

V. Ripoll  
vripoll@dma.ens.fr

## MK1 "Calcul formel" Maple, Premier semestre 2007

N.B.: ces TPs sont très largement inspirés des TPs écrits par Cécile Armana.

### Qu'est-ce que le "calcul formel" ?

Maple est un logiciel de calcul formel : il est capable de manipuler des nombres, mais aussi des symboles représentant des nombres ou des objets mathématiques plus compliqués (fonctions, équations,...). Il est beaucoup plus puissant qu'une calculatrice ordinaire, puisqu'il peut par exemple dériver une fonction, résoudre une équation,... de façon "abstraite", c'est-à-dire en manipulant des expressions symboliques ne contenant pas que des paramètres numériques.

Ce genre d'outil est très utilisé dans l'enseignement, les centres de recherche et l'industrie (on peut citer d'autres logiciels de calcul formel : Mathematica, MuPAD...).

Ce logiciel a été créé en 1981 (et a beaucoup évolué depuis) par l'université de Waterloo, Canada. Le mot *Maple* signifie *érable* en anglais, et est également un acronyme pour *Mathematical PLEasure*.

### Le but des TP MK1

Il s'agit d'apprendre à utiliser Maple en illustrant votre programme de mathématiques par des exemples et des exercices "à faire avec Maple". N'hésitez pas à tester vos propres exemples.

Bien entendu, **cela suppose que vous connaissiez votre cours de maths !** Vous êtes invités à apporter votre cours avec vous lors des séances de TP.

### Déroulement des TP

Le TP, d'une durée de deux heures, a lieu en salle K du SCRIPT (Halle aux farines, secteur C, 4ième étage). *Merci d'arriver à l'heure !*

### Contrôle des connaissances

Les séances de TP seront sur 13 semaines, de la semaine du 17 septembre à la semaine du 17 décembre (interruption la semaine du 29 octobre). Le dernier TP sera vraisemblablement une séance de révision.

Pendant la session d'examens de janvier 2008, il y aura un examen sur machine

d'une durée de deux heures. La note finale tiendra compte de la note à cet examen, mais aussi de l'**assiduité-participation en cours**, ainsi que d'un DM (devoir maison) et d'un partiel (début novembre). Les modalités précises vous seront indiquées ultérieurement.

Rappel : l'année n'est pas validée lorsqu'il y a une absence injustifiée à un examen.

### Bibliographie

Parmi les livres de tous niveaux qui existent sur Maple, je vous recommande "*Maple sugar*", de Guy Le Bris (éd. Cassini) dont l'esprit correspond bien à ce que nous ferons en TP.

### Et surtout, n'oubliez pas de vous (et de me) poser des questions !

#### 1. Présentation de l'interface

\* La feuille blanche est appelée *feuille de calcul*. C'est là que vous donnez à Maple des commandes, qu'il les exécute et vous affiche le résultat. C'est l'analogue de l'écran d'une calculatrice ordinaire (mais on peut y afficher bien plus de choses !)

\* La feuille de calcul commence par une *invite* (ou *prompt* : symbole  $>$ ) : cela signifie que Maple est prêt à recevoir des commandes de l'utilisateur.

\* Si on tape alors une commande comme 1+2 et qu'on valide en appuyant sur la touche "Entrée" :

$>$  1+2

Warning, inserted missing semicolon at end of statement  
3

(1)

...on reçoit un message d'erreur.

**Règle importante n°1: toute commande doit être suivie d'un caractère terminateur, la plupart du temps ; (point-virgule), parfois : (deux-points).**

$>$  1+2;

(2)

Ouf !

Si on utilise deux-points à la place du point-virgule, Maple effectue la commande mais n'affiche pas le résultat. Cela peut être pratique quand le résultat donné par Maple prend trop de place à afficher, ou si la valeur exacte ne nous intéresse pas. Remarque: la ligne de commande peut être complétée par un commentaire précédé du symbole #.

$>$  80! : #je veux voir le résultat

715694570462638022948115337231865321655846573423657525771094\ (3)  
45058227039255480148842668944867280814080000000000000000  
0000

$>$  80! : #je ne veux pas voir le résultat

\* On peut donner plusieurs commandes à Maple sur la même ligne :

> **2^5;11\*7;**

32  
77

(4)

\* Pour enregistrer sa feuille de calcul, on utilise le menu "File", "Save" ou "Save as". Le fichier a une extension en .mws.

\* Pour ouvrir une feuille enregistrée, on utilise le menu "File", "Open". Si on souhaite travailler à nouveau dessus, il faut alors re-exécuter toutes les commandes de la feuille, soit en les validant une à une avec la touche "Entrée", soit en utilisant le menu Edit, Execute, Worksheet.

