

## Pour démarrer avec MAPLE

MAPLE est un logiciel dit de **calcul formel**<sup>1</sup>. Il est capable de faire des calculs exacts avec des « lettres ». Par exemple, avec l'instruction `expand((a + b)^5);`, il développe l'expression  $(a + b)^5$  sans se soucier de savoir ce que représentent les variables  $a$  et  $b$ . Mais il peut faire mieux, par exemple taper

```
for k from 1 to 7 do expand((a + b)^k) od ; .
```

## 1 Saisie des commandes

On saisit les commandes après le symbole « > » dit prompt en anglais ou invite de commande en français. Chaque commande Maple doit se terminer par un « ; » ou un « : » (avec les « : », le calcul est effectué, **mais il n'est pas affiché**, ce qui parfois est bien utile). En appuyant sur Entrée, Maple nous renvoie sa réponse, dite *output* à opposer à *input*. Observer d'ailleurs le message renvoyé lorsque vous oubliez le « ; »

```
Exemples : 3*5; 5^3; 25!; solve(x^2+x+1=0,x);
```

Pour écrire un texte de plusieurs lignes avec la même invite de commandes, on tape la combinaison « Shift et entrée », cela sera utile pour écrire des procédures (des programmes).

## 2 Fonctions et constantes usuelles

- Maple connaît les **fonctions usuelles** : `exp()`, `ln()`, `cos()`, `sin()`, `tan()`, `sqrt()` (racine carrée dit *square root* en anglais). Il connaît aussi les **nombre classiques** : `Pi` pour  $\pi$  (attention à la majuscule), `exp(1)` pour  $e$ , `I` pour le nombre complexe  $i$ , `infinity` pour  $\infty$ . Attention Maple distingue les majuscules des minuscules. Ainsi la commande `PI` renvoie la lettre grecque  $\Pi$  et non pas le nombre  $\pi$ .

Exemples : `exp(ln(x)); cos(Pi); cos(PI); exp(I*Pi)` (attention, le symbole `*` est obligatoire). Enfin `sum(1/k^2,k=1..infinity); int(exp(-x),x=0..infinity);`

- **Quelques fonctions « arithmétiques »** : `floor` (partie entière), `iquo` (quotient d'une division euclidienne, *integer quotient*), `irem` (reste d'une division euclidienne, *integer remainder*), `ifactor` (*integer factorization*).

Exemple : `irem(3^7,7);`

- On peut aussi créer **ses propres fonctions** : par exemple

```
f :=x->1/(1+x^2); f(0); f(1); seq(f(k),k=0..9); plot(f);
```

Exercice : créer une fonction **norme** qui renvoie la norme d'un vecteur du plan de coordonnées  $(x, y)$  en démarrant par `g :=(x,y)->...`

- Comment faire si l'on ne souvient pas du nom d'une fonction ou de son utilisation ? On va voir **l'aide de Maple** (attention, elle est en anglais). On peut y accéder à l'aide du menu `help` ou en tapant ? suivi du nom de l'instruction. Par exemple `?irem`.

## 3 Évaluation numérique

Par défaut Maple fait des calculs exacts. Taper successivement `4/3+3/4; 100!; sqrt(4+2*sqrt(3));`

Mais on peut l'obliger à faire du calcul approché avec la fonction `evalf` (*evaluate float*). Taper `evalf(200/3); evalf(200/3,15);` cette dernière instruction donnant 15 chiffres significatifs.

## 4 Variables et affectations

### 4.1 Notion de variable et d'affectation

On appelle **variable** tout mot constitué d'une ou plusieurs lettres et ou chiffres (à l'exception de mots protégés tels que `Pi...`). Il peut aussi contenir le symbole souligné (*underline*). Par exemple, `2toto` et `coco34` sont deux noms de variable.

1. Il y a aussi XCAS qui est un logiciel libre.

On peut affecter des valeurs à ces variables, la variable peut être imaginée comme une case mémoire dans laquelle on stocke de l'information. Pour cela on utilise l'opérateur<sup>2</sup> « := ». La variable est alors automatiquement remplacée par sa valeur.

