Introduction à Maxima

PAR MARC GILG

Lug 68
http://lug68.free.fr

16 mai 2002

Version 0.1.3

Résumé

Ce document présente en quelques pages le logiciel de calcul formel **Maxima.** Son but est de donner quelques fonctions pour pouvoir utiliser **Maxima** rapidement. Par conséquent, il est incomplet et n'aborde pas l'aspect programmation.

1 Introduction

1.1 Fonctionnalités

Le programme **Maxima** est un logiciel permettant de :

- faire du calcul numérique à n'importe quel précision
- faire du calcul symbolique
- faire du calcul matriciel
- Tracer des courbes et des surfaces
- programmer ses tâches de calcul

C'est un logiciel libre sous licence GPL que l'on peut charger à l'adresse suivante :
http://www.ma.utexas.edu/maxima.html

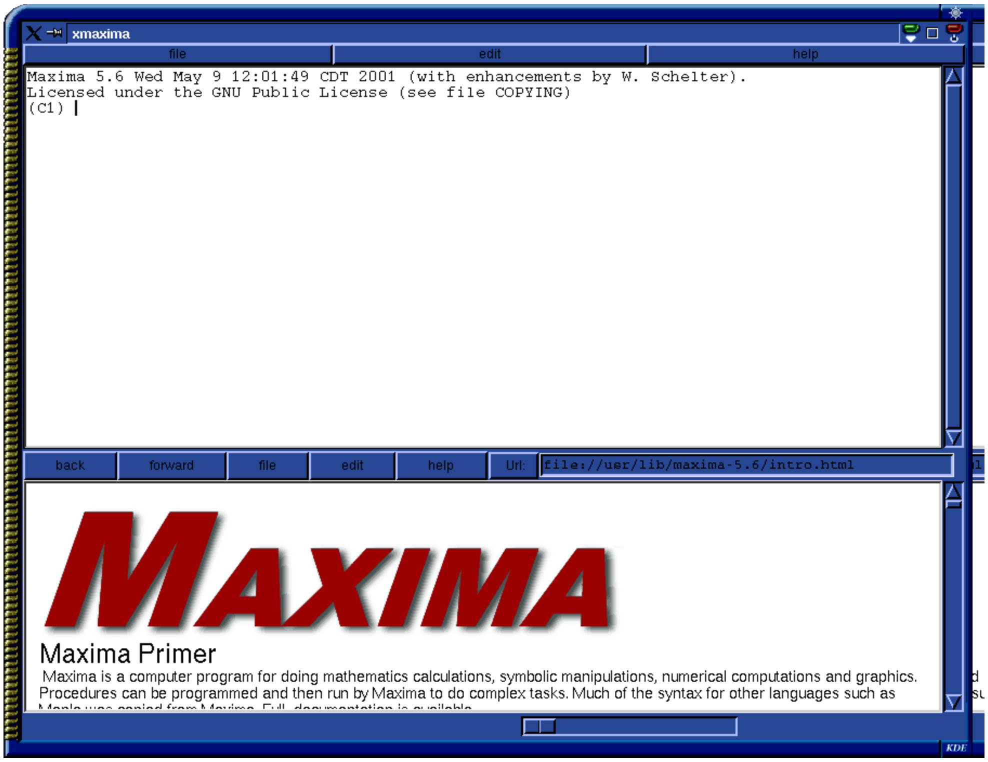
Si vous utilisez Linux Mandrake ou RedHat, alors je vous conseille d'installer le package **rmp** qu'on peut trouver à l'adresse suivante :
http://fr.rpmfind.net

1.2 Exécution de Maxima.

On peut exécuter **Maxima** d'au moins 3 manières différentes :

1. Dans un terminal texte en tapant `maxima`
2. Dans Xwindows en tapant la commande `xmaxima`
3. Dans l'éditeur TeXmacs par le menu `Insérer>session>maxima`

Voici un aperçu de `xmaxima` :



Dans ce document, écrit avec TeXmacs, nous allons utiliser des sessions maxima :

```
GCL (GNU Common Lisp)  Version(2.4.0) Wed May  9 12:02:00 CDT 2001
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)

(C1) quit();

The end
```

Pour quitter la session, cliquer sur l'icône  puis cliquer sous le texte de la session **Maxima** à l'endroit du nouveau paragraphe.

2 Généralités

2.1 Obtenir de l'aide.

La fonction `describe("`*function*`")` permet d'obtenir une description de *function*. Cette fonction pose quelques problèmes si on l'utilise via TeXmacs.

La fonction `example(`*function*`)` donne des exemples d'utilisation d'une fonction :

Exemple 1.

```
GCL (GNU Common Lisp)  Version(2.4.0) Wed May  9 12:02:00 CDT 2001
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
```

```
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)
```

(C1) example(log);

Not Found. You can look at:
(D1) [FUNCTIONS, ARRAYS, LAMBDA, LISTS, MATRICES, EQUATIONS, IF, BLOCK, DO, EVALUATION, EXP, TRIG, COMPLEX, EV, ZEROEQUIV, EXPAND, RATEXPAND, RATSIMP, RADCAN, COMBINE, MULTTHRU, XTHRU, PARTFRAC, FACTOR, FACTORSUM, SQFR, GFACTOR, PARTITION, LOGCONTRACT, ROOTSCONTRACT, DIFF, DEPENDS, GRADEF, INTEGRATE, RISCH, CHANGEVAR, SPECINT, PART, INPART, NOUNIFY, DPART, SUBSTITUTE, RATSUBST, SUBSTPART, SUBSTINPART, ATVALUE, AT, LISTOFVARS, COEFF, RATCOEFF, BOTHCOEFF, ISOLATE, PICKAPART, NUMFACTOR, DERIVDEGREE, REALPART, POLARFORM, DELETE, NROOTS, REALROOTS, ALLROOTS, LINSOLVE, ALGSYS, SOLVE, ENTERMATRIX, GENMATRIX, AUGCOEFMATRIX, ECHELON, TRIANGULARIZE, RANK, CHARPOLY, DOTSCRULES, "/ /", RATWEIGHT, HORNER, DIVIDE, CONTENT, RESULTANT, BEZOUT, $\mathrm{POLY_DISCRIMINANT}$, RATDIFF, TELLRAT, TAYTORAT, SUM, PRODUCT, LIMIT, NUSUM, FUNCSOLVE, RESIDUE, TAYLOR, DEFTAYLOR, POWERSERIES, TRIGEXPAND, TRIGREDUCE, OPTIMIZE, LAPLACE, ILT, MINFACTORIAL, FACTCOMB, QUNIT, CF, CFDISREP, CFEXPAND, FEATUREP, MAP, FULLMAP, FULLMAPL, SCANMAP, APPEND, REVERSE, DISPLAY, REVEAL, CATCH, ORDERLESS, ORDERGREAT, UNORDER, LINEAR, ADDITIVE, OUTATIVE, MULTIPLICATIVE, LASSOCIATIVE, RASSOCIATIVE, COMMUTATIVE, SYMMETRIC, ANTISYMMETRIC, NARY, ODDFUN, EVENFUN, POSFUN, ARRAYINFO, PROPERTIES, PRINTPROPS, PROPVARS, GET, IS, FREEOF, MATCHDECLARE, TELLSIMP, DEFMATCH, LET, LETRULES, POISSIMP, ODE2, SCSIMP, ELIMINATE, DESOLVE, SYNTAX]