\* Pour imprimer une feuille de calcul, on utilise le menu "File", "Print". Je vous demanderai de n'imprimer si besoin qu'à la fin du TP.

\* Ces actions sont également faisables en utilisant les icônes de raccourci.

## 2. L'affectation de variables

Il est très pratique de donner des noms à des résultats antérieurs, notamment pour pouvoir les réutiliser par la suite : c'est l'*affectation*. En voici un exemple :

> **produit:=6;**

*produit*:= 720

(5)

A gauche du signe :=, on entre le nom de la variable et à droite la valeur affectée. On peut vérifier l'affectation par :

> **produit;**

720

(6)

et l'utiliser pour faire de nouveaux calculs :

> **produit/5!**

6

(7)

Le nom de variable ne doit pas comporter de signes de ponctuation, d'espace ni de caractères spéciaux (par ex. +, \*, #, %, @). On peut utiliser des majuscules et des minuscules (attention ! *Maple différencie les deux !*). La procédure d'affectation est très générale. Ici, on a donné un nom de variable à un nombre entier (6) mais on peut nommer également des nombres rationnels, décimaux, complexes, des fonctions, des matrices,...

Pour réinitialiser (désaffecter) la variable *produit* et faire en sorte qu'elle ne contienne plus la valeur 6!, on effectue l'une ou l'autre des commandes suivantes:

> **produit:=!produit;**

*produit*:= *produit*

(8)

> **unassign('produit');**

On vérifie que la variable "produit" est bien désaffectée :

> **produit;**

*produit*

(9)

Si on veut réinitialiser toutes les variables, on utilise la commande *restart*.

Le signe % permet de rappeler (et utiliser) le dernier résultat calculé ; il existe aussi %% et %%% : tester sur des exemples.

## 3. L'ordre des commandes

Il est très important de comprendre que le comportement de Maple dépend de l'ordre chronologique de validation des commandes, et non de l'ordre d'apparition sur la feuille de calcul.

Dans une feuille de calcul, rien ne vous empêche de modifier une commande entrée précédemment : il suffit pour cela de remonter à la ligne qui vous intéresse (au clavier ou à la souris), de modifier la commande et de valider avec "Entrée". Cependant, attention à cette manipulation : il vaut mieux valider les commandes dans l'ordre dans lequel elles apparaissent dans la feuille de calcul, sous peine d'arriver à des choses bizarres... Par exemple :

> **a:=3;**

*a*:=3

(10)

> **b:=a/2;**

*b*:= $\frac{3}{2}$

(11)

Si en remontant, je décide de modifier *a*=3 en *a*=2, mais que j'oublie de valider avec "Entrée" la ligne suivante, *b* vaudra toujours  $\frac{3}{2}$  et ne sera plus égal à *a*/2 !

**Règle importante n° 2 : si vous modifiez une commande précédente dans la feuille, faites re-exécuter les lignes suivantes à Maple avec la touche "Entrée".**

**Règle importante n° 3 : il vaut mieux commencer une feuille de calcul par la commande :**

> **restart;**

afin d'être certain que toutes les variables sont désaffectées.

## 4. L'aide de Maple

Maple est un logiciel très riche et il n'est pas question d'en connaître toutes les commandes et leurs syntaxes. L'aide de Maple est très utile pour retrouver ce genre d'informations, et il est essentiel de savoir l'utiliser.

\* Si on veut des informations sur une commande dont on connaît le nom, il suffit de taper un ? suivi du nom de la commande (inutile de faire suivre d'un ;)

> **?isprime**

\* Si on ne connaît pas le nom de la commande, on utilise le menu Help, Topic search pour une recherche thématique.

## 5. Calculs sur les nombres entiers

Maple fait automatiquement des calculs exacts sur de très grands entiers. Les opérations usuelles sont +, -, \*, /.

Dans Maple, les différents objets ont un *type*. On peut demander le type d'un objet par la commande *whattype*. Par exemple, le type d'un entier est *integer*.

> **whattype(15!);**

*integer*

(12)

## 6. Calculs sur les nombres réels

La première façon d'*approcher* un nombre réel grâce à Maple est d'utiliser le point décimal (.). (c'est l'équivalent anglo-saxon de notre virgule décimale). Maple calcule alors des valeurs *approchées*, avec un nombre de chiffres significatifs fixé (par défaut, 10).

