

POINTS IMPORTANTS POUR LE PARTIEL

POINT 1. Le plus important pour bien utiliser Maple est de savoir consulter l'aide, grâce à la commande :

?nom.de.la.commande

Il ne faut jamais utiliser une nouvelle commande sans aller vérifier sa syntaxe dans l'aide!!!

On peut faire une recherche par mots-clés dans l'aide si on ne connaît pas le nom exact de la commande.

POINT 2. Toutes les commandes de Maple doivent se terminer par ; ou par : (dans ce cas, Maple n'affichera pas le résultat).

Pour passer à la ligne sans que Maple exécute (par exemple pour écrire une procédure), on utilise la commande **shift+entrée**.

POINT 3. Il est bon en général de donner un nom à ce que l'on définit pour pouvoir réutiliser facilement ce que l'on a fait.

On utilise pour ça la commande **:=** (et non =).

Au début d'un nouveau travail (par exemple d'un nouvel exercice), il est *fortement conseillé* de réinitialiser toutes les variables (c'est-à-dire d'effacer de la mémoire tous les noms que l'on avait donnés), on utilise pour cela la commande **restart** ;

POINT 4. Revoir la différence entre une séquence, un ensemble et une liste (TP2).

Savoir utiliser la commande **seq** pour créer une séquence.

Quelques commandes utiles pour travailler sur les listes et les ensembles :

- **nops(L)** permet d'obtenir le nombre d'éléments de L, lorsque L est une liste ou un ensemble.
- **op(L)** permet d'extraire le contenu de L (par exemple pour créer une séquence), lorsque L est une liste ou un ensemble.
- **L[i]** permet d'obtenir le i-ème élément de L, uniquement lorsque L est une liste (les éléments d'un ensemble ne sont pas ordonnés de toute façon).
- **L[i..j]** permet d'obtenir la sous-liste commençant à l'élément i de L jusque l'élément j.

POINT 5. Revoir la différence entre une fonction et une expression (rappel au début du TP 3). Savoir notamment définir une fonction.

T.S.V.P

POINT 6. Connaître la syntaxe pour écrire une procédure, une boucle **for**, une boucle **while** et un test **if**.

Pour éviter d'oublier de terminer une procédure, une boucle ou un test, il est conseillé de toujours commencer par écrire (par exemple pour une boucle **for**) :

```
for i from ... to ...  
do ...  
end do
```

et de remplacer ensuite les ... par ce qu'on veut.

POINT 7. Certaines fonctions de Maple ne sont pas accessibles directement, il faut donc charger une nouvelle bibliothèque (un "package") avec la commande :

```
with(nom.de.bibliothèque) ;
```

Par exemple, on a vu les commandes :

- **pointplot**, que l'on utilise après avoir chargé la bibliothèque *plots*.

Ce qui donne :

```
with(plots) ;  
pointplot([1,2],[2,3]) ;
```

- **vector** et les autres commandes d'algèbre linéaire que l'on utilise après avoir chargé la bibliothèque *linalg*.

Il ne faut pas oublier de re-charger une bibliothèque si on en a de nouveau besoin après un *restart*.

POINT 8. Se rappeler que Maple affiche les résultats comme il a envie de les afficher et qu'il faut parfois lui forcer la main pour obtenir ce que l'on veut, on a vu par exemple comme commandes :

- **simplify** pour simplifier une expression algébrique.
- **evalf**, **evalc**, **evalb**, **evalm** pour obtenir une évaluation d'un flottant, d'un complexe, d'un booléen, d'un vecteur.

POINT 9. Essayer de se rappeler quelques unes des commandes qu'on utilise trois fois par cours, par exemple (en plus de celles que j'ai déjà mentionnées) :

```
plot, solve, sqrt, ithprime, Im, Re, diff,...
```

POINT 10. Me poser toutes les questions possibles AVANT le partiel! (si besoin par mail).