

1 Calcul Élémentaire

Maple connaît les opérations élémentaires.

Les signes d'opération sont : + , - , * , / pour les quatre opérations et : ** ou l'exposant pour la puissance.

1.1 Calcul exact

```
> 99*99;
9801
```

Il calcule vite et bien, mais, à priori, il ne fait que du calcul exact.

```
> 99**99;
36972963764972677265718790562880544059566876428174110243025997242\
35525704552775234214106500101282327279409788895483265401194\
29996769494359451621570193644014418071060667659301384999779\
999159200499899
```

On a une variable spéciale qui contient le dernier résultat.

```
> _;
36972963764972677265718790562880544059566876428174110243025997242\
35525704552775234214106500101282327279409788895483265401194\
29996769494359451621570193644014418071060667659301384999779\
999159200499899
```

1.2 Calcul approché ou « flottant »

On peut forcer Maple à se mettre en calcul approché avec 10 ou un nombre variable de chiffres significatifs.

```
> evalf( );
.3697296376 10198
> Digits:=23;evalf(1/13);Digits:=10;
Digits := 23
.076923076923076923076923
Digits := 10
```

1.3 Constantes

Le nombre π , qu'il appelle « Pi », est connu exactement, on fera attention à la majuscule...

Maple connaît aussi i qu'il appelle « I » et ∞ qu'il appelle « infinity ».

```
> Pi;
π
> evalf(Pi);
3.141592654
```

1.4 Fonctions usuelles

Maple connaît de nombreuses fonctions usuelles :

```
> sqrt(5);
```

$$\sqrt{5}$$

Mais on notera avec soin que Maple fait toujours du calcul **exact**.

```
> (sqrt(5)+3)**2;
```

$$(\sqrt{5} + 3)^2$$

1.5 Premier calcul formel

Maple ne fait pas à priori de transformations importantes, comme, par exemple, développer un polynôme. Il faut lui demander explicitement.

La fonction « expand » développe tous types d'expressions.

```
> expand(");
```

$$14 + 6\sqrt{5}$$

Par contre, de lui-même, Maple simplifie de lui-même les fractions :

```
> 10/6;
```

$$\frac{5}{3}$$

1.6 Encore du calcul exact

Maple fait du calcul exact, on l'a déjà dit :

```
> sin(1);
```

$$\sin(1)$$

```
> sin(Pi/6);
```

$$\frac{1}{2}$$

Maple connaît beaucoup de résultats plus ou moins courants.

```
> sin(Pi/12);
```

$$\frac{1}{4}\sqrt{6}\left(1 - \frac{1}{3}\sqrt{3}\right)$$

Il répond de façon simple quand il ne sait pas :

```
> sin(Pi/7);
```

$$\sin\left(\frac{1}{7}\pi\right)$$

Maple connaît entre autres les fonctions \cos , \tan , \ln , abs (valeur absolue ou module) en plus de celles déjà utilisées...

2 Premières Variables et Calcul Formel

2.1 Variables

Maple connaît et utilise des variables.

- Une variable a un **nom**, constitué d'une lettre et d'un nombre variable de lettres ou de chiffres.
- Une variable peut être **libre**, sans valeur, ou **affectée**, avec une valeur, qui peut être une expression contenant des variables libres ou non.

A priori, une variable est libre...

L'affectation est « := » :

```
> x:=7;
x := 7
```

2.2 Evaluations

On vient d'affecter x , on va faire quelques calculs avec cette variable affectée, et regarder les mêmes calculs avec une variable y libre :

```
> x+2;
9
> (x+2)**7;
4782969
> (y+2)**7;
(y+2)^7
> expand(");
y^7 + 14y^6 + 84y^5 + 280y^4 + 560y^3 + 672y^2 + 448y + 128
```

On observera ce qu'on appelle **l'évaluation complète** :

```
> y:=x+2;
y := 9
```

On peut facilement **libérer** une variable affectée :

```
> x:='x';
x := x
```

Ce qui peut réserver quelques surprises :

```
> y;
9
```

On va réaffecter y pour bien voir l'évaluation complète :

```
> y:=x+2;
y := x + 2
> z:=2/y;
z := 2 / (x + 2)
```

2.3 Un peu de Trigonométrie

```
> p:=cos(5*x);
                                     p := cos(5 x)
> expand(p);
16 cos(x)5 - 20 cos(x)3 + 5 cos(x)
```

2.4 Dérivation et Intégration

On définit ici une expression :

```
> y:=ln(x)/(x+1);
                                     y :=  $\frac{\ln(x)}{x+1}$ 
```

Expression que l'on va dériver par rapprt à x .

```
> diff(y,x);
                                      $\frac{1}{x(x+1)} - \frac{\ln(x)}{(x+1)^2}$ 
```

On peut continuer à dériver :

```
> diff(",x);
                                      $-\frac{1}{x^2(x+1)} - \frac{2}{x(x+1)^2} + 2\frac{\ln(x)}{(x+1)^3}$ 
```

La fonction « simplify » n'est pas magique :

```
> simplify("");
                                      $\frac{-3x^2 - 4x - 1 + 2\ln(x)x^2}{x^2(x+1)^3}$ 
```

On peut aussi calculer des primitives ou des intégrales entre des bornes.

