

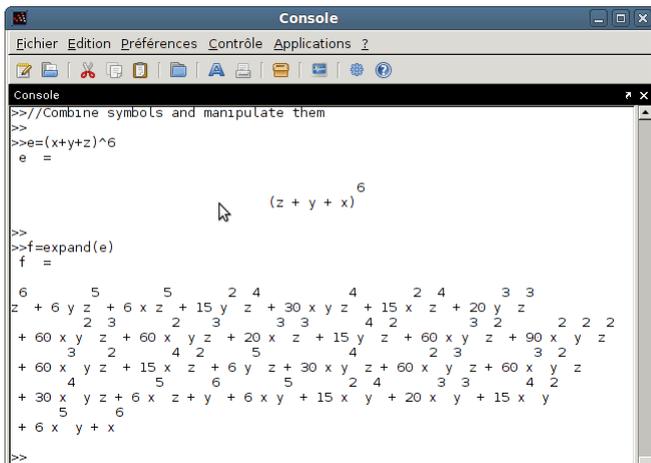
## Outils de Programmation 2 Calcul symbolique

### 1 SciMax Symbolic Toolbox

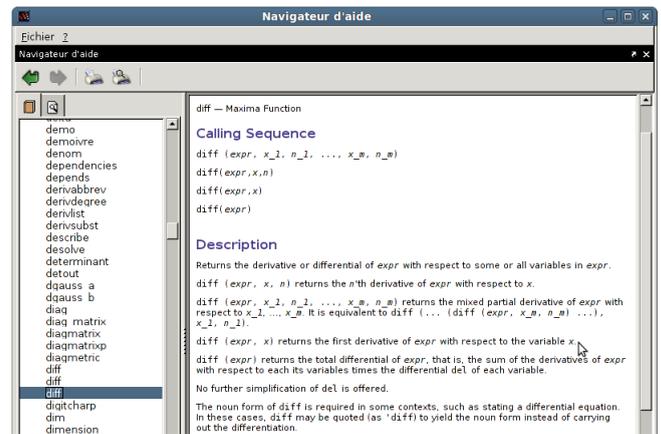
La boîte à outils symboliques (**SciMax Symbolic Toolbox**) qui utilise le noyau du logiciel Maxima <sup>1</sup>, permet de recourir au calcul formel depuis Scilab. son chargement intègre les possibilité de Maxima en matière de calcul formel à l’environnement Scilab. les scripts permettant d’accéder à ces possibilités ont des noms et une syntaxe qui paraîtra naturelle à l’utilisateur de Scilab.

Pour le moment, SciMax est uniquement disponible sous Linux. **Yann Collette** et **Allan Cornet** ont apporté des modifications pour avoir une version Windows, mais la boîte à outils doit être compilée avec Scilab built à partir de fichiers source.

Dans la mesure où les boîte à outils ne sont pas des parties intégrantes du logiciel Scilab, on peut les télécharger à partir de [https:// atoms.scilab.org](https://atoms.scilab.org).



Le calcul symbolique est naturellement intégré à Scilab.



La documentation Maxima est intégrée dans le navigateur d’aide Scilab.

### 2 Symbolic Math Toolbox

L’objectif de cette boîte à outils est de fournir une interface permettant de connecter MATLAB et Maple.

- La boîte à outils contient plus de 100 fonctions qui donnent accès au noyau Maple en utilisant une syntaxe et un style qui sont des extensions naturelles du langage MATLAB.
- La boîte à outils Symbolic Math Toolbox fournit les facilités numériques et graphiques de MATLAB avec plusieurs autres types de calcul mathématiques.

<sup>1</sup>Maxima est un logiciel de symboliques et numériques

## 2.1 Objets symboliques

- La boîte à outils de calcul symbolique définit un nouveau type de variable MATLAB appelé objet symbolique.
- Un objet symbolique est une structure de données qui enregistre une représentation du symbole sous la forme d'une chaîne.
- La boîte à outils utilise des objets symboliques pour représenter
  1. des variables symboliques
  2. des expressions symboliques
  3. des matrices symboliques.
- Les calculs symboliques sont effectués par Maple.

## 2.2 Calcul symbolique avec MATLAB

L'exemple suivant illustre la différence entre des types de variable standard de MATLAB et l'objet correspondant.