(C2) Quit();

The end

2.2 Principes de base.

Exemple 2.

Voici un simple calcul :

```
GCL (GNU Common Lisp)  Version(2.4.0) Wed May  9 12:02:00 CDT 2001
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)
```

(C1) 1+1;

(D1) 2

(C2) quit();

The end

On remarque que les lignes sont numérotées $(C1)$ et $(D1)$ .... :

- $(C1), ..., $(Cn) représente les lignes de commandes

- $(D1),\ldots.,$ (Dn) représente les lignes de résultats

Cette notation permet de rappeler les commandes, il faut utiliser deux `'` précédant les numéros d'une commande, par exemple `''C1`.

Pour qu'une instruction soit exécutée, elle doit se terminer par `;` ou $, si elle se termine par $ alors le résultat n'est pas affiché. Pour quitter **Maxima** il faut taper l'instruction `QUIT();` Les fonctions ne sont pas sensibles à la case, mais les variables le sont.

Exemple 3.

```
GCL (GNU Common Lisp)  Version(2.4.0) Wed May  9 12:02:00 CDT 2001
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)

(C1) x :1;

(D1) 1

(C2) X :0$
(C3) x;

(D3) 1

(C4) Quit()$

The end
```

Remarquer que l'affectation se fait par le symbole `":"`.

3 Arithmetique

Voici la façon de faire les principales opérations :

Opération	**Symbole Maxima**
Addition	+
Soustraction	-
Multiplication	*
Division	/
Puissance	^ ou **
Multiplication Matriciel	.
Racine carrée	sqrt()

Exemple 4.

```
GCL (GNU Common Lisp)  Version(2.4.0) Wed May  9 12:02:00 CDT 2001
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)

(C1) 1/2 + 1/3;
```

(D1) $\frac{5}{6}$

```
(C2) 132 ^ (1/2);
```

(D2) $2\sqrt{33}$

```
(C3) Quit();

The end
```

Remarquer que **Maxima** fait des calculs exacts et qu'il simplifie les expressions. Pour obtenir un calcul numérique, il faut rajouter `,numer` La précision des calculs est donnée par la variable `fpprec` Pour que les calculs se fassent avec une grande précision, il faut utiliser la fonction `bfloat()`.

Exemple 5.

```
GCL (GNU Common Lisp)  Version(2.4.0) Wed May  9 12:02:00 CDT 2001
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)

(C1) fpprec ;

(D1) 16

(C2) bfloat(1/234);

4.273504273504274B-3

(C3) fpprec :100;

(D3) 100

(C4) ''C2;

4.273504273504273504273504273504273504273504273504273504273504273504273504273504273504273504273504274B-3

(C5) Quit();

The end
```

4 Algèbre

4.1 Manipulation de polynômes et de fractions rationnelles

Le programme **Maxima** permet de développer, de factoriser des polynômes à coefficients entiers à l'aide des fonctions `expand()` et `factor()`. Les fractions rationnelles peuvent être réduites au même dénominateur par la fonction `ratsimp()`.

```
GCL (GNU Common Lisp)  Version(2.4.0) Wed May  9 12:02:00 CDT 2001
Licensed under GNU Library General Public License
```

```
Contains Enhancements by W. Schelter
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)
```

(C1) `expand((x+1)^2+(3*x+5)^3);`

(D1) $27\,x^3+136\,x^2+227\,x+126$

(C2) `factor(%);`

(D2) $(x+2)\left(27\,x^2+82\,x+63\right)$

(C3) `f : ((x+3)^2+5)/(x-5) + x^3/(x^2+1);`

(D1) $\dfrac{(x+3)^2+5}{x-5}+\dfrac{x^3}{x^2+1}$

(C2) `ratsimp(%);`

(D2) $\dfrac{2\,x^4+x^3+15\,x^2+6\,x+14}{x^3-5\,x^2+x-5}$

Remarquer que les calculs sont faits en réel. La fonction **factor()** peut avoir un second argument qui est un polynôme irréductible. Ceci permet d'étendre l'anneau sur lequel on fait la factorisation.

Exemple 6.

```
GCL (GNU Common Lisp)  Version(2.4.0) Wed May  9 12:02:00 CDT 2001
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)
```

(C1) `factor(X^4-1,I^2+1);`

(D1) $(X-1)\,(X+1)\,(X-I)\,(X+I)$

(C2) `Quit();`

The end

On peut calculer le quotient de deux polynômes avec la fonction `quotient(p1,p2,var);`

Exemple 7.

```
GCL (GNU Common Lisp)  Version(2.4.0) Wed May  9 12:02:00 CDT 2001
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)
```

(C1) `QUOTIENT(X^3+3*X^2+3*X+1,X+1,X);`

(D1) $X^2+2\,X+1$

```
(C2) Quit();

The end
```

4.2 Résolution d'équations

4.2.1 Résolution de système polynomial.

La fonction `ALGSYS([p1,p2,...],[var1,var2,...])` permet de résoudre le système de `p1,p2,...` polynômes avec `var1,var2,...` variables.

Exemple 8.

```
GCL (GNU Common Lisp)  Version(2.4.0) Wed May  9 12:02:00 CDT 2001
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)

(C1) P1 : X^2+Y^2-1$
(C2) P2 : X+Y$
(C3) ALGSYS([P1,P2],[X,Y]);
```

(D3) $\left[\left[X=-\frac{1}{\sqrt{2}},Y=\frac{1}{\sqrt{2}}\right],\left[X=\frac{1}{\sqrt{2}},Y=-\frac{1}{\sqrt{2}}\right]\right]$

```
(C4)
```

La fonction `ALLROOTS(poly)` donne les racines d'un polynôme.