Ici z est l'expression de la variable libre x : $(x^2 + 1)e^{-x}$.

```
> int(z,x);
                                      $-e^{(-x)}x^2 - 2xe^{(-x)} - 3e^{(-x)}$ 
> int(z,x=0..2);
                                      $-11e^{(-2)} + 3$ 
> evalf("");
                                     1.511311885
```

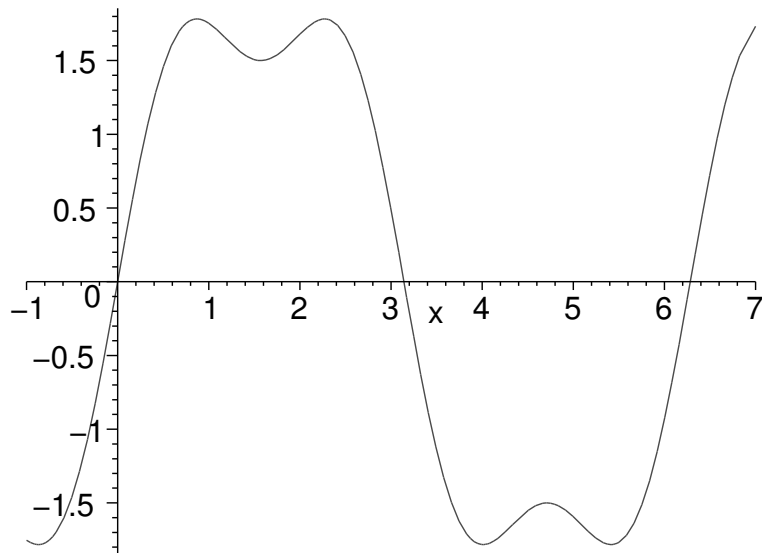
3 Tracé de courbes

Maple possède un bon module de tracé de courbes :

```
> y:=sin(3*x)/2+2*sin(x);
```

$$y := \frac{1}{2} \sin(3x) + 2 \sin(x)$$

```
> plot(y,x=-1..7);
```



4 A retenir

- 1) `>` et `;` qui sont le signal d'invite et le signal de fin de commande.
- 2) `+`, `-`, `*`, `/`, `**` les opérations usuelles, le parenthésage utilise les parenthèses habituelles : `()`.
- 3) Le calcul **exact** recherché à priori par Maple
et le calcul **flottant** qu'on peut forcer en utilisant la fonction : `evalf(expression)`.
- 4) Les **fonctions** qui sont très nombreuses.
 - a) Les fonctions mathématiques usuelles,
par exemple : `sqrt(expression)` `ln(expression)` `sin(expression)` `cos(expression)` ...
Il s'agit ici de la racine carrée, du logarithme, du sinus et du cosinus.
 - b) Les fonctions mathématiques qui ne sont pas perçues comme telles en mathématiques,
par exemple : `diff(expression)` `int(expression)` `plot(expression)` ...
pour le calcul de dérivées, d'intégrales ou le tracé de courbes.
 - c) Les fonctions de manipulation d'expressions ,
par exemple : `expand(expression)` `simplify(expression)` ...
pour le « développement » ou la « simplification » de tous types d'expressions.
- 5) La manipulation de **variables**.
 - a) Une variable peut être **libre** ou **affectée**, c'est à dire dans ce dernier cas qu'elle a reçu une valeur.
Cette valeur peut être n'importe quelle *expression*.
Un nom de variable commence par une lettre et contient ensuite des lettres ou des chiffres. Les noms déjà utilisés par Maple sont interdits.
 - b) L'affectation `:=` qui permet de donner une valeur, ou une nouvelle valeur, à une variable.
 - c) La règle de l'évaluation complète qui fait que Maple remplace dans une *expression* toutes les variables affectées par leur valeur.
 - d) La libération des variables affectées qui peut se faire :
 - i) globalement par : `restart;`
 - ii) individuellement par : `var := 'var' ;`
 - e) Les variables ou constantes du système,
par exemple : `Digits` `Pi` `infinity` ...