

TP 1 – Introduction à Maxima

Maxima est un logiciel de calcul formel, c'est-à-dire, un logiciel qui permet de faire des mathématiques en manipulant des expressions symboliques. Contrairement à la plupart des langages classiques de programmation il peut traiter non seulement des quantités numériques (entières, réelles, complexes) mais aussi des polynômes, des fonctions, des séries,... et effectuer des opérations courantes : dérivation, intégration, limites, simplifications,...

1 La feuille de travail

Une session de travail type avec **Maxima** commence évidemment par le lancement du logiciel avec son interface graphique nommé : **wxMaxima**. Une *feuille de travail* vide est alors affichée à l'écran, et **Maxima** est prêt à travailler en mode interactif : on lui donne des instructions au clavier, que l'on valide par la combinaison de touches <Majuscule>+<Entrée>, et il y répond de manière appropriée.

1.1 Premières commandes Maxima

Maxima indique qu'il est prêt à recevoir une commande en affichant le texte **Maxima is ready for input** en bas à gauche (la langue peut changer selon l'installation) et un curseur horizontal sur la zone du milieu. On peut alors entrer une commande en terminant par le symbole ; et un appui sur la combinaison de touches <Majuscule>+<Entrée> demandera à **Maxima** d'interpréter la commande saisie.

Exercice 1. Exécuter la commande `99*97`; pour obtenir

(%i1)	99*97;
(%o1)	9603

Remarquer les indications (%i1) apportées à gauche de la ligne `99*97` et (%o1) à gauche de la ligne `9603`, elles font respectivement référence à la première (1) entrée saisie (i) et la première (1) sortie (o) produite (le i est pour input et le o est pour output).

N'oubliez pas les signes * entre les facteurs d'une multiplication, par exemple les commandes `3 7` ou `a x` donneront un message d'erreur. Il faut vraiment taper `3*7` ou `a*x`.

Vous pouvez modifier le contenu de n'importe quelle entrée en cliquant dessus. Dans ce cas un symbole délimitant l'entrée éditée apparaît à gauche. Nous pouvons exécuter l'entrée modifiée comme lors d'une saisie normale. Si vous utilisez seulement la touche <Entrée>, vous créez une nouvelle ligne au sein de l'entrée éditée.

Exercice 2. Modifier l'entrée `99*97`; en `99-97`;

Exercice 3. Exécuter les commandes suivantes et expliquer ce qu'elles font exactement.

Faites tout de suite l'effort de mémorisation des commandes utilisées ci-dessous : les deux possibilités pour la puissance, `sqrt(...)`, `exp`, `log` et non pas `ln` qui n'existe pas dans **Maxima**, la syntaxe pour les constantes `%pi`, `%e`...

1. `2**4`;
2. `2^5`;
3. `4/6`;
4. `sqrt(12)`;
5. `exp(log(5))`;
6. `%pi`;
7. `float(%pi);float(%e)`;

1.2 Variables

Maxima permet de gérer des variables. La syntaxe générale d'une affectation est :

`variable : expression`

Au cours de l'interprétation de la commande `n:10*5`; Maxima évalue l'expression `10*5` et affecte le résultat obtenu à la variable `n`, cette affectation est valable tout au long de la session Maxima sauf commande contraire, comme l'affectation d'une autre expression à la même variable.

Exercice 4. Exécuter les commandes suivantes et commenter

1. `x`;
2. `x:4`;
3. `x`;
4. `x*x-2`;
5. `x`;
6. `kill(x)`;
7. `x`;

Attention, le comportement de Maxima dépend de l'ordre chronologique des validations des commandes et non de l'ordre apparent. En effet, toute commande visible sur la feuille Maxima peut être modifiée ou exécutée à tout instant. Une commande visible à l'écran peut aussi n'avoir jamais été exécutée. C'est le cas par exemple, si vous plantez le logiciel et que vous utilisez l'onglet [Maxima->Redémarrer Maxima]. Dans ce cas, il vous faudra réeffectuer la suite des combinaisons `<Majuscule>+<Entrée>` nécessaires au bon déroulement de votre programme.

1.3 Aide de Maxima

Une chose qu'il faut absolument savoir faire avec un logiciel tel que Maxima est d'utiliser l'aide disponible par l'intermédiaire du bouton **Help** puis **Maxima Help** de la barre de menu. Vous pouvez consulter l'aide de 4 façons différentes :

1. en noircissant à la souris un mot présent dans vos saisies Maxima et en appuyant sur la touche F1.
2. par thème à l'aide de l'onglet **content**;
3. par nom de commande à l'aide de l'onglet **index**;
4. par recherche dans les différents textes d'aide à l'aide de l'onglet **search**;

La plupart du temps nous utiliserons les méthodes 1 ou 3.

Le texte d'aide d'une commande comporte les syntaxes possibles, un descriptif puis des exemples. Évidemment, tout est en anglais.

Exercice 5. En utilisant l'aide de Maxima effectuer les tâches suivantes. Le mot entre crochet vous permettra de consulter l'aide.

1. Développer l'expression $(x - 1) * (x + 1) * x$ [**expand**].
2. Résoudre $x^3 - 6x^2 + 11x - 6 = 0$ [**solve**].
3. Tracer le graphe de $x \mapsto \sin(x^2)$ sur $[-\pi, \pi]$ [**plot2d**].

De manière générale, si une nouvelle commande est à utiliser pour résoudre un exercice, le nom de celle-ci sera rajouté entre crochets.

2 Fonctions

Il n'est pas évident de délimiter précisément la notion de fonction dans un logiciel de calcul symbolique. Disons simplement qu'une fonction de **Maxima** est un objet qui crée une expression à partir de variables.