Exemple 9.

```
GCL (GNU Common Lisp)  Version(2.4.0) Wed May  9 12:02:00 CDT 2001
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)

(C1) ALLROOTS(X^4-1);
```

(D1) $[X=1.0, X=-1.0, X=1.0\,i, X=-1.0\,i]$

```
(C2)
```

Remarque 10. Les coefficients ne sont pas des entiers mais des réels. Le calcul se fait donc de manière approché.

4.2.2 Résolutions d'équations algébrique

La fonction `solve(expr,var)` permet de résoudre des équations algébriques.

Exemple 11.

```
GCL (GNU Common Lisp)  Version(2.4.0) Wed May  9 12:02:00 CDT 2001
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)
```

```
(C1) SOLVE(COS(X)^3-1,X);


SOLVE is using arc-trig functions to get a solution.
Some solutions will be lost.
```

(D1) $\left[X=\arccos\left(\frac{\sqrt{3}\,i}{2}-\frac{1}{2}\right), X=\pi-\arccos\left(\frac{\sqrt{3}\,i}{2}+\frac{1}{2}\right), X=0\right]$

```
(C2) SOLVE(X^4-1);
```

(D2) $[X=i, X=-1, X=-i, X=1]$

```
(C3) SOLVE(COS(X)*SIN(X)-1,X);
```

(D3) $\left[\sin X=\frac{1}{\cos X}\right]$

```
(C4) TRIGREDUCE(COS(X)*SIN(X));
```

(D4) $\frac{\sin(2\,X)}{2}$

```
(C5) SOLVE(%-1,X);


SOLVE is using arc-trig functions to get a solution.
Some solutions will be lost.
```

(D5) $\left[X=\frac{\arcsin 2}{2}\right]$

```
(C6)
```

Remarque 12. Le résultat (D2) montre que nous obtenons des solutions exactes, contrairement au cas précédent. Le dernier exemple montre que selon la forme de l'équation nous obtenons des solutions différentes.

La fonction `solve()` peut être contrôlée pas certaines variables. Par exemple `mettre SOLVERADCAN` à TRUE permet d'obtenir de meilleurs résultats avec les fonctions logarithmes et exponentielles.

Exemple 13.

```
GCL (GNU Common Lisp)  Version(2.4.0) Wed May  9 12:02:00 CDT 2001
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)

(C1) SOLVE(5^X-125,X);
```

(D1) $\left[X=\frac{\log 125}{\log 5}\right]$

```
(C2) SOLVERADCAN : TRUE $
(C3) ''C1;
```

(D3) $[X=3]$

```
(C4)
```

Remarque 14. Vous trouverez d'autres variables pour contrôler `solve()` dans le manuel de **Maxima**.

4.2.3 Résolutions d'équations linéaires

La fonction `LINSOLVE([L1,L2,...],[var1,var2,...])` permet de résoudre des systèmes linéaires.

Exemple 15.

```
GCL (GNU Common Lisp)  Version(2.4.0) Wed May  9 12:02:00 CDT 2001
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)

(C1) L1 : X+Z=Y$
(C2) L2 : 2*A*X-Y=2*A^2$
(C3) L3 : Y-2*Z=2$
(C4) LINSOLVE([L1,L2,L3],[X,Y,Z]);
```

(D4) $[X = A + 1, Y = 2\,A, Z = A - 1]$

```
(C5) GLOBALSOLVE : TRUE$
(C6) ''C4;
```

(D6) $[X : A + 1, Y : 2\,A, Z : A - 1]$

```
(C7)
```

Remarque 16. Si variable `GLOBALSOLVE` est à TRUE, alors la solution est donnée comme une affectation aux variables.

4.3 Calcul Matriciel.

4.3.1 Définition d'une matrice

On peut saisir une matrice à l'aide de la fonction `entermatrix(ligne,colonnes)`.

Exemple 17.

```
GCL (GNU Common Lisp)  Version(2.4.0) Wed May  9 12:02:00 CDT 2001
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)

(C1) m : entermatrix(3,3);

Is the matrix  1. Diagonal  2. Symmetric  3. Antisymmetric  4. General

Answer 1, 2, 3 or 4 : 4
Row 1 Column 1: 1;
Row 1 Column 2: 0;
Row 1 Column 3: -1;
Row 2 Column 1: 0;
Row 2 Column 2: 1;
```

```
Row 2 Column 3: a;
Row 3 Column 1: 1;
Row 3 Column 2: -1;
Row 3 Column 3: a;

Matrix entered.
```

(D1) $\begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & a \\ 1 & -1 & a \end{pmatrix}$

(C2)

La fonction `MATRIX(L1,L2,...)` permet à partir des listes `L1,L2,...` de créer une matrice avec comme lignes `L1,L2,...`

Exemple 18.

```
GCL (GNU Common Lisp)  Version(2.4.0) Wed May  9 12:02:00 CDT 2001
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)

(C1) L1 : [1,0,-1]$
(C2) L2 : [0,1,a]$
(C3) L3 : [1,-1,a]$
(C4) m : MATRIX(L1,L2,L3);
```

(D4) $\begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & a \\ 1 & -1 & a \end{pmatrix}$

(C5)

La fonction `DIAGMATRIX(n,X)` permet de créer une matrice $n \times n$ diagonale avec X comme élément diagonal. La fonction `IDENT(n)` permet de créer une matrice identité de taille $n \times n$.

Exemple 19.

```
GCL (GNU Common Lisp)  Version(2.4.0) Wed May  9 12:02:00 CDT 2001
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)

(C1) d : DIAGMATRIX(5,COS(X));
```

(D1) $\begin{pmatrix} \cos X & 0 & 0 & 0 & 0 \\ 0 & \cos X & 0 & 0 & 0 \\ 0 & 0 & \cos X & 0 & 0 \\ 0 & 0 & 0 & \cos X & 0 \\ 0 & 0 & 0 & 0 & \cos X \end{pmatrix}$

(C2) `IDENT(2);`

(D2) $\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$

La fonction `GENMATRIX(Tableau,i2,j2,i1,j1)` permet de créer une matrice à partir d'un tableau tel que `Tableau(i1,j1)` soit le premier élément (en haut a gauche) et `Tableau(i2,j2)` soit le dernier élément (en bas à droite). Si $j1=i1$ alors on peut omettre $j1$ et si $j1=i1=1$ on peut omettre $i1$ et $j1$.

