

Séminaire 1994

Calcul Symbolique et Formel

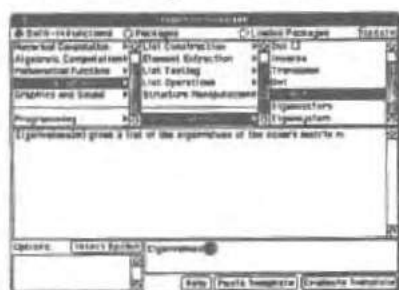
Jacqueline Zizi
Palaiseau

Les collègues de la commission nationale informatique m'ont mandatée pour proposer au séminaire annuel des réponses à un certain nombre de questions qu'ils se posaient concernant les systèmes de calcul symbolique. Il ne faut donc pas voir dans mon exposé, résumé en ces quelques pages, une présentation systématique des systèmes symboliques, pas plus qu'il ne faut chercher un ordre quelconque dans la présentation des réponses apportées. Les questions retenues étaient :

- Donner des exemples simples qui décrivent la puissance de l'outil ;
- Comment conserver trace du travail fait sur machine ? Peut-on le réutiliser, l'échanger ?
- Situer le phénomène dans le paysage informatique actuel et dans l'histoire de l'informatique ;
- La "triche". Est-il utile d'enseigner ce que la machine fait ?
- Les sujets d'examen et l'inégalité des chances ;
- Panorama des possibilités existantes ;
- L'influence sur les notions mathématiques abordées ;
- Les problèmes d'argent.

Constituées de plusieurs centaines de fonctions, voire milliers, les systèmes de calcul formel tournant sur micro-ordinateur, diffèrent par les potentialités offertes, leur cohérence interne et leur portabilité. Toutefois, tous évoluent. Partout le dialogue est prédominant et c'est l'utilisateur qui choisit sa route. Chaque fonction porte son mode d'emploi et on peut y accéder simplement. Deux modes d'accès aux informations de chaque fonction, sont généralement offerts :

- un mode d'accès direct : ? objet donne les informations attachées à un objet.
- un browser (cf Fig.) : il permet de cerner l'objet par analyse descendante des classes qui le contiennent.



Ces browsers ont été inventés par les auteurs de Smaltalk, un des premiers langages à objets qui a marqué le développement de tous les langages à objets et dont les idées sont à l'origine de l'ergonomie Mac, puis de celle de Windows dix ans après.

Les grandes classes d'objets constituent la colonne de gauche. Une classe étant choisie dans une colonne, toutes ses sous-classes ou ses éléments apparaissent dans la colonne suivante.

Les clefs incontournables du dialogue avec un système sont :

1. La syntaxe de définition et d'application d'une fonction, où le sens des délimiteurs ([et { est fondamental. Dans tous les systèmes, les composées de fonctions se lisent et se construisent comme en mathématiques, en partant du coeur et en bordant ensuite.
2. La plateforme de travail et l'interface qu'elle permet : ses caractéristiques matérielles ou conceptuelles (PC, Mac, Station) sont la pointe de l'iceberg accessible par l'utilisateur.

Les exemples ont été présentés en Mathematica sur Mac. Pour suivre la suite, il est utile de savoir que :

1. En Mathematica, les fonctions se définissent par :
 - $f[x_]$:= expression et leur application s'écrit simplement : $f[x]$.
 - Les crochets [] sont réservés à l'application des fonctions ;
 - Les accolades { } sont utilisées pour les couples, les triplets et plus généralement les n-uples ;
 - Quant aux parenthèses (), elles ne servent qu'au regroupement d'expressions.

