INITIATION AU SYSTEME DE CALCUL FORMEL MAPLE

I - Présentation du système Maple

1. Avantages de Maple

Maple est un système de calcul formel (Computer Algebra = Algèbre par ordinateur), c'est à dire un système capable de traiter des objets mathématiques de manière symbolique.

On peut faire des calculs exacts avec des nombres rationnels (par exemple 1/3) ou des nombres irrationnels (par exemple $\sqrt{3}$).

On peut également effectuer des calculs complexes en virgule flottante (évaluation d'expressions, intégration numérique, statistiques optimisation). La précision de ces calculs peut être aussi grande que l'on désire (le nombre π peut être calculé avec 500 décimales).

Possibilité de générer automatiquement du code de programmation Fortran ou C.

Dans la version 4 et suivantes, on peut insérer du texte et des titres dans une séquence de calculs et préparer des cours ou des rapports. La sortie peut être imprimée directement ou traduite dans le langage de traitement de texte scientifique TEX.

Maple dispose également d'une aide en ligne efficace (accès par sujets ou par mots-clefs).

2. Aspect interactif de Maple

Maple est un interpréteur de commandes et non pas un compilateur, c'est à dire qu'il effectue les calculs au fur et à mesure que ceux-ci sont demandés. Toutes les commandes sont mémorisées dans une sorte de journal de bord appelé "**feuille de travail**" qu'il est possible de sauvegarder. En cas d'erreur, on peut revenir en arrière et réinterpréter la commande après correction.

La programmation d'applications "**Clefs en mains**" est possible en rassemblant des commandes dans un fichier de texte, soit dans une feuille de travail (mais dans tous les cas, les commandes sont réinterprétées à chaque exécution). On peut également stocker des parties de programmes préinterprétées dans une bibliothèque personnelle.

Toutes les structures d'un langage évolué existent en Maple (boucles, conditions, fonctions, procédures).

Le seul inconvénient est que pour exécuter une application Maple, il faut disposer de l'interpréteur. Il n'est pas possible de compiler un programme et de l'exécuter sur un autre ordinateur.

II - Eléments de syntaxe de Maple

1. Alphabet

a - z, A - Z, 0 - 9 Les minuscules et les majuscules sont des caractères **différents**
@ $ &⎵ ⎵≠ Le blanc est non significatif, sauf dans une chaîne de caractères
+ - * ^ / , ; . : %
( )[ ] { } < > = ' " ` \

Lettres grecques

Le langage peut utiliser les lettres grecques, celles-ci doivent être écrites en toutes lettres (minuscules ou majuscules suivant la première lettre)

alpha	α	Gamma	Γ
delta	δ	Delta	Δ

Exceptions

Pi génère le caractère π **et lui affecte la valeur** (π). C'est une constante.
GAMMA (tout le mot étant écrit en majuscule) représente la fonction Γ (fonction Eulérienne de deuxième espèce).

Association de caractères spéciaux

: =	affectation
->	déclaration de fonction
** ou ^	élévation à la puissance
& ou &*	multiplication matricielle

2. L'identificateur Maple

Tout objet peut être identifié par un nom symbolique. Ce nom symbolique :

- Débute par un caractère.
- Il est de longueur quelconque.
- Le blanc souligné est utilisable.
- On fait la distinction entre majuscules et minuscules.

3. L'instruction Maple

Elle se termine par : ou ;. **Dans tous les cas, l'instruction est exécutée**, si elle se termine par ; le résultat est affiché.

- Les blancs ne sont pas significatifs.
- Le caractère de continuation est \.
- Les caractères qui suivent le caractère # sont des commentaires.

Exemple

```
> 2*x+1;
```

$$2\,x + 1$$

```
> (cos(2*theta+Pi))^2;
```

$$\cos(2\,\theta + \pi)^2$$

```
> Gamma + Delta + Pi;
```

$$\Gamma + \Delta + \Pi$$

```
> GAMMA(5);
```

$$24$$

4. La ligne Maple

Elle peut comporter plusieurs instructions séparées par : ou par ; suivant qu'on désire afficher les calculs ou non.