Exemple 20.
```
GCL (GNU Common Lisp)  Version(2.4.0) Wed May  9 12:02:00 CDT 2001
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)

(C1) H[I,J]:=1/(I+J+1)$
(C2) GENMATRIX(H,4,4);
```

(D2) $$\begin{pmatrix} \frac{1}{3} & \frac{1}{4} & \frac{1}{5} & \frac{1}{6} \\ \frac{1}{4} & \frac{1}{5} & \frac{1}{6} & \frac{1}{7} \\ \frac{1}{5} & \frac{1}{6} & \frac{1}{7} & \frac{1}{8} \\ \frac{1}{6} & \frac{1}{7} & \frac{1}{8} & \frac{1}{9} \end{pmatrix}$$

(C3)

4.3.2 Fonctions élémentaires

La fonction `DETERMINANT(m)` permet de calculer le déterminant de m. La fonction `TRANSPOSE(m)` permet de calculer sa transposée. `TRIANGULARIZE(m)` triangularise m qui n'est pas forcément une matrice carrée. `INVERT(m)` calcule l'inverse de la matrice carrée m. Si l'on rajoute l'option `,detout` alors l'inverse du déterminant de m est mis en facteur. La fonction `RANK(m)` calcule le rang de m, dans certain cas cette fonction donne un résultat faux.

Exemple 21.
```
GCL (GNU Common Lisp)  Version(2.4.0) Wed May  9 12:02:00 CDT 2001
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)

(C1) L1 : [1,0,-1]$
(C2) L2 : [0,1,a]$
(C3) L3 : [-1,0,a]$
(C4) m : MATRIX(L1,L2,L3);
```

(D4) $$\begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & a \\ -1 & 0 & a \end{pmatrix}$$

```
(C5) TRANSPOSE(M);
```

(D5) $\mathrm{TRANSPOSE}(M)$

```
(C6) TRANSPOSE(m);
```

(D6) $$\begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 0 \\ -1 & a & a \end{pmatrix}$$

```
(C7) DETERMINANT(m);
```

(D7) $a-1$

(C8) RANK(m);

(D8) 3

(C9) INVERT(m);

(D9) $\begin{pmatrix} \frac{a}{a-1} & 0 & \frac{1}{a-1} \\ -\frac{a}{a-1} & 1 & -\frac{a}{a-1} \\ \frac{1}{a-1} & 0 & \frac{1}{a-1} \end{pmatrix}$

(C10) INVERT(m),detout;

(D10) $\frac{\begin{pmatrix} a & 0 & 1 \\ -a & a-1 & -a \\ 1 & 0 & 1 \end{pmatrix}}{a-1}$

(C11) TRIANGULARIZE(m);

(D11) $\begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & a \\ 0 & 0 & a-1 \end{pmatrix}$

(C12)

La fonction ADJOINT(M) calcule la matrice adjoint de M. La fonction ECHELON(M) échelonne la matrice M.

Exemple 22. GCL (GNU Common Lisp) Version(2.4.0) Wed May 9 12:02:00 CDT 2001
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)

(C1) L1 : [1,0,-1]$
(C2) L2 : [0,1,a]$
(C3) L3 : [-1,0,a]$
(C4) M : MATRIX(L1,L2,L3);

(D4) $\begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & a \\ -1 & 0 & a \end{pmatrix}$

(C5) ADJOINT(M);

(D5) $\begin{pmatrix} a & 0 & 1 \\ -a & a-1 & -a \\ 1 & 0 & 1 \end{pmatrix}$

(C6) ECHELON(M);

(D6) $\begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & a \\ 0 & 0 & 1 \end{pmatrix}$

4.3.3 Manipulation de colonnes et de lignes

La fonction `ADDCOL(M,L1,L2,...)` permet d'ajouter à M les colonnes données par les listes $L1$, $L2, \ldots$. La fonction `ADDROW(M,L1,L2,...)` permet d'ajouter à M les lignes données par les listes $L1, L2, \ldots$. La fonction `COL(M,i)` (resp. `ROW(M,i)`) permet d'extraire la $i^{\text{éme}}$ colonne (resp. ligne) de M.

Exemple 23.
```
GCL (GNU Common Lisp)  Version(2.4.0) Wed May  9 12:02:00 CDT 2001
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)
```

(C1) `m: MATRIX([1,2]);`

(D1) $\begin{pmatrix} 1 & 2 \end{pmatrix}$

(C2) `ADDCOL(m,[3]);`

(D2) $\begin{pmatrix} 1 & 2 & 3 \end{pmatrix}$

(C3) `ADDROW(m,[4,5,6]);`

```
Incompatible structure - ADDROW//ADDCOL
 -- an error.  Quitting.  To debug this try DEBUGMODE(TRUE);)
```

(C4) `n : ADDROW(m,[4,5]);`

(D4) $\begin{pmatrix} 1 & 2 \\ 4 & 5 \end{pmatrix}$

(C5) `COL(n,2);`

(D5) $\begin{pmatrix} 2 \\ 5 \end{pmatrix}$

(C6) `ROW(n,1);`

(D6) $\begin{pmatrix} 1 & 2 \end{pmatrix}$

(C7)

Remarque 24. Les fonctions `ADDCOL(M,L)` et `ADDROW(M,L)` ne modifient pas le contenu de M.

4.3.4 Valeurs propres et vecteurs propres : le pacage eigen

Dans cette rubrique, on va utiliser le package `eigen`. Pour cela on utilise la commande `LOAD(EIGEN)`. Cette bibliothèque contient des fonctions pour le calcul de vecteurs propres et de valeurs propres. On peut obtenir une description de ce package à l'aide de la fonction `PRINTFILE("eigen.usg")`. Pour le moment, le descriptif ne fonctionne que dans `xmaxima`.

La fonction `COLUMNVECTOR(L)` permet de définir un vecteur donné par la liste L. La fonction `INNERPRODUCT(L1,L2)` permet de calculer le produit scalaire ou hermitien des vecteurs $L1$ et $L2$.

