

## Chapitre II

## CALCULS ALGÈBRIQUES

## 1 Les variables

## 1.1 Noms des variables

Nous avons déjà utilisé des variables dans les chapitres précédents. Le nom d'une variable se note entre deux accents graves (*backquotes*) :

```
`Nom`
```

```
`Autre nom de variable !`
```

```
`Résultat`
```

Un nom peut contenir des espaces, des accents, des signes de ponctuations etc.

**Cas particulier.** Pour éviter l'utilisation systématique (et contraignante) des accents graves, certains noms peuvent être écrits sans délimiteurs, à condition de satisfaire les conditions suivantes :

- Le nom comporte au moins un caractère ;
- Le premier caractère est une lettre non accentuée (a-z ou A-Z) ;
- Les caractères suivants sont des lettres de préférence non accentuées, chiffres ou soulignement (*underscore*).

Ainsi, `x`, `N`, `Mon_resultat` désignent des noms.

**Attention !** Bien distinguer majuscules et minuscules.

Effectuer et corriger le cas échéant :

```
>n:=3;`p`:=5;
```

```
>résultat1:=n*p;
```

```
>autre résultat:=n/p;
```

```
>dernier_résultat:=n^p
```

## 1.2 Variables affectées ou non, affectation

Une variable peut être utilisée de deux façons :

- soit c'est une variable affectée (*assigned*), c'est-à-dire dont la valeur est complètement déterminée (un nombre, une expression, une chaîne de caractère) ;
- soit c'est une variable non affectée (*unassigned*), et sa valeur est alors son propre nom.

Lorsque l'on fait du calcul formel sur des expressions, on utilise des variables non affectées qui jouent le rôle des indéterminées ou paramètres de nos expressions mathématiques.

L'affectation, on l'a déjà vu, suit la syntaxe suivante :

```
>`nom de variable`:=expression;
```

### 1.3 Règle d'évaluation totale

Dans le cas général, Maple évalue les variables en suivant la règle d'évaluation totale (*full evaluation rule*). Chaque fois que Maple doit manipuler une expression contenant des variables, il remplace celles-ci par leurs valeurs. Si la valeur d'une variable est une variable, il recommence pour cette seconde variable. Et ainsi de suite jusqu'à ce que l'expression ne comporte plus que des constantes et des variables non affectées.

*Effectuer pour comprendre la règle d'évaluation totale :*

Entrée	Sortie
<pre>&gt;restart; &gt;b:=3*a; &gt;b; &gt;a:=1; &gt;b; &gt;a:=2; &gt;b; &gt;c:=3*a; &gt;a:=3; &gt;b;c;</pre>	

*Comment expliquer la différence entre **b** et **c** ?*

On verra plus tard que certaines variables ne suivent pas cette règle.

On peut empêcher l'application de cette règle en plaçant autour du nom de la variable des apostrophes (*quotes*). (*Voir les exemples ci-après*)

### 1.4 Réinitialisation d'une variable

L'affectation d'une variable reste effective pendant toute la durée de la session. Pour qu'une variable préalablement affectée soit à nouveau non affectée (pour pouvoir l'utiliser comme indéterminée mathématique par exemple), on peut :

- Réinitialiser la session en tapant la commande :

```
>restart;
```

Bien sûr, ce n'est pas cette méthode qu'il faut choisir si d'autres variables doivent conserver leurs valeurs pour des calculs ultérieurs.

Je vous conseille de débiter systématiquement chaque exercice par cette commande pour être sûr de ne pas manipuler des variables précédemment affectées sans le savoir.

- On a vu qu'une variable non affectée avait pour valeur son nom. Il suffit donc d'effectuer cette affectation, en prenant soin d'empêcher l'évaluation de la variable<sup>1</sup>. Sinon, l'opération est sans effet !
- On peut enfin utiliser la fonction `unassign()`, en prenant soin de l'appliquer à des variables pour lesquelles on empêche l'évaluation<sup>1</sup>.