> **300/45;**  
$$\frac{20}{3}$$
 (13)

Pour Maple, le nombre précédent n'est pas un nombre réel, c'est un nombre rationnel (une fraction de deux nombres entiers) : d'ailleurs, il nous a proposé spontanément une simplification. Pour le lui faire comprendre comme un nombre réel (approché), on peut faire :

> **300./45;**  
6.66666666667 (14)

ou bien :  
> **evalf(300/45);**  
6.66666666667 (15)

Le type d'un réel est *float* (nombre flottant).  
> **whattype(300./45);**  
*float* (16)

Maple connaît certains réels comme *e* :  
> **exp(1); evalf(exp(1));**  
*e*  
2.718281828 (17)

et *pi* :  
> **Pi;**  
**evalf(Pi);**  
*π*  
3.141592654 (18)

(attention, **P majuscule !**)  
On peut choisir le nombre de chiffres significatifs par :  
> **evalf(Pi, 20);**  
3.1415926535897932385 (19)

On peut aussi utiliser la variable **Digits** (voir l'aide).

Voici quelques fonctions prédéfinies de Maple. Utilisez l'aide pour en savoir plus :  
*exp*  
*ln* ou *log*  
*sqrt*  
*sin*, *cos*, *tan*  
*abs*

*trunc*, *floor*, *ceil*  
*max(x1, x2, ..., xn)*, *min(x1, ..., xn)*

## 7. Calculs sur les nombres complexes

Pour définir un nombre complexe, on utilise le nombre imaginaire *i*, que Maple représente par **I (attention, i majuscule !)**.

> **z:=3+4\*I;**  
$$3 + 4I$$
 (20)

Pour obtenir ses parties réelles et imaginaires :  
> **Re(z); Im(z);**  
3  
4 (21)

> **z\*(1+sqrt(2)\*I);**  
$$(3 + 4I) (1 + I\sqrt{2})$$
 (22)  
Pour forcer Maple à écrire le nombre sous forme cartésienne (partie réelle + *i* \* partie imaginaire), on utilise *evalc* :

> **evalc(%);**  
$$3 - 4\sqrt{2} + I (4 + 3\sqrt{2})$$
 (23)  
Les commandes pour obtenir le nombre complexe conjugué, le module et l'argument sont : *conjugate*, *abs*, *argument*. Le module est un réel positif. L'argument (donné par Maple) est dans  $]-\pi; \pi]$ .

> **conjugate(z); abs(z); argument(z);**  
$$3 - 4I$$
 (24)  
$$\arctan\left(\frac{4}{3}\right)$$

Pour définir un nombre complexe sous forme trigonométrique *r. exp(i theta)*, où *r* est le module et *theta* l'argument, on utilise la commande *polar* (le premier argument est *r*, le deuxième *theta*) :

> **polar(3, Pi/6);**  
$$\text{polar}\left(3, \frac{1}{6} \pi\right)$$
 (25)

> **evalc(%);**  
$$\frac{3}{2}\sqrt{3} + \frac{3}{2}I$$
 (26)

Pour passer de l'écriture cartésienne (*a+b\*I*) à l'écriture polaire, on utilise encore *polar* mais la syntaxe est différente :

> **polar(1+I);**  
$$\text{polar}\left(\sqrt{2}, \frac{1}{4} \pi\right)$$
 (27)

## 8. Les fonctions simplify et assume

Lorsqu'un résultat obtenu par calcul algébrique n'a pas la forme voulue, certaines fonctions de Maple permettent d'y remédier. La fonction *simplify* est

l'une d'entre elles. Elle est très riche, aussi nous ne regarderons que des exemples.

```
> 4^(1/2)+4;
```

$$\sqrt{4} + 4 \quad (28)$$

```
> simplify(%);
```

$$6 \quad (29)$$

```
> (sin(x))^4-(cos(x))^4;
```

$$\sin(x)^4 - \cos(x)^4 \quad (30)$$

```
> simplify(%);
```

$$1 - 2 \cos(x)^2 \quad (31)$$

Un exemple important :

```
> y:=sqrt(x^2);
```

$$\sqrt{x^2} \quad (32)$$

```
> simplify(y);
```

$$\operatorname{csgn}(x) x \quad (33)$$

Quelle est la fonction *csgn* ? Le résultat est-il correct ?

Par défaut, Maple ne sait rien de la variable non affectée *x* et la considère comme un nombre complexe. Si on suppose (*assume* en anglais) que *x* est positif, on peut encore simplifier l'expression :

```
> simplify(y, assume=positive);
```

$$x \quad (34)$$

La supposition est temporaire (le temps que la commande soit effectuée). Si on souhaite qu'elle soit permanente, c'est-à-dire jusqu'à la fin de la session ou jusqu'à ce que la variable soit réinitialisée, on utilise *assume* avec une syntaxe différente :

```
> assume(x, positive);
```

```
> simplify(y);
```

$$x \sim \quad (35)$$

Le  $\sim$  (tilde) rappelle qu'une hypothèse a été faite sur la variable *x*.