

# Commandes et expressions de base en Maple

## 1 Commandes générales

On termine toujours une instruction Maple par ; ou :

Trouver de l'aide: `?nom_fonction`

Rappel de la dernière instruction exécutée: `%`

Changer globalement la précision de Maple: `Digits:=precision;`

Evaluer une expression sous forme décimale: `evalf(expr,precision)`

Longueur d'un chiffre ou autre (bien lire l'aide avant): `length(expr)`

## 2 Commandes Maths

### Fonctions

On définit une fonction à une variable avec la syntaxe suivante:

`f:=x->expr(x)`

Ex: `f:=x->ln(2*x+1)`

Pour composer des fonctions on utilise le symbole @.

Pour une fonction à deux variables, on met les variables entre parenthèses:

`f:=(x,y)->expr(x,y)`

Ex: `f:=(x,y)->sin(x^2+y^2)`

Sinon, Maple connaît déjà pas mal de fonctions...

### Constantes

`I`, `Pi`, `exp(1)` (et non `e`), `infinity`, `-infinity`...

### Graphiques

Dessiner une fonction de une variable: `plot(f(x),x=a..b,options)` ou `plot(f,a..b,options)`  
(dans les commandes qui suivent, on peut utiliser l'une ou l'autre de ces syntaxes)

Dessiner plusieurs fonctions sur le même graphique: `plot([f,g],a..b)`

Dessiner une courbe paramétrée  $t \rightarrow (u(t), v(t))$ ,  $t \in [a, b]$ : `plot([u,v],a..b)`

Dessiner une fonction à deux variables: `plot3d(f(x,y),x=a..b,y=c..d)`

Remarque: n'oubliez pas, la plupart des commandes Maple ont la syntaxe suivante: `commande(arguments,options)`

## Commandes de base

### Manipuler des expressions

On peut développer, factoriser, simplifier avec les commandes `expand()`, `factor()`, `simplify()` et leurs options (voir aide).

### Arrondir

`trunc()` (arrondi à l'entier le plus proche de 0), `round()` (arrondi à l'entier le plus proche), `floor()` (arrondi à l'entier inférieur), `ceil()` (arrondi à l'entier supérieur), `frac()` (donne la partie fractionnaire du nombre, *i.e.* renvoie  $x - \text{trunc}(x)$ ).

### Complexes

On a les commandes de base `Re()`, `Im()`, `abs()`, `argument()`, `conjugate()`.

Passer un nombre sous la forme  $a + ib$ : `evalc()`

Utiliser un nombre sous forme polaire: `polar(module, argument)`

Convertir un nombre sous forme polaire: `polar(nombre)` ou `convert(nombre, polar)`

### Equations et systèmes

Résoudre une équation: `solve(equation(s), variable(s))`

S'il y a plusieurs équations (système), on les rentre entre accolades, séparées par une virgule. De même pour les variables, s'il y en a plusieurs.

Ex: `solve({3*x+y=0, x^2*y=1}, {x, y})`

Si Maple vous rend une expression avec des `RootOf`, vous pouvez utiliser la commande `allvalues(expr)` pour finir le calcul des solutions.

Pour résoudre une équation ou un système de façon approchée: `fsolve(equation(s), options)`

Pour avoir toutes les solutions d'une équation polynomiale, on utilise l'option `complex` de la commande `fsolve`.

### Limites, dérivées, intégrales...

Calculer une limite: `limit(f(x), x=a, option)`

Ici on peut utiliser l'option `left` ou `right` pour calculer une limite à gauche ou à droite.

Dériver une fonction: `diff(f(x), x)` ou `D(f)`

Dérivée multiple: `diff(f(x), x$n)`

Dériver par rapport à plusieurs variables: `diff(f(x, y), x$n1, y$n2)`

Donner la primitive: `int(f(x), x)`

Intégrer: `int(f(x), x=a..b)`

Les fonctions `limit`, `diff` et `int` possèdent des fonctions inertes équivalentes:

`Limit`, `Diff` et `Int` qui rendent l'expression sous une jolie forme sans l'évaluer.