La commande MATLAB `sqrt(2)` renvoie le nombre suivant en virgule flottante:

```
ans = 1.4142
```

Si on convertit 2 en un objet symbolique à l'aide de la commande `sym`, on obtient:

```
>> a=sqrt(sym(2))  
a =  
2^(1/2)
```

On peut obtenir la valeur numérique d'un objet à l'aide de la commande `double` :

```
>> double(a) % valeur d'un objet symbolique  
ans =  
1.4142
```

Quand on crée une fraction utilisant des objets symboliques, MATLAB enregistre le numérateur et le dénominateur.

```
>> sym(2)/sym(5)  
ans =  
2/5
```

MATLAB ne fait pas les calculs de la même manière sur les variables de type double et les objets symboliques.

```
>> 2/5 + 1/3 %variable double  
ans =  
0.7333  
  
>> sym(2)/sym(5) + sym(1)/sym(3) % objet symbolique  
ans =  
11/15
```

### 3 Création de variables et d'expressions symboliques

La commande `sym` permet de construire:

**Des variables symboliques**

```
>> x = sym('x') % une variable symbolique x
>> a = sym('alpha') % une variable symbolique alpha
```

**Des expressions symboliques**

Le rapport  $\rho = \frac{1+\sqrt{5}}{2}$  est créé par la commande

```
>> rho = sym('(1+sqrt(5))/2')
```

On peut faire des opérations mathématiques telles que :

```
>> f = rho^2-rho-1
f=
(1/2 + 1/2 *5^(1/2))^2-3/2-1/2*5^(1/2)
```

**Exemple: Création de la fonction  $ax^2 + bx + c$**

```
>> f = sym('a*x^2+b*x+c')
f=
a*x^2+b*x+c
```

Dans ce cas, il n'y a pas création de variables correspondant aux termes  $a, b, c$  et  $x$ . Pour faire des opérations de calcul symbolique (différentiation, intégration, substitution, ...) sur  $f$  on doit créer ces variables explicitement.

```
>> a = sym('a'); b = sym('b'); c = sym('c'); x = sym('x');
```

ou plus simplement

```
>> syms a b c x
>> f = a*x^2+b*x+c
f=
a*x^2+b*x+c
```

Il est préférable d'utiliser `syms`

### 4 Calcul différentiel et intégral

La commande `diff` permet d'obtenir la dérivée formelle d'une expression formelle.

- Création de l'expression symbolique :

```
>> syms a x
>> f = sin(a*x)
f=
sin(a*x)
```

- Calcul de sa dérivée

```
>> diff(f) % calcul de la derivee de f par rappot a x
ans=
cos(a*x)*a
```

- Pour calculer la dérivée par rapport à  $a$  :

```
>> diff(f,a)
ans=
cos(a*x)*x
```

- Pour calculer la dérivée seconde par rapport à  $x$  et  $a$  :

```
>> diff(f,x,2)
ans=
-sin(a*x)*a^2
>> diff(f,a,2)
ans=
-sin(a*x)*x^2
```

La fonction `diff` peut également agir sur une matrice. Dans ce cas la dérivation est effectuée sur chaque élément de la matrice. Considérons l'exemple :

```
>> syms a x
>> A = [cos(a*x), sin(a*x) ; -sin(a*x), cos(a*x)]
A=
[ cos(a*x), sin(a*x)]
[-sin(a*x), cos(a*x)]
```

La commande `diff(A)` donne :

```
>> diff(A)
ans=
[ -sin(a*x)*a, cos(a*x)*a]
[-cos(a*x)*a, -sin(a*x)*a]
```

La commande `int` permet de calculer des primitives. Voici un premier exemple

```
>> syms a b t x y z theta
>> int(sin(a*t+b))
ans =
-cos(b + a*t)/a
>> int(sin(a*theta+b),theta)
ans =
-cos(b + a*theta)/a
>> int(x*y^2+y*z,y)
ans =
(y^2*(3*z + 2*x*y))/6
>> int(x^2*sin(x))
ans =
2*cos(x) - x^2*cos(x) + 2*x*sin(x)
```

Comme pour `diff`, la variable d'intégration peut être omise, auquel cas MATLAB va choisir cette variable automatiquement. Dans certains cas, MATLAB est incapable de proposer une réponse faisant intervenir des fonctions élémentaires. C'est le cas si l'on demande la primitive de  $e^{-x^2}$  ou  $\sqrt{1+x^3}$ . Dans le premier cas fait intervenir la fonction d'erreur `erf` (qui est définie par une intégration) dans la réponse. Dans le second cas, la primitive est ramenée à une autre primitive par intégration par partie.

Bien entendu, MATLAB sait aussi calculer des intégrales définies.

Pour calculer  $\int_0^\pi \sin x dx$ , on peut écrire

```
>> int(sin(x),0,pi)
```

ou:

```
>> int(sin(theta),theta,0,pi)
```

Dans la première instruction,  $x$  est retenu par défaut comme variable d'intégration alors que la seconde instruction explicite la variable d'intégration.

On peut utiliser la commande `vpa` pour obtenir une précision arbitraire sur la valeur numérique désirée:

```
>> int(exp(-x^2),0,inf)
ans =
pi^(1/2)/2
>> vpa(ans,25)

ans=
0.8862269254527580136490837
```