FAQ MAXIMA

Version 0.93 du 13 juillet 2010

Auteur Michel Gosse
Mel : michel.gosse@free.fr

Cette Faq a pour objectif de répondre de manière concrète et pratique à toutes les questions que l'on peut se poser sur Maxima et son utilisation. La dernière version de cette faq est téléchargeable à l'adresse :

http://michel.gosse.free.fr/

Certaines réponses sont extraites de la liste internationale de Maxima dont nous remercions tous les participants.

Table des matières

1 Maxima

1.1 Le statut de Maxima

1.1.1 Maxima est-il gratuit ?

Totalement. Le code de Maxima est en open source. Vous avez droit de le copier, de le modifier, de le distribuer librement. Vous pouvez intaller le logiciel sur autant de postes que vous le désirez.

1.1.2 Maxima est-il performant ?

Tout à fait. Le code de Maxima existe depuis plus de 15 ans. Maxima est utilisé avec succès pour résoudre de nombreux problèmes mathématiques.

1.1.3 Quelles différences avec des logiciels commerciaux comme Maple ou Mathematica ?

L'interface de Maxima n'est pas très performante. Les logiciels commerciaux proposent une interface beaucoup plus ergonomique. Cependant, au niveau mathématique, Maxima n'a rien à envier aux logiciels commerciaux. Signalons que l'usage de Maxima avec TeXmacs ou WxMaxima permet d'obtenir une interface moderne et wysiwyg.

1.1.4 Qui est l'auteur de Maxima ?

Les informaticiens du département de l'Energie américain ont élaboré le premier code source. Ce code fut d'abord vendu à une société privée qui l'a développé afin de commercialiser le logiciel Macsygma. Le docteur William Schelter a ensuite obtenu l'autorisation d'exploiter le code du logiciel original, qui fut mis en open source sous le nom de Maxima. Le docteur Schelter l'a maintenu et amélioré durant une quinzaine d'année. A sa mort prématurée en 2001, le développement du logiciel a été repris par une équipe de développeurs bénévoles.

1.1.5 Quelle est la différence entre Macsygma et Maxima

Macsygma est un logiciel commercial dont le code source reposait sur le noyau de Maxima. Ce logiciel n'est plus commercialisé. La version en open source s'appelle Maxima.

1.1.6 Quel est le site de Maxima

Le site officiel est désormais hébergé par Sourceforge à l'adresse :

http://maxima.sourceforge.net/

1.2 Les versions de Maxima

1.2.1 Quelle est la dernière version de Maxima ?

La dernière version stable est la 5.9.17 en date de janvier 2009.

1.2.2 Les différentes versions sont-elles compatibles ?

ATTENTION : depuis la version 5.9.2, Maxima différentie les majuscules et les minuscules, à la fois dans le nom des fonctions et dans les variables. Cela implique que certains programmes anciens ou scripts sont à adapter.

1.2.3 Quelles sont les plate-formes supportées par Maxima

Maxima existe en version Linux et Windows. Il existe aussi une version pour Windows CE. Pour Mac, il n'existe pas de version native, mais on peut utiliser les sources de Maxima.

1.2.4 Y a-t-il de nouvelles versions de prévu ?

Oui, la version 6.0 est en construction. Elle doit permettre de nettoyer le code du logiciel et de supprimer des bugs rapportés par les utilisateurs. Cette version comportera des améliorations majeures, notamment au niveau de l'interface.

1.2.5 Qui développe Maxima ?

Une équipe de développeurs bénévoles travaille sur le code de Maxima. Ils communiquent entre eux grâce à Internet. Un site CVS dédié permet de faire les mises à jour du code. Les utilisateurs communiquent les bugs trouvés qui sont éradiqués.

1.3 Les interfaces graphiques à Maxima

Maxima est le moteur de calcul, programmé en Lisp. On peut se connecter au logiciel par le biais de différents logiciels, qui fournissent ou non une interface graphique. Dans ce cas, Maxima est le serveur, et l'interface est le client. Parmi les interfaces existantes, les suivantes sont les plus performantes :

1.3.1 L'interface xmaxima (tcl)

C'est l'interface graphique qui est livrée avec le programme. Ancienne et un peu sommaire, son remplacement par Wxmaxima est dorénavant effectuésg2e19.

1.3.2 WxMaxima

Interface moderne et fonctionnelle, qui permet d'entrer une grande partie des commandes de Maxima à l'aide d'icônes. Ses fonctionnalités d'édition et sa simplicité en font un logiciel idéal pour utiliser Maxima. Elle fonctionne sous toutes les plate-formes. Le site de référence est :

http://wxmaxima.sourceforge.net/

1.3.3 Texmacs

Texmacs est un traitement de textes scientifiques, qui permet d'entrer directement des commandes Maxima. Son intérêt est de produire des documents d'une qualité remarquable. Par contre, toutes les commandes Maxima doivent être entrées au clavier. Le site de référence :

http://www.texmacs.org/

1.3.4 Imaxima

Imaxima est une extension pour le traitement de textes Emacs qui permet d'envoyer et de recevoir des commandes Maxima dans un buffer. Le site de référence :

http://members3.jcom.home.ne.jp/imaxima/Site/Welcome.html

2 Calcul numérique

2.1 Comment trouver une valeur approchée d'un nombre ?

La commande ev(x,numer) renvoie une valeur approchée de x.