Ainsi par exemple, on utilisera les parenthèses pour distinguer numérateur et dénominateur dans l'expression

$(2x + 1)/(x - 1)$ et on écrira $f[x]$ ou `Integrate [f[x], x]` pour désigner l'intégrale de f ou encore :

`Plot [{f[x], Log [x+10]}, {x, -2, 5}]`

pour représenter sur un même graphique les deux fonctions définies par $f[x]$ et $\text{Log } [x+10]$, sur un intervalle où x est la variable et les bornes sont -2 et 5 ;

2. les contingences matérielles sont réduites au minimum sur Mac; c'est pourquoi cette plateforme a été choisie. En théorie, avec Windows, on peut maintenant obtenir des résultats très proches, mais il arrive, dans la pratique, que la diversité du monde PC entraîne l'utilisateur vers des efforts supplémentaires de technicien, même pour des choses aussi élémentaires que "sortir" un cercle à l'imprimante et il n'est pas le lieu de les développer ici.

La puissance de l'outil. Exemples

Demandes et réponses du système constituent le dialogue.

* Numérique :

- ◆ Oui, on peut dépasser 25 !

100!

```
93326215443944152681699238856266700490715968264381621468592963\
895217599993229915608941463976156518286253697920827223758251\
1852109168640000000000000000000000000000
```

- ◆ Les 2 000 première décimales de π ?

Elles s'obtiennent en demandant `N [Pi, 2001]`. Pour économiser une page de chiffres, voici comment obtenir la 2000^{ème} décimale exacte de π et sa suivante (peut-être approchée)

```
StringTake [ToString [N [Pi, 2002]], -2]
95
```

* Manipulations algébriques :

- ◆ Développement d'expression :

```
Expand [x (x+1)^2 (x-1)^3]
      2   3   4   5   6
-x + x + 2x - 2x - x + x
```

- ◆ Résolution d'équation du 3^{ème} degré à coefficients rationnels :

```
Simplify [Solve [3 x^3 + 5 x^2 - 7 x - 1 == 0]]
      -4 - Sqrt[13]      -4 + Sqrt[13]
{{x -> 1}, {x -> -----}, {x -> -----}}
                   3                   3
```

◆ Résolution d'équation du 2^{ème} degré à coefficients algébriques dans le corps des complexes :

Solve $\{z^2 - 2 \sqrt{3} z + 4 == 0, z\}$
 $\frac{-2 I + 2 \sqrt{3}}{2}, \frac{2 I + 2 \sqrt{3}}{2}$
 $\{(z \rightarrow \frac{-2 I + 2 \sqrt{3}}{2}), (z \rightarrow \frac{2 I + 2 \sqrt{3}}{2})\}$

◆ Résolution de systèmes :

Solve $\{(2 x + 5 y + 3 z == 7,$
 $5 x - 7 y + 13 z == -1,$
 $3 x + 4 y - 11 z == 5), \{x, y, z\}\}$
 $\frac{538}{643}, \frac{634}{643}, \frac{85}{643}$
 $\{(x \rightarrow \frac{538}{643}, y \rightarrow \frac{634}{643}, z \rightarrow \frac{85}{643})\}$

"Tirer,..."

Solve $\{5 x - 7 y == -1, x\}$
 $-1 + 7 y$
 $\{(x \rightarrow \frac{-1 + 7 y}{5})\}$

"..., Porter"

$2 x + 5 y == 7 /. \#$
 $2 (-1 + 7 y)$
 $\{5 y + \frac{2 (-1 + 7 y)}{5} == 7\}$

* Dérivées, intégrales et équations différentielles

f [x_] := (x^2 - 3 x + 4) Exp [x]

◆ Valeurs exactes en quelques points

f [{-1, -1/2, 0, 1, 2, 7.5, 10}]

$\frac{8}{E}, \frac{23}{E \sqrt{E}}, 4, 2 E, 2 E^2, 68253.6, 74 E^{10}$

◆ Valeurs approchées

N [f [{-1, -1/2, 0, 1, 7.5, 10}]]

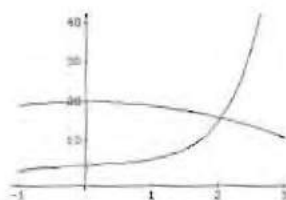
$\{2.94304, 3.48755, 4., 5.43656, 68253.6, 1.62996 \cdot 10^6\}$