Par exemple `x :=2; bip :=3*x+5; 4*bip;y+x` Ici `x` et `bip` sont deux variables affectés tandis que `y` est une **variable libre**. Par défaut, une variable (hormis certaines lettres comme **I, D...**) est libre.

#### Remarques :

- l'ordre chronologique d'affectation est important. Comparer les deux suites d'instructions suivantes :

$\begin{aligned} n &\leftarrow 2; \\ u &\leftarrow n^4; \\ n &\leftarrow n + 1; \\ n; u; \end{aligned}$	$\begin{aligned} n &\leftarrow 2; \\ n &\leftarrow n + 1; \\ u &\leftarrow n^4; \\ n; u; \end{aligned}$
---	---

- Attention à ne pas confondre le symbole d'affectation « := » au symbole d'égalité<sup>3</sup> « = ».
- Pour **désaffecter une variable**, on tape `x :='x'` ; Pour désaffecter l'ensemble de toutes les variables, on pourra taper `restart` ; qui réinitialise le système. On peut vérifier l'affectation ou non de la variable `bip` à l'aide `whattype(bip)`, ou de `about(bip)`
- Un exemple pratique : **la création d'un compteur**. L'instruction `x :=0; x :=x+1; a` initialisé `x` à la valeur 0 puis lui a ajouté 1. On dit que l'on a incrémenté la variable `x` de 1.

## 4.2 Types

Du point de vue de la machine, une variable est une case mémoire avec une adresse qui peut contenir une certaine quantité d'informations. Par exemple, si l'on veut stocker le nombre 5 ou un nombre entier à 100 chiffres, la quantité de mémoire nécessaire à leur stockage n'est pas la même. En langage C, par exemple, avant d'affecter une variable, il est nécessaire de préciser de quelle type de variables, il s'agit afin de réserver dans la RAM la quantité de mémoire nécessaire. Par exemple, un entier peut être codé avec une case mémoire de 16 bits ou 64 bits pour un entier long...

**Attention :** pour Maple, les nombres 4 et 4.0 ont la même valeur mais sont d'un **type** différent. L'objet 4 est de type **integer** (un entier) tandis que 4.0 est de type **float** (un nombre flottant). Observer les réponses aux commandes suivantes : `4+1/2` ; `4.0+1/2` ;

Pour connaître le type d'un objet, on peut utiliser la fonction `whattype` ou `about`. Exemple : `whattype(4.0)` ;

Il y a d'autres types d'objets en Maple : les **booléens** (boolean) qui s'écrivent `true` or `false`, les **chaînes de caractères** (string) qui s'écrivent entre apostrophes ou entre back quotes, par exemple `'to'` ou `‘to‘`, ce qui permet d'insérer des espaces. On peut concaténer deux chaînes de caractères en une seule à l'aide de l'opérateur `cat`. Nous verrons par la suite d'autres types tels que les listes, les séquences, les tableaux...

## 4.3 Affectation de propriétés

Une des spécificités de Maple est de savoir faire du *calcul symbolique* avec des **variables libres**. C'est le principe d'un logiciel de calcul formel. Par exemple `sinju+sinju` ;.

Et combien vaut  $\cos(n\pi)$  ? Demander confirmation à Maple...Il semble que Maple ne sache pas simplifier...C'est normal, cette simplification n'est possible que si la variable `n` est un entier !! Il faut donc « typer » notre variable libre.

La fonction `assume()` permet d'attribuer une propriété à une variable libre. Lorsqu'une variable libre est munie d'une propriété, elle apparaît suivant d'un tilde (  $\sim$  ) à l'écran. On peut utiliser `about` pour obtenir de l'information sur la variable.

Exemples :

- `assume(x>0)` affirme que `x` est un réel strictement positif.
- `assume(x, real)` affirme que `x` est un réel (Maple connaît bien sûr les nombres complexes).
- `assume(x, integer)` affirme que `x` est un entier relatif.
- `assume(x, natural)` affirme que `x` est un entier naturel.