*Effectuer :*

```
>restart;
>x:=3.14159;
>x;
>'x';
>x:=x; x; # Ceci n'a pas d'intérêt !
>x:='x'; x; # C'est mieux ainsi.
>x:=3.14159;
```

---

<sup>1</sup>Penser aux apostrophes

```
>unassign(x);
>unassign('x');
>x;
```

## 2 Les expressions

### 2.1 Première approche : opérateurs et opérandes

On appelle **expression** l'association de nombres, symboles, noms et opérateurs. Par exemple,

```
>sin(((2*n+1)/2)*theta)/sin(theta/2);
>a*x+b;
```

sont des expressions. Dans la seconde, on distingue **a**, **x** et **b**, appelés des **opérandes** de **\*** et **+** que l'on appelle des **opérateurs**. Les principaux opérateurs sont :

+	Somme	<=	Inférieur ou égal
-	Différence		Strictement supérieur
*	Produit	=	Égal
/	Quotient		Différent
** ou ^	Puissance	or	Ou
	Composition	and	Et
	Concaténation	-	Opérateur flèche
..	Intervalle ( <i>range</i> )	union	Union d'ensembles
:=	Affectation	intersect	Intersection d'ensembles
& *	Multiplication non commutative		et toutes les fonctions

L'utilisation de certains est très intuitive. Nous reparlerons le moment venu de la composition, de l'opérateur flèche, de la multiplication non commutative, de l'union et de l'intersection.

#### Fonctions **sum** et **product**.

La syntaxe est la suivante :

```
>sum(expression_en_i,i=a..b);
```

**a..b** désigne les entiers consécutifs entre **a** et **b**, et **i** doit être une variable non affectée. On peut utiliser les apostrophes pour être sûr que la variable utilisée n'est pas affectée.

On peut remplacer **sum** par **Sum**, la forme inerte correspondante. Elle se différencie de la première par le fait qu'aucune évaluation de la somme n'est entreprise par Maple. On peut demander l'évaluation d'une expression à l'aide de la fonction **value**.

Le calcul d'un produit s'effectue de façon analogue, à l'aide de la fonction **product** ou de sa forme inerte **Product**.

Effectuer :

```
>restart;
>j:=1;
>s:=sum(j,j=1..n);
>s:=sum('j','j'=1..n);
>Sum(i^2,i=1..n);
>value(%);
>sum(x^k/k!,k=0..infinity);
>Product(i,i=1..n):%=value(%);
```

Remarquer la signification du signe = dans la dernière entrée.

### Opérations sur les chaînes de caractères et les noms.

Une chaîne de caractère (*string*) est délimitée par des guillemets (*double quotes*).

```
"Chaîne de caractère"
```

```
"ici aussi ; et ci-dessous, c'est la chaîne vide"
```

```
""
```

Une chaîne de caractères peut contenir des espaces, des accents, des signes de ponctuations etc. L'opérateur de concaténation agit sur les chaînes de caractères ou sur les noms de variables. Il faut noter que le premier opérande dans une opération de concaténation n'est pas évalué. Les exemples ci-dessous expliquent comment s'en sortir.

Effectuer :

```
>restart;
```

```
>x:="J'aime ";
```

```
>y:="Maple.";
```

```
>z:="les maths.";
```

```
>x||y;
```

```
>" "||x||z;
```

## 2.2 Représentation interne : les arbres

Découvrons les fonctions qui permettent de manipuler les opérandes et les opérateurs d'une expression :

Effectuer :

```
>restart;
```

```
>expression:=a*x+b;
```

```
>op(expression);
```

```
>nops(expression);
```

```
>k:=2:op(k,expression);
```

```
>op(0,expression);
```

La première fonction donne les opérandes de l'expression, la seconde le nombre d'opérandes, la troisième le kème opérande et la dernière l'opérateur principal de l'expression. On peut appliquer ces fonctions à chacun des opérandes pour les décortiquer de la même façon. On regroupe cela dans un **arbre** représenté en FIG. 1.

Figure 1: Représentation en arbre de  $ax + b$

Construire les arbres pour les expressions suivantes :

$$ax^2 + bx + c \quad ; \quad \ln 3x + 4 \quad ; \quad \frac{5x}{y}$$

### 3 Transformation des expressions

Les expressions obtenues par un logiciel de calcul formel doivent souvent être simplifiées pour être utilisées.