◆ Dérivée

f' [x]
 $x^2 (-3 + 2 x) + E (4 - 3 x + x^2)$

◆ Représentation graphique

Plot [f [x], -x^2+20], {x, -1, 3} ;



◆ Intégrale, calcul d'aires

```
Integrate [- x^2 + 20 - f[x] , {x, 0, 1}]
```

```
86
```

```
-- - 5 E
```

```
3
```

◆ Equations différentielles : forme générale

```
DSolve [y'' [x] - y' [x] - 2 y [x] == 0, y[x], x]
```

```

      C[1]      2 x
((y[x] -> ----- + E      C[2]))
      x
      E

```

◆ Equations différentielles : avec conditions initiales

```
DSolve [{y'' [x]-3 y' [x]-y[x]==x^2,y'[0]==1,y[0]==0},y[x], x]
```

```

      35      ((3 - Sqrt[13]) x)/2
((y[x] -> -20 + (10 + -----) E      +
      Sqrt[13]

```

```

      35      ((3 - Sqrt[13]) x)/2      2
(10 - -----) E      + 6 x - x })
      Sqrt[13]

```

◆ Représentation des solutions

```
Plot [y[x] /. %, {x, -3, 3/2}] ;
```

* Nombres et fonctions de variables complexes

```
T [z- := (z+1) / (z + I)^2
```

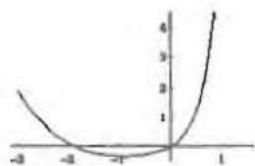
```
L'image de 1 + i et son module
```

```
{T [1+I], Abs [T [1+I]]}
```

```

      2      11 I      1
{-(---) - ----, -----}
      25      25      Sqrt[5]

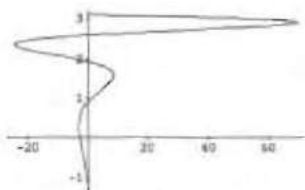
```



* Graphiques :

◆ Représentations de courbes et de surfaces

```
ParametricPlot[{Cos[3t] Exp[t], Log[t^2]},
```



◆ Figures géométriques

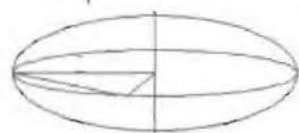
```
A = {1, 2} ; B = {-4, 2} ;
```

```
Origine = {0, 0} ;
```

```
Show [Graphics [(Circle [A, 5], Circle [A,
```

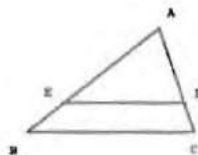
```
(5,2)], Line [{A, B, Origine, A} ,
```

```
Line [{A, {1,-6}, {1,9}}]] ;
```



* Tous les domaines peuvent être abordés. Voici par exemple une façon d'aborder la propriété de Thalès.

◆ Définition de la fonction :

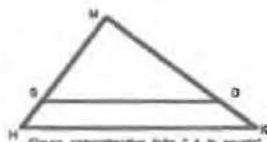


$$\text{thales}[\text{triangle}[A, B, C], E, F]^{\wedge} := m[A, E]/m[A, B] == m[A, F]/m[A, C] == m[E, F]/m[B, C]$$

◆ Applications à un exemple :

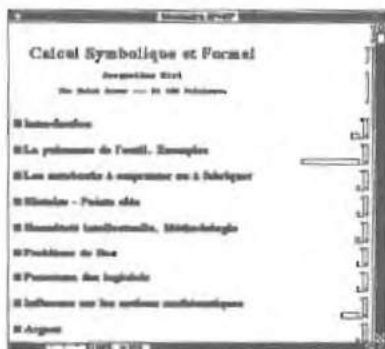
$$\begin{array}{ccc} \text{thales}[\text{triangle}[M, N, P], G, H] \\ m[M, G] & m[M, H] & m[G, H] \\ \hline m[M, N] & m[M, P] & m[N, P] \end{array}$$

