

Pense-bête Maple

RAPPEL : SAUVEGARDEZ REGULIEREMENT VOTRE TRAVAIL
Sauvegarder tous les quart d'heure, c'est s'autoriser à perdre un quart d'heure de travail.

1 Commandes de base

- input Maple : [**>**] On en insère un en cliquant sur le bouton correspondant.
- fin d'instruction : **;** (avec affichage), **:** (sans affichage)
- exécuter une commande : touche Entrée. Pour revenir à la ligne sans exécuter : Maj+Entrée.
- aide : **?help** ou bien **?** suivi de la commande.
Exemple : **?iquo**
- commentaire : **#**
Exemple : **a:=5; #a:=6; ceci est un commentaire, a vaut toujours 5**
- rappel du dernier résultat en mémoire : **%** Avant-dernier résultat : **%%**
- affectation : *variable* := *valeur*
- test d'égalité : *variable* = *valeur*
- test de différence : *variable* <> *valeur*
- libération d'une variable : **unassign('variable')** ou bien *variable* := '*variable*'
- effacement de la mémoire : **restart**
- chargement de package : **with(package)**
- interruption du calcul : cliquer sur le bouton STOP.

2 Calcul numérique et symbolique

- opérations élémentaires : +, -, *, /, ^
- opérations répétées : **add**, **mul**
- test d'inégalité : <, >, <=, >=
- constantes : π **Pi**, i **I**, e **exp(1)**
- fonctions : **sqrt**, **exp**, **ln**, **sin**, **cos**, **tan**, **arcsin**, **arccos**, **arctan**, **sinh**, **cosh**, **tanh**, **arcsinh**, **arccosh**, **arctanh**, etc.
- valeur approchée : **evalf**
- opérations sur les nombres complexes : **Re**, **Im**, **abs**, **argument**
- opérations sur les nombres entiers : **iquo** (quotient d'une division euclidienne), **irem** ou **mod** (reste de division euclidienne), **igcd** (PGCD), **ilcm** (PPCM)
- opérations sur les expressions formelles : **expand**, **factor**, **simplify**, **eval(expression, variable=valeur)**
Exemple : **eval(y*x^2+y, y=x);** renvoie $x^3 + x$.
Guide de simplification en Maple : <http://algo.inria.fr/salvy/M1ENS/simplification.pdf>
- opérations sur les polynômes et fractions rationnelles : **degree**, **coeff(polynôme, x, n)** (coefficient devant x^n), **quo** (quotient d'une division euclidienne), **rem** (reste de division euclidienne), **gcd** (PGCD), **lcm** (PPCM), **collect(polynôme, variable)** (regroupe par puissances de la variable), **normal** (réduction des fractions rationnelles), **convert(fraction, parfrac)** (décomposition en éléments simples)
- dérivée/primitive/intégrale d'expression : **diff(expression, variable)**, **int(expression, variable)**, **int(expression, variable=borne₁..borne₂)**
- équations : **solve**, **dsolve** (équations différentielles)

3 Graphiques

- tracé de fonctions d'une variable réelle : **plot**
- tracé de fonctions à deux variables réelles : **plot3d**
- tracé de courbe implicite : **implicitplot** (dans le package **plots**)
- tracé de surface implicite : **implicitplot3d** (dans le package **plots**)

4 Programmation

- valeurs et opérations booléennes : `true`, `false`, `and`, `or`, `not`, `evalb` (valeur d'une expression booléenne)
- conditionnelle :

```
if condition1 then
  instruction1
(elif condition2 then
  instruction2)
(else
  instruction3)
end if
```

Exemple :

```
if x mod 3 = 0 then
  x:
elif x mod 3 = 1 then
  x-1:
else
  x-2:
end if;
```
- boucle for : for *variable* from *valeur*₁ to *valeur*₂ (by *pas*) do *instruction* end do
Exemple :

```
for i from 10 to 1 by -1 do
  print(i):
end do;
```
- boucle while : while *condition* do *instruction* end do
- fonctions : (*argument*₁, *argument*₂, ...) -> *sortie*
Exemple : `f := (x,y) -> (x2+y2)/(x*y)`;
- procédures : `proc(argument1, argument2, ...) instruction return sortie end`
Exemple :

```
maximum:=proc(a,b)
if a>b then
  return a:
else
  return b:
end if:
end;
```
- déclaration de variables locales dans une procédure : `local variable1, variable2, ... :`
- option `remember` : demande à Maple d'enregistrer les résultats des appels de la procédure, pour ne pas les recalculer.
- séquences : *élément*₁, *élément*₂, ... (séquence), `NULL` (séquence vide), `S1,S2` (concaténation),
`seq(expression, variable=borne1..borne2)`
Exemple : `seq(n2,n=1..10)` renvoie la séquence des 10 premiers carrés.
- listes : [*élément*₁, *élément*₂, ...] (liste), [] (liste vide), [S] (transforme une séquence en liste), `op(L)` (transforme une liste en séquence), `L[i]` (accès au i-ème élément), `L[i..j]` (liste des éléments de i à j), `L[-1]` (dernier élément de la liste)
Exemple : concaténation de deux listes : `[op(L1),op(L2)]`