Club
MATH

Centre informatique
pédagogique (CIP)
Case Postale 172
1211 GENEVE 3
Tél. (022) 318.05.30
Responsable:
Raymond Morel

*Lettre nº 18*

Lundi 11 janvier 1993 à 16 h 30.

# *Le calcul numérique avec Mathematica*

Présentation: Bernard Vuilleumier

*Les opérations de nature infinie ne peuvent pas être effectuées numériquement*

De nombreuses opérations mathématiques sont intrinsèquement infinies: les limites, certaines sommes et certains produits, les intégrales définies et même les dérivées puisqu'elles font intervenir une limite. Comme ces opérations sont de nature essentiellement infinie, il n'est pas possible de les effectuer numériquement. Le mieux que nous puissions faire est d'évaluer des grandeurs analogues finies de manière telle que, dans la plupart des situations pratiques, l'erreur soit relativement petite.

Naturellement, *Mathematica* peut parfois effectuer ces opérations symboliquement et il est en général recommandé de faire usage de cette possibilité. Voici quelques exemples:

```
Limit[(1+1/x)^x,x->Infinity]
E
```

Le module SymbolicSum.m étend les capacités de calcul formel de *Mathematica* aux sommes et aux produits:

*Autant que possible, il faut utiliser les capacités de calcul symbolique de* Mathematica

```
<<Algebra`SymbolicSum`
Sum[1/k!,{k,0,Infinity}]
E
Product[4k(1+k)/(1+2k)^2,{k,Infinity}]
Pi
--
4
```

Mais, si vous devez calculer une somme finie comportant $10^8$ termes, il vaut mieux le faire numériquement que symboliquement !

L'intérêt du concept de limite, c'est qu'il fournit une définition rigoureuse d'une situation qui ferait problème sans cette notion. La fonction sin($x$)/$x$ par exemple est indéfinie en $x$=0, mais, en prenant la limite lorsque $x$ tend vers 0, il devient possible d'attribuer une valeur à la fonction en $x$=0. Vous pouvez calculer symboliquement cette limite avec *Mathematica*:

```
Limit[Sin[x]/x,x->0]
1
```

*Toutes les fonctions numériques utilisées par un ordinateur peuvent donner des résultats faux !*

S'il était possible de traiter de tels problèmes en un nombre fini d'opérations, il serait aussi possible d'écrire des algorithmes effectuant numériquement des opérations mathématiques infinies. Malheureusement, ce n'est pas le cas et toutes les fonctions numériques utilisées par un ordinateur peuvent donner des résultats faux ! Voici un exemple que j'emprunte à Jerry Keiper. Formons la liste des points que Plot considère pour tracer une fonction:

```
liste={};
Plot[AppendTo[liste,x];0,{x,0,4},
DisplayFunction->Identity]
```

Construisons un polynôme *p* dont les zéros correspondent aux points de notre liste:

```
p=Apply[Times,liste-x];
```

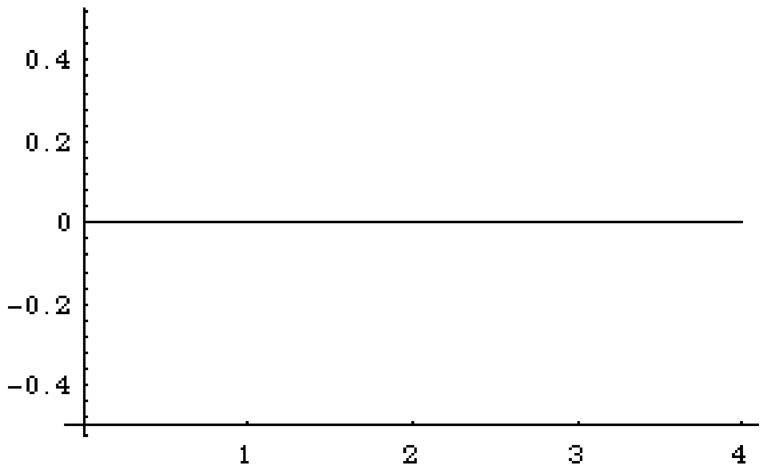
Calculons la valeur de ce polynôme en quelques autres points:

```
p/.{{x->0.2},{x->0.4},{x->0.6},{x->0.8}}
{99.7783, -10.3062, 1.16493, -0.124287}
```

Ce polynôme n'est donc pas identiquement nul. Et pourtant, voici ce qu'on obtient si on le représente graphiquement:

```
Plot[p,{x,0,4},AxesOrigin->{0,-0.5}]
```

*Des difficultés peuvent surgir chaque fois qu'un algorithme utilise une technique d'échantillonnage*




Le problème qui surgit ici est dû au fait que *Mathematica* ne considère qu'un nombre fini de points pour tracer le graphe de la fonction et qu'il suppose que le comportement de cette fonction est beaucoup plus simple qu'il ne l'est en réalité. Et cette difficulté peut se rencontrer dans n'importe quel algorithme qui utilise une technique d'échantillonnage !

# Exercices

**Exercice 1**

Intégrez numériquement les équations différentielles des exercices 1 à 9 de la lettre 17.

**Exercice 2**

Pour chaque exercice, comparez, pour quelques valeurs de *x*, les résultats obtenus par intégration numérique à ceux obtenus par intégration formelle.

*Prochaine réunion: lundi 1er février 1993 à 16 h 30.*