2.1 Déclaration

Le nombre de fonctions déjà intégrées à **Maxima** est important (`sin,cos,exp,log,...`), mais on a souvent besoin de définir, au moins momentanément, ses propres fonctions. La syntaxe la plus simple pour cela est:

```
nom_fonction(variable_1,...,variable_n) := expression
```

Par exemple la commande `f(x):=x**2+1` crée la fonction $f : x \mapsto x^2 + 1$.

Attention, contrairement à l'affectation de variable on utilise le symbole `:=` et non le symbole `:`.

Après cette définition, on peut appliquer la fonction `f` à une expression quelconque, constante ou symbolique.

```
(%i2) f(5);
(%o2) 26
(%i3) f(3*a^2);
(%o3) 9a^4+1
```

Il faut bien faire la différence entre fonctions et expressions. La commande `expr:x**2+1` définit une expression et non pas une fonction. A partir d'une expression `expr` dépendant de `x` on crée une fonction `f` par `define(f(x),expr)`. Inversement, on obtient une expression à partir d'une fonction `f` en évaluant `f` en une indéterminé : `expr:f(x)`.

Pour évaluer une fonction `f(x)` en `x=5` on appelle `f(5)` ou encore `at(f(x),x=5)`. Pour une expression `expr` on utilise la commande `subst : subst([x=5],expr)`.

Exercice 6.

1. Tester vous même ces exemples.
2. Tester de même et comprendre : `kill(all);expr:x**4+1;x:5;expr;`

2.2 Dérivation

La commande `diff(expr,var)` permet de dériver une expression `expr` par rapport à la variable `var` :

Exercice 7. Tester les commandes suivantes et commenter :

1. `t:x**3+2*x**2+x-1;`
2. `dt:diff(t,x);`
3. `subst([x=3],t);`
4. `subst([x=3],dt);`
5. `expr:x**3+2*x**2+x-1;`
6. `define(g(x),expr);`
7. `define(dg(x),diff(g(x),x));`
8. `g(3);`
9. `dg(3);`

Attention l'approche suivante ne marche pas

```
f(x):=x**3+2*x**2+x-1;
df(x):=diff(f(x),x);
df(3); --> Erreur
```

Exercice 8. Créer une fonction `f` pour $f(x) = \cos(x^2 + \ln(x))$. Créer la fonction `df`, dérivée de `f`. Vérifier que **Maxima** donne une valeur quand vous lui demandez de calculer $f'(1)$.

2.3 Intégration

Pour calculer une primitive on utilise la commande `integrate(expr,var)`. Pour calculer une intégrale définie, on utilise la commande `integrate(expr,var,a,b)`.

Si vous désirez faire apparaître une forme élégante de votre résultat, vous pouvez utiliser une forme inerte en faisant précéder la commande `integrate` par un accent `'`. Vous pourrez aussi les utiliser pour la commande `limit`. Par exemple, essayez:

```
'integrate (3+3*x+x^2+x^4, x)=integrate (3+3*x+x^2+x^4, x);
```

Exercice 9. Calculer les primitives ou intégrales suivantes :

1. $\int \cos(x)dx$.
2. $\int_0^1 \cos(x)dx$.
3. $\int \sin(x)^3 dx$.
4. $\int_0^\pi \cos(x)^2 \times e^x dx$.
5. $\int_{-\infty}^{+\infty} x^2 e^{-x^2} dx$. On pourra utiliser `minf` et `inf`

2.4 Limites

La commande pour calculer des limites est `limit(expr,var,a)`. Pour calculer une limite à gauche, on utilise `limit(expr,var,a,minus)`. Pour calculer une limite à droite on utilise `limit(expr,var,a,plus)`.

Exercice 10. Faire calculer les limites suivantes à Maxima.

1. $\lim_{x \rightarrow 0} \frac{x^2 + x}{x + 2}$.
2. $\lim_{x \rightarrow 0} \frac{1}{x^2}$.
3. $\lim_{x \rightarrow 0^+} \frac{1}{x}$.
4. $\lim_{x \rightarrow +\infty} \frac{(x^2 - 1)}{(x^2 + 1)}$.

3 Résoudre des équations

La principale commande pour résoudre des équations est `solve`. Exemple :

```
(%i4) kill(all);
(%o0) done
(%i1) eq1: x**2-2*x+1;
(%o1) ...
(%i2) solve(eq1);
(%o2) [x=1]
(%i3) multiplicities;
(%o3) [2]
```

On note que `solve(eq1)` est un raccourci pour `solve(eq1=0)`.

Essayons de résoudre l'équation $ax^2 + bx + c = 0$ pour a, b et c quelconques.

```
(%i4) f(x):=a*x**2+b*x+c;
(%o4) ...
(%i5) solve(f(x));
(%o6) error
```

Le problème est que $f(x)$ comprend 4 inconnues : a, b, c et x . Précisons à **Maxima** que l'équation doit être résolue par rapport à x .

Recopier les réponses de **Maxima**.

```
(%i7) sol:solve(f(x),x)
(%o7)

(%i8) sol[1];
(%o8)
```

Exercice 11. Résoudre les équations suivantes à l'aide de **Maxima**.

1. $x^2 = x + 1$.
2. $\cos(x) = \frac{\sqrt{2}}{2}$.
3. $x^3 + ax = -1$ par rapport à x .

Pour résoudre un système d'équations on utilise la commande `solve([eqn_1, ..., eqn_n], [x_1, ..., x_n])`.

Exercice 12. Résoudre dans \mathbb{R}^2 les systèmes d'équations suivants :

1.

$$(S) \begin{cases} x + 3y = 2 \\ 2x - y = 1. \end{cases}$$

2.

$$(S) \begin{cases} 2x + \frac{1}{3}y = 7 \\ 8x + \frac{4}{3}y = 28. \end{cases}$$