Si une instruction dépasse la largeur de l'écran, on utilise le caractère \ pour rattacher la ligne suivante à la ligne courante. Cette césure est effectuée automatiquement lorsqu'on diminue la fenêtre Maple pour que toutes les instructions soient visibles à l'écran.

5. Librairies et paquetages

Ce sont des ressources spécialisées que l'on appelle lorsqu'on veut traiter des problèmes dans certains domaines particuliers.

Appel d'un paquetage :

with (paquetage)	lorsqu'on veut appeler toutes les ressources d'un paquetage
paquetage[fonction]	pour appeler uniquement une fonction dans un paquetage

Exemples

with(linalg)	appel de toutes les ressources de l'algèbre linéaire
plots[implicitplot(equ, parm)]	tracé d'une fonction implicite en utilisant les ressources du paquetage plots

Appel d'une librairie :

readlib(librairie)

Remarque : Cette instruction est obsolète depuis la version VII de MAPLE

III - Les types en Maple

1. Les types

Sauf exception, les **objets Maple n'ont pas de type déclaré et celui-ci peut varier au fur et à mesures d'un travail.**

Exemples

p:= 2	entier
p:= (p+1)* x^p-(p+2)*x	polynôme
q:= p^2-2	polynôme
p:= p/q	fraction rationnelle
p:= [p,q,p-q]	liste

Le seul type qui doit être déclaré est le type tableau parce qu'il faut **réserver de la place en mémoire et des pointeurs pour un tableau**.

A: = array (1..2, 1..3)

Cette instruction crée un tableau bidimensionnel avec les bornes 1 et 2 pour le premier indice et les bornes 1 et 3 pour le deuxième indice.

On peut utiliser des variables à condition de leur avoir donné une **valeur entière avant** la déclaration du tableau.

Les bornes ne doivent pas être **symboliques**.
Le premier indice du tableau est l'indice de ligne.

2. Détermination du type d'un objet Maple

Il existe deux manières de connaître le type d'un objet

whattype (objet)	renvoie le type (travail interactif)
type (objet, type testé)	vrai si l'objet est du type "type testé"

type est une fonction à résultat booléen permettant de tester un type. On l'utilise plutôt en programmation.

If type (p, rational) then ...(exécution des instructions adéquates)

Cette fonction n'est pas la réciproque de whattype. Tant que l'objet peut être considéré comme étant du type testé, la fonction type (objet, type) donnera le résultat vrai.

Exemples

```
> p:= 2;
```

p:=2

```
> whattype (p);
```

integer

```
> type (p, complex);
```

true

```
> type (p, rational);
```

true

```
> type (p, polynom);
```

true

3. Conversions de type en Maple

Pour les objets simples (nombres, polynômes, etc...), la conversion est automatique au moment de l'affectation.

On peut également utiliser la fonction convert, surtout pour convertir des objets composites (listes, ensembles, vecteurs, matrices).

4. Différents types en Maple

2	entier
$\frac{3}{2}$	rationnel (fraction)
1.5	réel flottant (1.5 est un objet différent de $\frac{3}{2}$)
3* x** 2-6	polynôme
3* x**2-6=0	équation
(3x + 2)/(x - 3)	ce type n'est pas reconnu comme fraction rationnelle
s:= sin (x^2 - 6) * exp(-x)	type * expression non reconnue comme fonction
f(x):= unapply (s, x)	type procédure ou type fonction
{2, 3, 5, x - 2, z - 6}	ensemble
[2, 3, 5, x - 2, z - 6]	liste
array (1..3, 1..3)	tableau à deux dimensions whattype donne le type string pour cet objet
matrix(2, 2, [1, 0, 2, 1])	lorsqu'un tableau est déclaré comme matrice, l'instruction whattype retourne bien le type matrix mais la déclaration d'une matrice n'est possible que si l'on a chargé auparavant le paquetage d'algèbre linéaire.