◆ Application à un autre exemple, suivi d'une exploitation :



$$\begin{array}{ccc} m[M, B]=4 ; \\ \text{thales}[\text{triangle}[M, H, K], B, D] \\ 4 & m[M, D] & m[B, D] \\ \hline m[m, H] & m[M, K] & m[H, K] \\ m[M, H]=5 ; m(H, K)=17 ; \\ \text{Solve}[4/m[M, H] == m[B, D]/m[H, K], m[B, D]] \\ 68 \\ \{ \{ m[B, D] \rightarrow - \} \} \\ 5 \end{array}$$

Les notebooks à emprunter ou à fabriquer



Mathematica est aussi un traitement de textes. Les échanges avec *Mathematica* peuvent être complétés, modifiés, échangés et réutilisés à loisir. Difficile à rendre sur papier, mais bien pratique pour les exposés, un notebook, s'ouvre à différents niveaux. C'est une fiche ou un polycopié électronique qui regroupe texte, demandes (ou entrées), réponses (ou sorties), dessins et images rapportées. Ainsi, par exemple, on peut toujours avoir le plan sous la main, ou plutôt sous la souris...

Voici une copie d'écran du plan de ce texte. Les icônes à droites du texte portent les caractéristiques des cellules. Quant aux rectangles, ils indiquent la taille du contenu sous-jacent et en sont également des boutons d'accès.

Histoire - Points clés

De même qu'il y a plusieurs théories mathématiques et différentes spécialités, il y a différentes informatiques. Ce qui suit et qui est très schématique, ne concerne qu'une partie de l'informatique symbolique.

Aussi vieille que Fortran et grandie en osmose avec la théorie mathématique du lambda-calcul, l'informatique applicative, ou fonctionnelle, en est le ciment. Elle a pris depuis différentes formes : Logo dans le monde de l'enseignement, Lisp¹ dans celui de l'intelligence artificielle et du calcul formel, Autocad dans le monde industriel.

Dans ce monde, tout calcul s'exprime par l'application d'une fonction. Toutes les applications de fonctions ont la même forme dans laquelle le symbole représente la fonction. Cette représentation unique des fonctions, quelque soit l'espace sur lequel elles travaillent, se gère informatiquement très bien et permet la composition des fonctions par un chaînage de liens. Elle libère l'utilisateur des réservations de place mémoire et autres réjouissances qui sont dynamiques, optimisées et automatisées.

Honnêteté intellectuelle

Avoir une réponse immédiate à des questions qui demandaient, il n'y a pas si longtemps, des constructions ou des calculs plus ou moins importants auparavant est-il souhaitable ? N'est-ce pas tricher ? Peut-on accepter des résultats dont on ne sait pas d'où ils sortent ? Faut-il laisser de tels systèmes dans les mains des élèves et pour faire quoi ? Toutes ces questions légitimes méritent réflexion et un parallèle avec les pratiques et les développements des mathématiques. On ne peut pas les aborder sans pénétrer dans ce monde, plus ressemblant à un TGV à prendre en marche, qu'à un outil que l'on domine, sans savoir qu'il est régi par les mêmes méthodologies que les mathématiques et que la rigueur et l'honnêteté intellectuelles en sont les seuls garants.

Il y a lieu de distinguer :

- l'utilisation de l'outil, qui comme les mathématiques par les non mathématiciens, peut donner des résultats satisfaisants, surprenants, voire décevants s'il est mal appliqué ;
- la programmation, qui peut, comme la pratique des mathématiques, être perçue comme conceptuellement difficile, mais représente l'aspect formateur. C'est d'autant plus vrai que la programmation colle à la réflexion mathématique, loin des contingences de la machine.

1) invention de Mac Carthy pour faire la dérivation formelle en 1960 et à l'origine de tous ces développements

Problème de Bac

Il n'est pas de sujet de Bac qui ne soit abordable avec un système. Suivant le sujet, son aptitude à entrer dans le jeu et les moyens dont on dispose, matériels et logiciels, tout ou partie du sujet peut être traité. Les problèmes d'analyse traditionnelle sont particulièrement faciles à résoudre : les réponses aux questions clefs demandent souvent moins de temps et de place à l'utilisateur que n'en requiert le texte lui-même.