Exemple 25.
```
GCL (GNU Common Lisp)  Version(2.4.0) Wed May  9 12:02:00 CDT 2001
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)

(C1) LOAD(EIGEN);

Warning - you are redefining the MACSYMA function EIGENVALUES
Warning - you are redefining the MACSYMA function EIGENVECTORS
```
(D1) /usr/lib/maxima-5.6/share/eigen.mc

(C2) `V : COLUMNVECTOR([a,b,c]);`

(D2) $\begin{pmatrix} a \\ b \\ c \end{pmatrix}$

(C3) `INNERPRODUCT(V,[1,2,3]);`

(D3) $\begin{pmatrix} a & 2a & 3a \\ b & 2b & 3b \\ c & 2c & 3c \end{pmatrix}$

(C4) `INNERPRODUCT([a,b,c],[1,2,3]);`

(D4) $3c+2b+a$

(C5)

Remarque 26. Si les arguments de `INNERPRODUCT(V,[a,b,c])` sont un vecteur et une liste, alors le résultat est la matrice formée par les colonnes $(a.V,\ b.V,\ c.V)$. En fait, cette fonction calcule simplement le produit $\bar{V}.(a,b,c)$.

La fonction `UNITVECTOR(L)` permet de créer un vecteur unitaire. La fonction `EIGENVALUES(M)` donne la liste des valeurs propres de M et la liste de leur multiplicité. La fonction `CHARPOLY(M,var)` permet de calculer le polynôme charatéristique de M avec la variable var. La fonction `EIGENVECTORS(M)` permet de calculer le résultat de la fonction `EIGENVALUES()` avec les vecteurs propres en prime.

Exemple 27.

```
GCL (GNU Common Lisp)  Version(2.4.0) Wed May  9 12:02:00 CDT 2001
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)
```

(C1) `m : MATRIX([1,0,1],[0,1,2],[0,0,2]);`

(D1) $\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 2 \end{pmatrix}$

(C2) `p : CHARPOLY(m,X);`

(D2) $(1-X)^2\,(2-X)$

```
(C3) EIGENVALUES(m);

Warning - you are redefining the MACSYMA function EIGENVALUES
Warning - you are redefining the MACSYMA function EIGENVECTORS
```

(D3) $[[1,2],[2,1]]$

```
(C4) EIGENVECTORS(m);
```

(D4) $[[[1,2],[2,1]],[1,0,0],[0,1,0],[1,2,1]]$

```
(C5) V : UNITVECTOR([1,2,1]);
```

(D5) $\left[\frac{1}{\sqrt{6}}, \frac{2}{\sqrt{6}}, \frac{1}{\sqrt{6}}\right]$

```
(C6) m.V;
```

(D6) $\begin{pmatrix} \frac{2}{\sqrt{6}} \\ \frac{4}{\sqrt{6}} \\ \frac{2}{\sqrt{6}} \end{pmatrix}$

```
(C7) ''c6/2;
```

(D7) $\begin{pmatrix} \frac{1}{\sqrt{6}} \\ \frac{2}{\sqrt{6}} \\ \frac{1}{\sqrt{6}} \end{pmatrix}$

```
(C8)
```

5 Analyse

5.1 Dérivation

5.1.1 Calcul de dérivées

La fonction `DIFF(f,x)` permet de calculer la dérivée de f par rapport à x. Elle se généralise en `DIFF(f(v1,v2,...),v1,n1,v2,n2,...)` où f est une fonction des variables $v1, v2, \ldots$ et la dérivation est d'ordre n_1 pour v_1, n_2 pour v_2, etc ... La fonction `AT(f(X,Y,..),[X=v1,Y=v2,...])` permet de calculer la valeur de la dérivée au point $(v_1, v_2, \ldots.)$.

Exemple 28.
```
GCL (GNU Common Lisp)  Version(2.4.0) Wed May  9 12:02:00 CDT 2001
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)

(C1) f : %E^(X^2)*cos(X)+X^2+3*X;
```

(D1) $e^{X^2}\cos X+X^2+3\,X$

(C2) DIFF(f,X);

(D2) $-e^{X^2}\sin X+2\,X\,e^{X^2}\cos X+2\,X+3$

(C3) DIFF(f,X,2);

(D3) $-4\,X\,e^{X^2}\sin X+4\,X^2\,e^{X^2}\cos X+e^{X^2}\cos X+2$

(C4) g : cos(Y)*f+1/Y;

(D4) $\left(e^{X^2}\cos X+X^2+3\,X\right)\cos Y+\frac{1}{Y}$

(C5) DIFF(g,Y);

(D5) $-\left(e^{X^2}\cos X+X^2+3\,X\right)\sin Y-\frac{1}{Y^2}$

(C6) DIFF(g,X,Y);

(D6) $\frac{d^Y}{d\,X^Y}\left(\left(e^{X^2}\cos X+X^2+3\,X\right)\cos Y+\frac{1}{Y}\right)$

(C7) DIFF(g,X,1,Y,1);

(D7) $-\left(-e^{X^2}\sin X+2\,X\,e^{X^2}\cos X+2\,X+3\right)\sin Y$

(C8) AT(''c7,[X=0,Y=%Pi/2]);

(D8) -3

(C9)

5.1.2 Equations différentielles

La fonction `'DIFF(f(v1,v2,...),v1,n1,v2,n2,...)` permet d'écrire les dérivées de $f(v_1,$ $v_{2,\ldots.})$. Attention, la fonction f doit être explicitement avec ses variables. Ceci permet d'écrire des équation différentielles. La fonction `DESOLVE([eq1,eq2,...,eqn],[var1,var2,...varn])` permet dans certain cas de résoudre ces équations. Si elle n'arrive pas, elle retourne le résultat `FALSE`.

La fonction `ATVALUE(g(x,y,...), [x=v1,y=v2,...],V)` permet de donner la valeur V à la fonction g au point $(v_1, v_2, \ldots)$. Ceci permet de définir des conditions initiales.

