Les expressions et leur évaluation

Maple est un logiciel de calcul formel, permettant de manipuler des expressions mathématiques contenant des symboles. Au début, nous l'utiliserons comme une "super-calculatrice".

Principes généraux

- Chaque ligne de commande est saisie après le signe `>` et est validée par la touche <*entrée*>,
- les expressions mathématiques sont constituées de constantes, symboles, opérateurs (par exemple `+, -, *, /, ^`), parenthèses et appels de fonctions (prédéfinies ou programmées par soi-même),
- toute expression est évaluée automatiquement par Maple, sauf si elle est entourée d'apostrophes. Si l'on veut que le résultat s'affiche, il faut terminer la commande par un point-virgule (le contraire de Matlab !), sinon on met le signe "deux points",
- pour consulter l'aide en ligne sur un objet (opérateur, fonction etc.), il faut entrer `?` suivi de son nom.

Exemple 1

L'opérateur d'*assignation* `:=` permet de nommer une expression :

nom `:=` *expr*

L'opérateur de *substitution* `subs` permet de remplacer une sous-expression par une autre :

`subs(`x `=` *valeur*`,` *expr*`)`

donne une nouvelle expression obtenue en remplaçant x par *valeur* dans *expr*.

1. Entrer l'expression $\frac{1+x^2}{x-1}$ et la nommer y : `> y:=(1+x^2)/(x-1);`
2. Calculer sa valeur en $x = -2$: `> subs(x=-2,y);`
3. Calculer sa limite à gauche en $x = 1$: `> limit(y,x=1,left);`

Exemple 2

En s'inspirant de l'exemple précédent, calculer les limites suivantes :

$$\lim_{x\to+\infty} \sqrt{x+\sqrt{x}}-\sqrt{x}, \quad \lim_{x\to0} \frac{\sqrt{1+3x}-1-x}{\sin x}, \quad \lim_{x\to0} \left(\frac{2}{x}-\frac{\ln(\mathrm{e}^2+x)}{|x|}\right)$$

Fonctions à utiliser : `sqrt, sin, ln, exp, abs.` L'infini se note `infinity`.

Exemple 3 *(expressions symboliques)*

L'opérateur `restart` réinitialise le système et il est vivement conseillé de l'utiliser avant de commencer toute nouvelle série de calculs.

Taper les commandes suivantes, et commenter les résultats affichés :

```
> restart : x:=2*y : y:=1 : y:=2 : x;
> restart : y:=1 : x:=2*y : y:=2 : x;
> restart : y:=1 : x:=2*'y' : y:=2 : x;
```

Exemple 4 *(approximations numériques)*

Maple peut calculer numériquement en précision arbitraire (par défaut, il donne 10 chiffres significatifs).

1. Taper `evalf(Pi,100);` que signifie le 100 de cette commande ?

2. Soit y l'expression $(\exp(\pi\sqrt{163}) - 744)^{1/3}$. Calculer une valeur approchée de y avec 20 chiffres significatifs. Calculer ensuite une valeur approchée de y avec 100 chiffres significatifs, puis recalculer une valeur approchée de y avec 20 chiffres significatifs. Au vu des résultats, que peut-on conclure ?

Exemple 5 *(séquence, sommation)*

Un *séquence* est une suite d'expressions séparées par des virgules. L'opérateur `seq` permet, à partir d'une seule expression, de construire une séquence en faisant varier un symbole entre deux valeurs :

`seq(`*expr*`, n=a..b)`

donne la séquence des *expr* pour n allant de a à b (avec un pas de 1).

Pour n entier naturel, on pose :

$$u_n = \frac{\sqrt{8}}{9801}\,\frac{(4n)!\,(1103 + 26390\,n)}{(n!)^4\,396^{4n}}$$

1. Entrer cette fraction et la nommer u (factorielle n se note `n!`).

2. Faire afficher la valeur de u_0, puis la séquence des valeurs de u_n pour n allant de 0 à 10, enfin la valeur de $S = \sum_{n=0}^{10} u_n$ (utiliser l'opérateur de sommation `sum`).

3. Comparer numériquement $1/S$ avec π (la formule est due au mathématicien indien *Ramanujan*).

Exemple 6 *(nombres complexes)*

Par défaut, Maple calcule dans le corps des complexes. Le nombre complexe i (racine carrée de -1) se note `I`. L'expression `evalc(z)` met le nombre complexe z sous forme algébrique $x + iy$ (x et y réels).

1. Soit z un nombre complexe d'écriture algébrique $x + iy$. On pose $Z = \frac{z^2}{z+i}$. Ecrire Z sous forme algébrique (avec `evalc`). En déduire les expressions des parties réelles et imaginaires de Z, en fonction de x et y.

2. Soient a et b deux nombres complexes. En posant $j = \mathrm{e}^{2i\pi/3}$, démontrer que l'on a :

$$a^3 + b^3 = (a + b)(a + bj)(a + bj^2).$$

(développer le membre de droite avec `expand`).

***

Exercice 1

1. Construire la liste L des entiers de la forme $k^2 + k + 41$ pour k allant de 0 à 100 (utiliser l'opérateur `seq`).

2. Parmi les éléments de L, afficher ceux qui sont des nombres premiers (utiliser `select` et `isprime`). Combien y en a-t-il ?

Exercice 2

La *suite de Perrin* (u_n) est définie par : $u_0 = 3, u_1 = 0, u_2 = 2$ et $u_n = u_{n-3} + u_{n-2}$ pour $n \geq 3$.

1. Calculer u_{1000} (utiliser une boucle `for`). Combien de chiffres comporte u_{1000} ?

2. Montrer qu'en supposant $2 \leq n \leq 1000$, alors n divise u_n si et seulement si n est premier.