Panorama des logiciels (voir page suivante)

Une des principales caractéristiques des systèmes est leur évolution constante. Les renseignements ci-dessous sont donc à prendre comme le flash d'un instant et sont sujets à modifications.

Influence sur les notions mathématiques

De même que ce siècle a vu l'essor des mathématiques étroitement lié au développement de la physique, il est probable que le développement des mathématiques ne puisse résister plus longtemps à l'infiltration de l'informatique dans tous les domaines, et ceci, tout en influençant son développement.

Ce n'est pas tant que les mathématiciens comptent sur les machines pour résoudre les problèmes qui les intéressent, mais plutôt :

- leur curiosité ne saura longtemps résister à la tentation d'utiliser ces merveilleux outils d'investigation (et c'est déjà le cas pour les fractales ou les systèmes dynamiques)
- les jeunes informaticiens "purs" qui s'intéressent à leur discipline en tant que science ne manquent pas d'aller voir et de s'interroger sur les fondements des sciences développées par leurs aînés, à savoir : la logique et les mathématiques.

Pour témoin, j'en veux les ouvrages mathématiques récents sur les automates, les réseaux de Petri, la théorie des codes, le renouveau de la logique avec des développements nouveaux (comme les logiques modales ou la logique floue) et j'en passe. Autre témoin : alors que l'enseignement de la logique n'entre pas dans le cursus des mathématiciens, il est très difficile d'y échapper lorsqu'on étudie l'informatique et plus particulièrement si l'on souhaite se spécialiser dans les disciplines qui se situent à la pointe du développement.

Il se trouve que le vocabulaire des mathématiques ensemblistes, le concept de fonction, ne suffisent pas aux informaticiens, ne serait-ce que pour exprimer les problèmes qui se posent à eux. Il n'est donc pas complète

	Dérive	Maple	Mathematica
Plateforme	PC	PC ou Mac	PC ou Mac
traitement textes	non	partiel	oui
hypertexte	non	documentation	partiel
<i>programmation</i> procédurale applicative	non faible	oui possible mais Pb de cohérence	oui bonne
par objet filtrage réécriture	non non non	pas vraiment très faible non	couche minimale sophistiqué base du système
<i>graphiques</i> mobilité animation programmation	non non non	à la souris oui non	paramétré oui oui
<i>Prix pour un étudiant</i>	1 550 FF TTC (prix licence mixte)	783 FF TTC	1 400 FF HT
<i>Livres avec la version étudiant</i>	"L'assistant en Mathématiques" (260 pages)	"First Leaves" (253 pages) + "Release 3 Notes" + "Getting Started"	"Mathematica, a system for Doing Mathematics" (961 pages) + Doc Interface 2 à 300 pages
<i>Place sur disque : mémoire : à l'aise :</i>	minimum: 512 k 640 k	Release 3 : 13,5 Mo minimum : 4 Mo 6 Mo	version 2.2 : 6,6 Mo minimum : 4 Mo 6 Mo
<i>Version Française du langage</i>	OUI	OUI	OUI avec diction- naire "Math Diderot"
<i>Doc. papier Franç</i>	OUI	NON	En cours
<i>Doc en ligne française</i>	OUI	NON	OUI avec "Help Stack"

ment absurde d'envisager un développement et une extension de théories plus générales que celles des ensembles, en particulier la théorie des graphes, des catégories et des logistiques non classiques et une perception plus élargie du concept de relation souvent réduit à des relations binaires ou à de simples fonctions.

L'effet miroir ou ping pong sur l'informatique se fait déjà ressentir, tant il est vrai que la programmation, autrefois assimilée à l'algorithmique, s'ouvre largement vers d'autres styles de programmation :

- La programmation applicative émerge des centres universitaires et fournit des produits solides : Logo, Autocad et les systèmes de calcul symbolique. Lambda Calcul et Fonctions en sont les mots clés.
- La programmation à objets a envahi tous les domaines, des bases de données aux systèmes d'exploitation. Classes, relations entre objets et méthodes définies au niveau des classes sont déterminants.
- La substitution et l'unification sont incontournables dans le royaume des systèmes experts.