Exemple 29. GCL (GNU Common Lisp) Version(2.4.0) Wed May 9 12:02:00 CDT 2001
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)

(C1) df : 'DIFF(f(x),x);

(D1) $\frac{d}{d\,x}\,f(x)$

(C2) eq : df + 3* f(x) = 0;

(D2) $\frac{d}{d\,x}\,f(x)+3\,f(x)=0$

(C3) `DESOLVE(eq,f(x));`

(D3) $f(x)=f(0)\,e^{-3x}$

(C4) `ddf : 'DIFF(f(x),x,2);`

(D4) $\frac{d^2}{d\,x^2}\,f(x)$

(C5) `eq2 : ddf + 3 *df + f(x) = x^2+1;`

(D5) $\frac{d^2}{d\,x^2}\,f(x)+3\left(\frac{d}{d\,x}\,f(x)\right)+f(x)=x^2+1$

(C6) `DESOLVE(eq2,f(x));`

(D6) $f(x) = e^{-\frac{3x}{2}}\left(\frac{\sinh\left(\frac{\sqrt{5}\,x}{2}\right)\left(2\left(\frac{d}{dx}\,f(x)\Big|_{x=0}+3\,f(0)-45\right)-3\,(f(0)-17)\right)}{\sqrt{5}} + (f(0) - 17)\cosh\left(\frac{\sqrt{5}\,x}{2}\right)\right)+x^2-6\,x+17$

(C7) `ATVALUE(df,x=0,A);`

(D7) A

(C8) `''c6;`

(D8) $f(x) = e^{-\frac{3x}{2}}\left(\frac{(2\,(A+3\,f(0)-45)-3\,(f(0)-17))\sinh\left(\frac{\sqrt{5}\,x}{2}\right)}{\sqrt{5}} + (f(0) - 17)\cosh\left(\frac{\sqrt{5}\,x}{2}\right)\right)+x^2-6\,x+17$

(C9)

5.1.3 Développement de Taylor et limites.

La fonction `TAYLOR(f,var,pt,n)` calcule le développement de Taylor (ou de Laurent) de $f(\mathrm{var})$ au point pt à l'ordre n.

Exemple 30.
```
GCL (GNU Common Lisp)  Version(2.4.0) Wed May  9 12:02:00 CDT 2001
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)
```

(C1) `f : (cos(X)-1)/X;`

(D1) $\frac{\cos X-1}{X}$

(C2) `TAYLOR(f,X,0,3);`

(D2) $-\frac{X}{2}+\frac{X^3}{24}+\cdots$

(C3) g : sin(X)/(X^2);

(D3) $\frac{\sin X}{X^2}$

(C4) TAYLOR(g,X,0,2);

(D4) $\frac{1}{X}-\frac{X}{6}+\cdots$

(C5)

La fonction LIMIT(f,X,val,dir) calcule la limite de $f(X)$ quand $X\to$ val. dir peut prendre les valeurs PLUS ou MINUS selon que X tend vers val par valeur supérieure ou inférieure. On a aussi les mots INF pour $+\infty$, MINF pour $-\infty$ et INFINITY pour l'infini complexe. La fonction TLIMIT(f,X,val,dir) à la même fonctionnalité que LIMIT(), mais utilise les séries de Taylor.

Le résultat peut être UND pour un résultat indéfini et IND pour un résultat indéfini mais borné.

Exemple 31. GCL (GNU Common Lisp) Version(2.4.0) Wed May 9 12:02:00 CDT 2001
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)

(C1) LIMIT((COS(x)+1)/x,x,INF);

(D1) 0

(C2) LIMIT(1/X,X,0,PLUS);

(D2) ∞

(C3) LIMIT(1/X,X,0,MINUS);

(D3) $-\infty$

(C4) LIMIT(1/X,X,0);

(D4) UND

(C5) LIMIT(COS(Z),Z,INFINITY);

(D5) UND

(C6) LIMIT(COS(X),X,INF);

(D6) IND

(C7)

5.1.4 Equations aux dérivées partielles

Le package **ODE** permet de résoudre des EDP du premier et second ordre. La fonction **ODE2(edp,vard,vari)** permet de résoudre l'*edp* avec les variables dépendantes *vard* et les variables indépendantes *vari*. Le résultat, s'il est obtenu, sera donné à l'aide des variables dépendantes.

Exemple 32. GCL (GNU Common Lisp) Version(2.4.0) Wed May 9 12:02:00 CDT 2001
Licensed under GNU Library General Public License

```
Contains Enhancements by W. Schelter
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)
```

(C1) edp : X^2*'DIFF(Y,X)+3*X*Y=sin(X)/X;

(D1) $X^2\left(\frac{d}{dX}Y\right)+3\,X\,Y=\frac{\sin X}{X}$

(C2) ODE2(edp,Y,X);

(D2) $Y=\frac{\%C-\cos X}{X^3}$

(C3)

Les fonctions **IC1(S1,v1,v2)** et **IC2()** permettent de fixer des conditions initiales. La fonction **BC2()** permet de fixer des conditions aux limites.

Exemple 33. GCL (GNU Common Lisp) Version(2.4.0) Wed May 9 12:02:00 CDT 2001

```
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)
```

(C1) X^2*'DIFF(Y,X)+3*X*Y=sin(X)/X;

(D1) $X^2\left(\frac{d}{dX}Y\right)+3\,X\,Y=\frac{\sin X}{X}$

(C2) ODE2(%,Y,X);

(D2) $Y=\frac{\%C-\cos X}{X^3}$

(C3) IC1(D2,X=%Pi,Y=0);

(D3) $Y=-\frac{\cos X+1}{X^3}$

(C4) eqd : 'DIFF(Y,X,2)+Y*'DIFF(Y,X)^3=0;

(D4) $\frac{d^2}{dX^2}Y+Y\,\%\mathrm{DERIVATIVE}^3((Y,X,1))=0$

(C5) ODE2(eqd,Y,X);

(D5) $\frac{Y^3+6\,\%\mathrm{K1}\,Y}{6}=X+\%\mathrm{K2}$

(C6) RATSIMP(IC2(D5,X=0,Y=0,'DIFF(Y,X)=2));

(D6) $-\frac{2\,Y^3-3\,Y}{6}=X$

(C7) BC2(D5,X=0,Y=1,X=1,Y=3);

(D7) $\frac{Y^3-10\,Y}{6}=X-\frac{3}{2}$

(C8)

5.2 Calcul d'intégrales

5.2.1 Calcul de primitives

La fonction `INTEGRATE(exp,var)` permet de calculer la primitive de l'expression `exp` en fonction de la variable `var`. Si **maxima** ne sait pas répondre, il donne la notation "intégrale" de la fonction :

Exemple 34.
```
GCL (GNU Common Lisp)  Version(2.4.0) Wed May  9 12:02:00 CDT 2001
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)
```

(C1) `INTEGRATE(COS(X)*SIN(X)^4,X);`

(D1) $\frac{\sin^5 X}{5}$

(C2) `INTEGRATE(1/(COS(X)+LOG(X)),X);`

(D2) $\int \frac{1}{\log X + \cos X}\, dX$

(C3)

5.2.2 Calcul d'intégrales définies

La fonction `INTEGRATE(exp,var,var0,var1)` permet de calculer l'intégrale de l'expression `exp` où la variable `var` parcourt l'intervalle [var0, var1]. La fonction `DEFINT(exp,var,var0,var1)` donne le même résultat en utilisant des méthodes exclusivement symboliques.

