1} & = (2k+1+R_k) \bmod \omega \end{cases}$$

Ce qui me donne l'algorithme suivant

```
F := 0; R := 0;
Pour k := 0 à (ω − 1) faire
   début
      C := (2k + 1 + R) div ω;
      R := (2k + 1 + R) mod ω;
      F := F + C
      écrire (F)
   fin.
```

Pour l'algorithme ainsi obtenu il n'y a plus de dépassement de capacité possible et on constate que C ne prend effectivement que des valeurs standard.

Par ailleurs à chaque étape le couple (F_k, R_k) représente exactement $f(k/\omega)$ et il n'y a pas d'accumulation d'erreurs. La fonction F déterminée est non seulement un représentant de $f(x) = x^2$ mais sa représentation entière. L'algorithme est parfait.

Rien ne m'interdit d'écrire, à présent un programme en Pascal en prenant $\omega = 10000$ et de faire tourner l'algorithme sur un micro ordinateur. Il suffit alors de placer la virgule sur les entiers affichés pour obtenir la tabulation voulue, les valeurs étant données à $1/10000$ près par défaut.

Il est possible de se ramener, grâce au code d'ordre 2, à un algorithme n'utilisant plus que des additions, des soustractions et des tests.

En effet, d'après ce qui précède :

$$\begin{cases} C_{k+1} & = [2(k+1)+1+R_{k+1}] \text{ div } \omega \\ R_{k+2} & = [2(k+1)+1+R_{k+1}] \bmod \omega \end{cases}$$

Par ailleurs

$$2k + 1 + R_k = C_k \times \omega + R_{k+1}$$

D'où en posant

$$\begin{cases} D_k & = C_{k+1} - C_k \\ dR_k & = R_{k+1} - R_k. \end{cases}$$

Il en résulte

$$\begin{cases} D_k & = (2 + R_{k+1} + dR_k) \text{ div } \omega \\ R_{k+2} & = (2 + R_{k+1} + dR_k) \text{ mod } \omega. \end{cases}$$

Remarque : On démontre que quel que soit k, $D_k \in \{-1, 0, 1\}$.

En dc la variable associée à D_k et en introduisant la variable auxiliaire t pour l'expression $2 + R_{k+1} + dR_k$ j'obtiens pour $\omega = 10000$ l'algorithme suivant :

```
ω := 10000; r0 := 0; r1 := 1; c := 0; f := 0;
pour k := 0 à 9998 faire
   début
      dr := r1 − r0;
      t := 2 + r1 + dr
      si t ⩾ 0 alors
         début
            si t < ω alors
               début
                  dc := 0;
                  r0 := r1;
                  r1 := t;
               fin
            sinon
               début
                  dc := 1;
                  r0 = r1;
                  r1 = t − ω
               fin
         fin
      sinon
         début
            dc := −1;
            r0 := r1
            r1 := ω + t
         fin;
      c := c + dc; f := f + c;
      écrire (f)
   fin.
```

Cet algorithme donne évidemment des résultats identiques à ceux obtenus avec le code d'ordre 1.

J'ai fait l'expérimentation sur un micro-ordinateur Apple II_C, les programmes étant écrits en Pascal UGSD (compilation en un P-code qui lui est interprété). La tabulation de $f(x) = x^2$ sur $[0, 1]$ avec un pas de $1/10000$ a ainsi nécessité les temps de calcul suivants :
— en entiers
. avec le code d'ordre 1 : 39 s
. avec le code d'ordre 2 : 34 s
— en virgule flottante : 52 s.

L'exemple choisi pour l'application peut paraître bien naïf. Cependant, outre son intérêt pédagogique pour l'exposé de la méthode, le calcul de x^2 joue un rôle fondamental dans le calcul sur ordinateur des puissances entières de x.

Je propose l'exercice suivant : écrire un algorithme permettant la tabulation de $g(x) = x^4$ à partir du code d'ordre 1 de $f(x) = x^2$.

5.— Quelques commentaires pour conclure

⋆ Utilisant la représentation des réels par des *classes d'équivalence* d'entiers (évitant ainsi les notions de coupure et autres...) Jacques HARTHONG a fait le pari de faire toute l'analyse avec uniquement les entiers. Ce qui a été exposé ici entre parfaitement dans le cadre de ce travail et profite ainsi d'une base théorique qui constitue à la fois un moyen de contrôle et un guide pour la prospective.

⋆ Nombre de lecteurs auront sans doute remarqué une étrange ressemblance de la démarche proposée ici avec celle de la virgule fixe (voir article de R. SEROUL dans '*L'Ouvert*' n° 48). Se pose alors la question : "*N'est-ce pas simplement de la virgule fixe déguisée ?*". Pour répondre à cette interrogation il n'y a qu'un moyen, c'est d'essayer de tenir un discours similaire dans le cadre de la virgule fixe.

Je ne tiens pas à imposer ma réponse, qui est évidemment non. Cependant je donnerai deux indices :

— dans le cadre proposé ici il est interdit de calculer la représentation du produit de deux réels comme l'a fait R. SEROUL dans son article,

— l'importance et l'interprétation de la notion de code sont immédiates dans ce qui précède. Est-ce aussi simple dans le cadre de la virgule fixe ?