Mathematica permet tous ces styles de programmation. Programmer, c'est alors simplement analyser sa pensée mathématique et choisir le style de programmation correspondant.

Par exemple, supposons que l'on souhaite avoir une idée de "la valeur de la dérivée seconde d'une fonction, là où la dérivée première s'annule". Que s'agit-il de faire ?

- trouver les racines de la fonction ? C'est appliquer une fonction pour résoudre ; cette fonction s'appelle `Solve` en Mathematica ;
- calculer la dérivée seconde ? C'est composer l'opérateur dérivée deux fois de suite
- remplacer dans l'expression de la dérivée seconde le x par les valeurs trouvées ? C'est faire opérer une fonction de substitution ;
- chercher une valeur approchée des valeurs exactes trouvées ? C'est faire opérer une fonction d'approximation sur le résultat précédent. Cette analyse faite, la réponse est immédiate et s'écrit pour une fonction f quelconque :

```
N[D[f[x], {x, 2}]/. Solve[f'[x] == 0]]
```

Autre exemple, symbolique si j'ose dire, c'est celui du fondement des nombres entiers vu d'un point de vue ensembliste. La première construction est celle des entiers à partir de l'ensemble vide. On dénomme le vide par 0, puis 1 désigne l'ensemble constitué d'un élément, vide ; 2 est alors l'ensemble constitué de 0 et 1, c'est-à-dire vide et 1 etc. Ceci se programme immédiatement en Mathematica (ou en Lisp) de façon récursive :

```
entier [n_] :=
  If [n == 0,
    List [{"0"}],
    Prepend [entier [n - 1], entier [n - 1]]]
```

Les premiers entiers se demandent alors ainsi :

```
Do[Print [i-1, "->", Part[Reverse[entier[i]], i]], {i, 1, 5}]
```

Ce qui donne :

```
0->{}
1->{{}}
2->{{{}}, {} }
3->{{{{}}, {}}, {{}}, {} }
4->{ {{{{}}, {}}, {{}}, {}}, {{{{}}, {}}, {{}}, {} } }
```

Le dernier exemple, est une simple remarque sur la logique sous-jacente. De même que la clef du contrôle en programmation procédurale est le "if then else", dans nos classes, une propriété mathématique est "vraie ou fausse". Pourtant, il peut nous arriver de répondre à un élève : "je ne sais pas" et de la même façon qu'il existe des mathématiciens qui travaillent en acceptant le raisonnement par l'absurde, il en existe également qui "font" des mathématiques sans ce principe. Les systèmes symboliques et de calcul formel, disponibles sur micro-ordinateurs, travaillent en logique trivaluée, ce qui permet d'envisager qu'une propriété est vraie, fausse ou autre. Par exemple, le signe d'un nombre peut se programmer ainsi :

```
signe [a_] := If[a>0,plus, moins, "OuhLa, je ne sais pas ! "]
```

Voici quelques exemples d'application de cette fonction et les résultats correspondants :

```
signe [2]
plus
signe [-9]
moins
signe [bonjour]
OuhLa, je ne sais pas !
```

Argent. L'argent est une composante nouvelle liée à l'introduction de l'informatique. Si elle ne surprend pas les physiciens et les biologistes, cette présence mérite réflexion car jamais l'argent n'entraîne en ligne de compte dans nos choix. Si toutefois cela devrait remettre en cause l'ordre de valeurs qui a toujours guidé les mathématiciens et enseignants de mathématiques, alors il faudrait faire un sitting de réflexion et prôner le retour à la craie et au chiffon. L'inégalité des candidats aux examens est un premier symptôme et les *mesures ministérielles* : le bac à formulaire et le sujet national sont des cautères sur une jambe de bois.