Exemple 35.
```
GCL (GNU Common Lisp)  Version(2.4.0) Wed May  9 12:02:00 CDT 2001
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)
```

(C1) `INTEGRATE(LOG(X),X,1,4);`

(D1) $4\log 4 - 3$

(C2) `DEFINT(LOG(X),X,1,4);`

(D2) $4\log 4 - 3$

(C3) `INTEGRATE(COS(X)/LOG(X),X,2,4);`

(D3) $\int_2^4 \frac{\cos X}{\log X}\, dX$

(C4)

La fonction `ROMBERG(exp,var,var0,var1)` permet de calculer une valeur numérique de l'intégrale de l'expression `exp` où la variable `var` parcourt l'intervalle [var0, var1].

Exemple 36.
```
GCL (GNU Common Lisp)  Version(2.4.0) Wed May  9 12:02:00 CDT 2001
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
Maxima 5.6 Wed May 9 12:01:49 CDT 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)
```

```
(C1) ROMBERG(COS(X)/LOG(X),X,2,4);
```

(D1) -1.560726568870818

(C2)

5.3 Calcul d'intégrales indéfinies.

On peut définir des bornes infinies par `INF` pour $+\infty$ et `MINF` pour $-\infty$.

Exemple 37.

```
GCL (GNU Common Lisp)  Version(2.4.0) Fri Dec  7 15:30:59 CST 2001
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
Maxima 5.6 Fri Dec 7 15:30:47 CST 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)
```

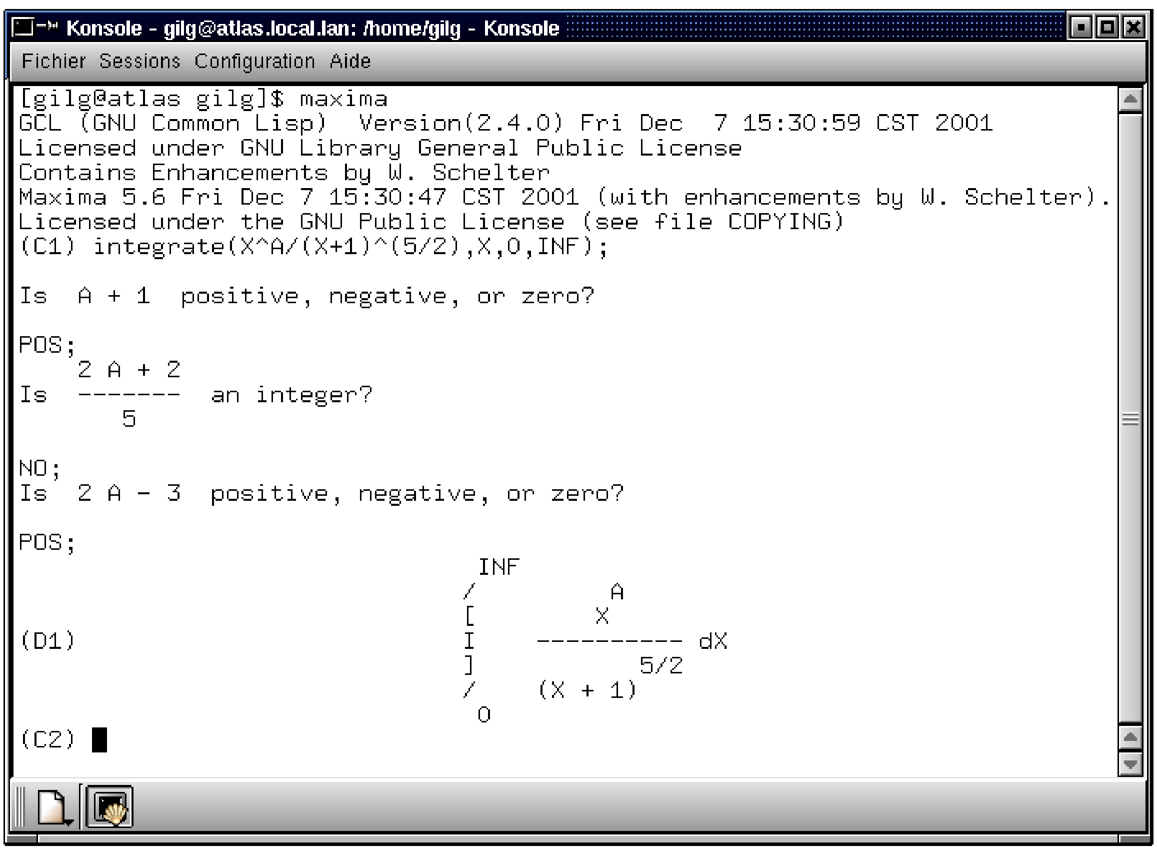
```
(C1) INTEGRATE(EXP(-X^2),X,MINF,INF);
```

(D1) $\sqrt{\pi}$

5.4 Calcul d'intégrales à paramètres.

Il est possible de calculer des intégrales à paramètres. Il se peut alors que **Maxima** demande le signe de certaines expressions. Les réponses possibles sont `POS;`, `NEG;` ou `ZERO;`.

Exemple 38.



6 Graphismes

6.1 Traçages de courbes

La fonction `PLOT2D([exp1,exp2,...],range,opts);` permet de tracer des courbes.