#### 3.1 simplify

C'est la fonction à utiliser en priorité. Observons les exemples suivants :

Effectuer :

```
>restart;
>4^(1/2)+3;
>simplify(%);
>(sin(x))^4-(cos(x))^4;
>simplify(%);
>y:=ln(exp(x));
>simplify(y);
>assume(x,real);
>y;
>b:=sqrt(a^2);
>b;
>assume(a>=0);
>simplify(b);
```

#### 3.2 Autres transformations

##### Développer avec expand(expression)

Cette fonction permet de développer l'expression en utilisant la distributivité de  $*$  sur  $+$ , les formules trigonométriques  $\sin(a+b) = \dots$  ou la formule  $e^{a+b} = e^a e^b$ .

Effectuer :

```
>restart;
>expand((x+3)*(x-4));
>expand((x^2+x+1)^8);
>sort(%);
>expand((3*x^2+4)*(x+1),x+1);
>expand(sin(x+2*y));
```

##### Utiliser combine(expression, option)

Effectue l'opération inverse d'expand. On peut orienter la factorisation en indiquant l'option trig, exp, ln ou power.

Effectuer :

```
>combine(cos(a)*cos(b)-sin(a)*sin(b));
>combine((cos(a))^3);
>combine((x^a)^2);
>combine(exp(sin(a)*cos(b))*exp(cos(a)*sin(b)));
```

##### Réduire au même dénominateur et simplifier avec normal(exp\_rationnelle)

Effectuer :

```
>(x^3+y^3)/(x^2-y^2);
>normal(%);
```

```
>1/(x+1)+x/(x^2-1);
>normal(%);
```

**Factoriser avec des coefficients rationnels**  
à l'aide de `factor(expression)`

Effectuer :

```
>factor(6*x^2+18*x-24);
>factor((x^3+7*x^2+14*x+8)/(x^2+5*x+6));
>factor(8*x^3-12*x);
>factor(8*x^3-12*x,sqrt(6));
```

**Rendre le dénominateur rationnel (donc réel)**  
à l'aide de `rationalize(expression)`

On peut en particulier utiliser cette fonction pour multiplier numérateur et dénominateur par la quantité conjugué.

Effectuer :

```
>phi:=(1+sqrt(5))/2;
>1/phi;
>rationalize(%)
>z:=(3+4*I)/(2+x*I);
>rationalize(%);
```

**Modifier la forme d'une expression avec `convert`**

Cette fonction a de multiples usages. Citons la décomposition d'une fraction rationnelle en éléments simples, l'écriture exponentielle des expressions trigonométriques et réciproquement...

Effectuer :

```
>convert(1/((x^2-5*x+6)*(x^2+x+1)),parfrac,x);
>convert(sin(x),exp);
>convert(%,trig);
>rationalize(%);
>combine(%);
```

**Substituer des opérandes avec `subs` ou `subsop`**

Pour remplacer une sous-expression par une autre, à chaque fois qu'elle apparaît dans l'expression, on utilise la fonction `subs` :

Effectuer :

```
>subs(u=exp(x),ln(u));
>subs(a=sqrt(x),a+exp(a)+ln(a));
```

Si on ne veut remplacer qu'un seul opérande, on repère sa position dans l'expression et on utilise la fonction `subsop` :

Effectuer :

```
>subsop(2=ln(x),x+y+z);
>subsop(2=10,2/3);
```

## 4 Exercices

**II.1** Simplifier les expressions suivantes :

$$(a) \quad \sqrt{(x-5)^2} \quad (x > 5) \qquad (b) \quad \sqrt[3]{7+5\sqrt{2}} - \sqrt[3]{-7+5\sqrt{2}} \qquad (c) \quad 8^{1/3} + 5$$

**II.2** Retrouver les relations trigonométriques suivantes :

$$\begin{array}{ll} (a) \quad \cos^2 a = \frac{1}{1 + \tan^2 a} & (b) \quad \cos^2 a - \sin^2 a = \cos 2a \\ (c) \quad \sin a \cos b = \frac{1}{2}(\sin(\dots) + \sin(\dots)) & (d) \quad \frac{1 - \tan^2 \frac{a}{2}}{1 + \tan^2 \frac{a}{2}} = \cos a \\ (e) \quad \cos^2 a = \frac{1 + \cos 2a}{2} & (f) \quad \sin p + \sin q = 2 \dots \end{array}$$