ANNEXE

Introduction à l'Analyse non Standard

Les fondements

Les ensembles infinis constituaient le sujet central de la crise des fondements du début du siècle. L'ensemble des entiers en est un exemple type. D'une part il y a les entiers usuels, c'est-à-dire ceux qui, préalables à toute mathématique, font l'objet des manipulations d'usage dont la plus simple est le dénombrement : 1, puis son suivant 2, et ainsi de suite. D'autre part il y a l'ensemble $\mathbb{N}$ satisfaisant aux axiomes de PEANO. Il est clair que les entiers usuels peuvent être considérés comme des éléments de $\mathbb{N}$. Mais est-ce que tout élément de $\mathbb{N}$ est un entier usuel (ou naïf selon la terminologie de G. REEB)? Si oui pourquoi mettre parmi les axiomes un principe de récurrence?

Les mathématiciens semblent d'accord pour admettre que, d'une façon générale, il n'y a guère concordance entre les conceptions intuitives et leur formalisation que dans le cas fini. En formalisant, on introduit des objets *idéaux* dans le cas infini.

Alors que la mathématique formelle classique ne prend pas en compte l'existence même de tels objets, que la mathématique constructiviste refuse une partie du formalisme pour éviter de les considérer, l'analyse non standard leur fait jouer un rôle essentiel.

De façon précise pour fonder une théorie axiomatique de l'analyse non standard (1) :

— on considère **la théorie axiomatique des ensembles de** ZERMELO-FRAENKEL **avec l'axiome du choix**, c'est-à-dire la mathématique classique (les objets sont des ensembles et toutes les propositions s'énoncent à l'aide du prédicat binaire $\in$).

— on introduit un **prédicat unaire "standard"** (un objet pourra ainsi être qualifié de standard ou non standard) et trois axiomes supplémentaires.

Axiome 1 : Tous les éléments d'un ensemble E sont standard si et seulement si E est standard et fini.

Axiome 2 : Si $A(x, t_1, \ldots, t_n)$ est une formule de la mathématique classique alors, quels que soient $t_1, \ldots, t_n$ standard, la propriété que $A(x, t_1, \ldots, t_n)$ est vraie pour tout x standard entraîne celle que $A(x, t_1, \ldots, t_n)$ est vraie pour tout x.

Axiome 3 : Si A est un ensemble standard et P une propriété, alors il existe un ensemble standard B dont les éléments standard sont les éléments standard de A qui satisfont à P.

---

(1) La présentation de l'Analyse Non Standard proposée est une version, à peine simplifiée, de la théorie I.S.T. (International Set Theory) telle qu'elle a été introduite par E. NELSON dans le '*Bulletin de l'American Mathematical Society*' n° 83 de 1979.

Commentaires et remarques

1. On peut considérer que *standard* formalise *usuel*. Ainsi on reconnaît par l'axiome 1 l'existence d'éléments *idéaux* ou *non usuels* dans tout ensemble infini. De plus par l'axiome 2 on admet que la mathématique classique n'a pas les moyens de distinguer ces éléments *idéaux*.

2. Dans cette nouvelle théorie n'importe quelle propriété ne définit plus (comme en mathématique classique) un sous-ensemble d'un ensemble donné. Ainsi les entiers standard ne forment pas un sous ensemble de $\mathbb{N}$ (sinon on pourrait leur appliquer le principe de récurrence et tout entier serait standard). Ce défaut est en partie compensé par l'axiome 3 et se révèle paradoxalement fructueux dans de nombreuses démonstations.

3. Les objets de la mathématique classique sont les objets de l'analyse non standard. L'axiome 2 permet de montrer que ceux qui sont définis de manière unique sont standard. Ainsi $0, 1, 2, \ldots, \sqrt{2}, \ldots \pi, e, \ldots \mathbb{N}, \mathbb{Q}, \mathbb{R}, \mathbb{C}, \ldots \cos, \sin, \exp, \log, \ldots$ sont des objets standard.

4. Tous les théorèmes de la mathématique classique sont des théorème de l'analyse non standard.

5. Si la théorie de ZERMELO-FRAENKEL n'est pas contradictoire, alors la nouvelle théorie ne l'est pas non plus.

Infiniment grands et infiniment petits

Un intérêt essentiel de l'analyse non standard est de permettre une définition rigoureuse de notions d'infiniment grand et d'infiniment petit échappant aux paradoxes qui les accompagnaient traditionnellement.

Définitions:
— Un réel est dit **infiniment grand** s'il est supérieur en valeur absolue à tout réel standard.
— Un réel est dit **infiniment petit** s'il est inférieur en valeur absolue à tout réel standard strictement positif.
Le théorème suivant est immédiat.

Théorème : Il existe $\alpha, \alpha \in \mathbb{R}$ et il existe $\beta, \beta \in \mathbb{R}$ tels que α est infiniment grand et β est infiniment petit.

En effet si α est un entier non standard alors α est infiniment grand et $1/\alpha$ est infiniment petit.