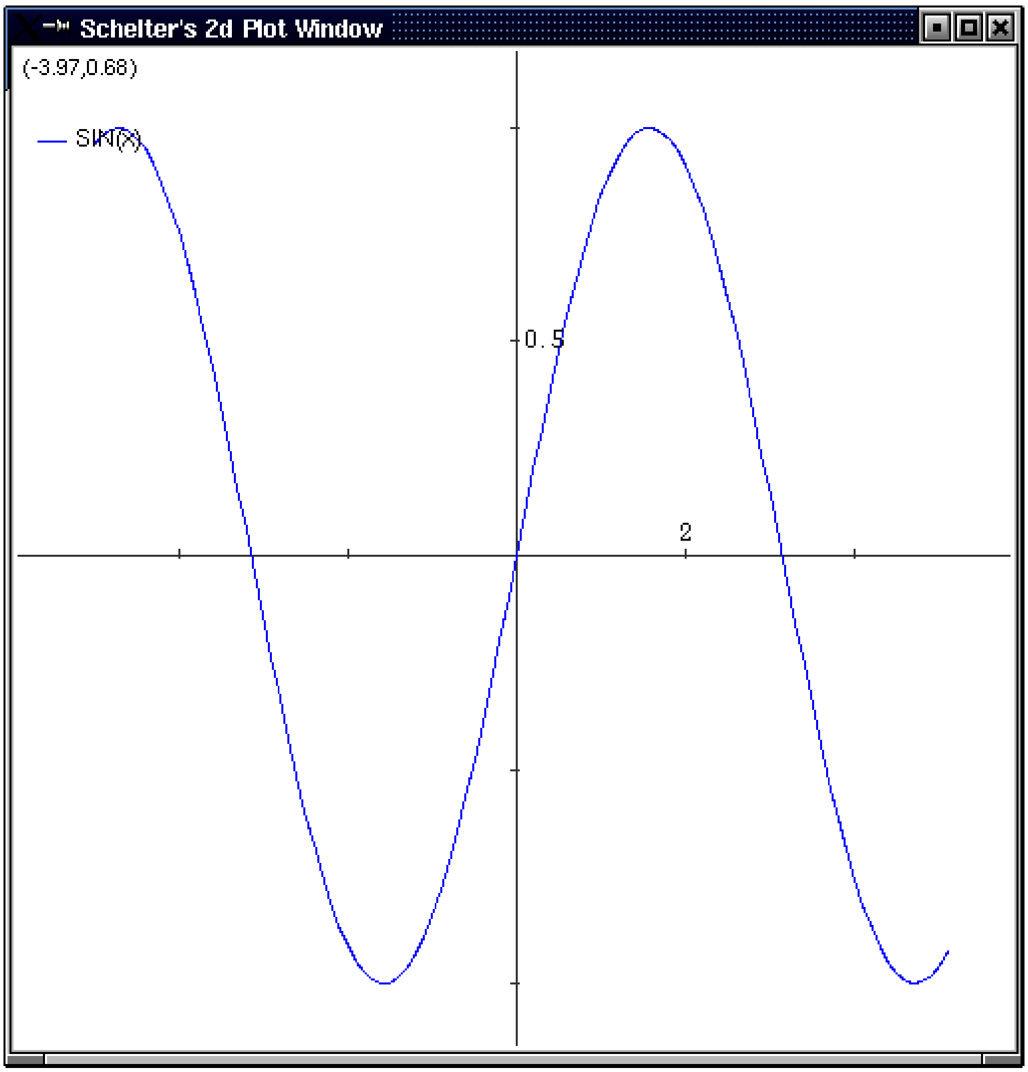
Exemple 39. GCL (GNU Common Lisp) Version(2.4.0) Fri Dec 7 15:30:59 CST 2001
```
Licensed under GNU Library General Public License
Contains Enhancements by W. Schelter
Maxima 5.6 Fri Dec 7 15:30:47 CST 2001 (with enhancements by W. Schelter).
Licensed under the GNU Public License (see file COPYING)
```

(C1) PLOT2D(SIN(x),[x,-5,5]);

(D1) 0

(C2)

Voici le résultat :

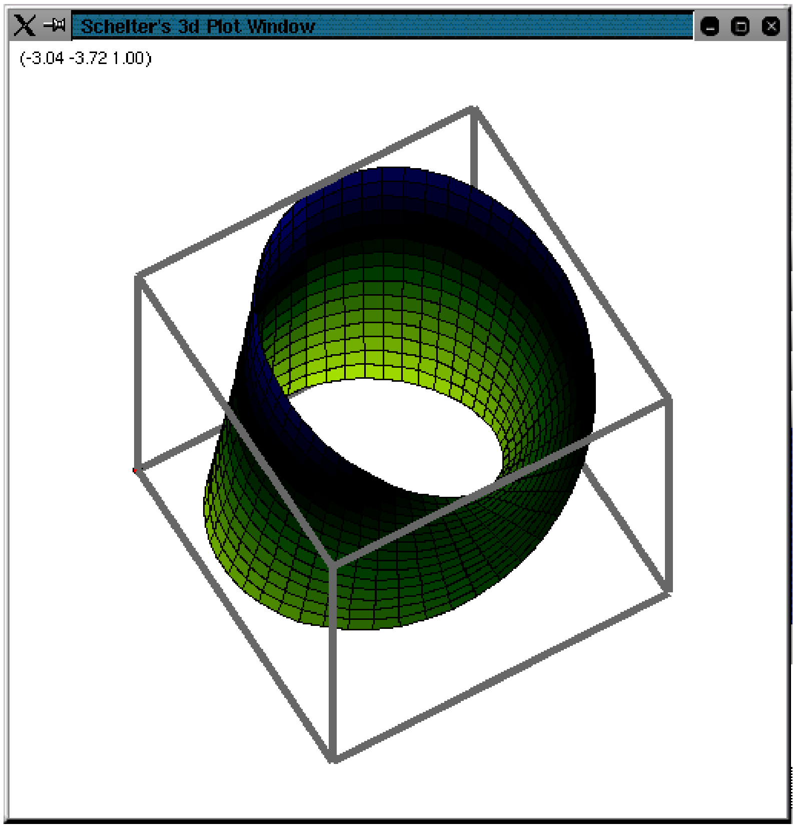


6.2 Traçages de surfaces

La commande `PLOT3d(expr,xrange,yrange,opts)` permet de tracer des surfaces.

Exemple 40.

La commande
plot3d([cos(x)*(3+y*cos(x/2)),sin(x)*(3+y*cos(x/2)),y*sin(x/2)],[x,-%Pi,%Pi],[y,-1,1],['grid,50,15]);
permet de tracer un anneau de mobeus :



6.3 Options de traçages

Le dernier argument des fonctions PLOT2D et PLOT3D permet de donner des options pour le traçage des courbes ou des surfaces. En voici les principales options disponibles :

- [PLOT_FORMAT,<*format*>] ou <*format*> définit le format de sortie comme OPENMATH, GNUPLOT, PS et GEOMVIEW.
- [RUN_VIEWVER,TRUE] permet le lancement d'un visualisateur (par exemple *gv* pour l'option [PLOT_FORMAT,PS]), si l'option est à FALSE, un fichier de données est produit.
- [GRID,30,30] définie un maillage pour PLOT3D de 30 points pour l'axe X et 30 points pour l'axe Y.
- [COLOUR_Z,false] permet d'obtenir une sortie noir et blanc pour le format PS.

La fonction SET_PLOT_OPTION(options) permet de définir les options par défaut.