2. Dans des langages comme le C ou Python, le symbole d'affectation est « = » et de façon plus universelle, la notation  $u \leftarrow 2$  signifie que j'affecte de la valeur 2 la variable `u`.

3. Dans des langages comme le C ou Python, le symbole d'égalité est « == ».

- `assume(x>0, x<=1)` affirme que  $x \in ]0, 1]$
- Exercices :
- calculer  $\cos(n\pi)$  pour  $n$  entier.
- simplifier  $\ln(x^2)$  (attention au piège).

## 5 Manipulation de nombres complexes

La fonction essentielle est `evalc(z)` qui donne l'écriture algébrique du nombre complexe  $z$ . Il y a aussi `conjugate`, `abs` et `argument`. On rappelle que  $I$  désigne le nombre complexe  $i$ .

Exercice : exprimer les parties réelles et imaginaires de  $1/z$  en fonction des parties réelles de  $z$ .

## 6 Exercices complémentaires

### Exercice 1 (Évaluation numérique)

1. Quelle est la centième décimale de  $\pi$  ?
2. Définir la variable  $x$  égale à  $\left(\frac{\ln(640320^3+744)}{\pi}\right)^2$  puis dire si c'est un entier.
3. Le nombre  $x = (\sqrt{2} + 1)^2 - 2\sqrt{2}$  est-il un entier ?

**Exercice 2** Le produit de quatre entiers consécutifs plus 1 est-il un carré ?

**Exercice 3 (Un bug de Maple ?)** On pose  $a = \sqrt[3]{20 + 14\sqrt{2}} + \sqrt[3]{20 - 14\sqrt{2}}$ .

1. Évaluer  $a$  à plusieurs niveaux de décimales.
2. Est-il un entier ? Répondre à cette question avec l'une de ces instructions
  - `is(x=4)` ;
  - `is(x, integer)` ;
  - `is(floor(x)=x)` ;
 Commenter.

**Exercice 4 (Plus fort que Maple)** Le nombre de chiffres de l'écriture décimale d'un entier  $n > 0$  s'obtient par la formule

$$E(\log n) + 1 \quad (\text{avec } \log(n) = \frac{\ln n}{\ln 10}).$$

1. Déterminer le nombre de chiffres de  $5000!$  puis  $10000!$ .
2. Utiliser la fonction `time` pour «chronométrer» le temps mis par Maple pour effectuer les calculs de la première question.
 

```
> st :=time() ;
l'instruction que vous voulez «chronométrer» :
time()-st ;
```
3. Et pour  $100000!$  ? Comment s'en sortir ? Avec un morphisme ? Qu'est-ce que c'est ? (utiliser aussi `evalf`).

**Exercice 5 (Somme double Vs Deux sommes simples)** Calculer en utilisant la fonction `sum` de Maple la somme

$$\sum_{1 \leq i, j \leq n} i \times j.$$

**Exercice 6 (Pêchons les décimales)** Déterminer une suite d'instructions permettant d'afficher le 100-ième chiffre (en partant de la droite) de  $2^{1000}$  (utiliser la partie entière `floor` et le reste d'une division euclidienne (`irem` ou `mod`)).

**Exercice 7** Nous sommes le mardi 7 septembre 2010. Quel jour de la semaine serons-nous le 7 septembre 2011 (on pourra effectuer une division euclidienne) ?

**Exercice 8 (Il est entier, la preuve)** Démontrons à l'aide de Maple que  $a = \sqrt[3]{20 + 14\sqrt{2}} + \sqrt[3]{20 - 14\sqrt{2}}$  est bien entier.

1. Simplifier à la main  $b = 3(20 + 14\sqrt{2})^{1/3}(20 - 14\sqrt{2})^{1/3}$  puis avec Maple à l'aide de `combine`.
2. Montrer que  $a$  est solution de l'équation  $x^3 - 6x - 40 = 0$  à l'aide des instructions `expand` et `simplify` ou `combine`.
3. Résoudre l'équation  $x^3 - 6x - 40 = 0$  avec `solve` et conclure.
4. Regarder de l'aide sur la fonction `radnormal` et utiliser là.