La commande bfloat(x) renvoie le nombre x en virgule flottante, à la précision donnée par fpprec que l'on peut modifier :

```
(C7) x:%pi;
(D7) π

(C8) ev(x,numer);
(D8) 3.141592653589793

(C9) fpprec:10;
(D19) 10

(C20) bfloat(x);
3.141592654B0

(C21) fpprec:100;
(D21) 100

(C22) bfloat(x);
3.14159265358979323846264338327950288419716939937510582097494459230781
 64062862089986280348253421170 68B0
```

2.2 Comment résoudre numériquement une équation ?

Pour déterminer des valeurs approchées d'une ou des racines d'une équation, on commence par localiser grossièrement cette racine, par exemple avec un graphique. Par exemple, on s'aperçoit que l'équation $\cos(x) - x = 0$ a une solution appartenant à l'intervalle $[0, 1]$. Dans ce cas, on dispose des commandes Maxima suivantes :

```
(C1) interpolate(cos(x)-x=0,x,0,1);
```

(D1) 0.73908513321516

```
(C2) load(newton);
```

(D2) /usr/lib/maxima-5.6/share/newton.mc

```
(C3) NEWTON(cos(x)-x,x,1,1/1000);
```

(D4) 0.73911289091136

```
(C5) NEWTON(cos(x)-x,x,0,1/1000);
```

(D5) 0.73911289091136

```
(C6) NEWTON(cos(x)-x,x,0,1/10000000);
```

(D6) 0.73908513338528

Dans un premier temps, il vaut mieux utiliser la commande interpolate. Lorsqu'elle échoue, on peut utiliser la commande NEWTON. Il faut dans ce cas charger le package Newton. On fera attention au fait que Maxima est alors sensible à la casse, et qu'il faut bien respecter les majuscules dans cette commande. La syntaxe de la commande NEWTON est :

NEWTON(expression,variable,point de départ, précision)

2.3 Comment trouver les solutions positives d'une équation

Solution proposée par Monsieur Rodriguez sur la liste maxima : Il définit une fonction positive, qui renvoie les éléments d'une liste dont le membre de droite est positif. Cette fonction agit uniquement sur la liste renvoyée par la commande de résolution d'une équation.

```
(%i21) positive(res):=sublist(res,
       lambda([z],member(sign(rhs(z[1])),['pos,'pz])))
```

(%o21) $\text{positive(res)} := \text{sublist}\Big(\text{res}, \text{lambda}\Big([z], \text{member}\Big(\text{sign}(\text{rhs}(z_1)), [\,'\text{pos}, '\text{pz}\,]\Big)\Big)\Big)$

```
(%i22) algsys([x^2+2*x-3],[x])
```

(%o22) $[[x = -3], [x = 1]]$

```
(%i23) positive(%)
```

(%o23) $[[x = 1]]$

2.4 La fonction partie entière

La partie entière d'un réel x est donnée par ENTIER(x) :

```
(C1) entier(2);
```

```
(D1) 2

(C2) entier(0.489);
(D2) 0

(C3) entier(-1.45);
(D3) -2
```

2.5 La fonction valeur absolue

La valeur absolue d'un réel x est donnée par ABS(x) :

```
(C1) abs(2);
(D1) 2

(C2) abs(-1);
(D2) 1

(C3) abs(x^2);
(D3) x^2

(C4) abs(x^2-x);
(D4) |x^2-x|

(C5) assume(x>1);
(D5) [x>1]

(C6) abs(x^2-x);
(D6) x^2-x
```

Comme on le voit, on peut imposer une condition sur x pour permettre à Maxima de déterminer la valeur absolue d'une expression.

2.6 Majorer une somme par la somme des valeurs absolues

Soit $X = a + b - c$. Pour majorer X par $|a| + |b| + |c|$, on écrit :

```
(C1) X:a+b+-c;
(D1) -C+b+a

(C2) apply(op(%),map(abs,args(%)));
(D2) |C|+|b|+|a|
```

2.7 Simplifier des expressions comportant des racines carrées

2.7.1 En utilisant le package sqdnst

Le package sqdnst permet certaines simplifications :

```
(C1) load("sqdnst");
```

(D1) /usr/share/maxima/5.9.0/share/simplification/sqdnst.mac

```
(C2) a:sqrt(2)+sqrt(6-4*sqrt(2));
```

(D2) $\sqrt{2}+\sqrt{6-4\sqrt{2}}$

```
(C3) sqrtdenest(d2);
```

(D3) 2

2.7.2 avec la commande ratsimp

Par défaut, la commande ratsimp de Maxima ne simplifie pas les expressions contenant des racines carrées. Mettre l'option algebraic à true permet d'imposer les simplifications :

```
(C1) q1:1/(sqrt(5)-1);
```

(D1) $\dfrac{1}{\sqrt{5}-1}$

```
(C2) ratsimp(q1);
```

(D2) $\dfrac{1}{\sqrt{5}-1}$

```
(C3) algebraic:true;
```

(D3) **true**

```
(C4) ratsimp(q1);
```

(D4) $\dfrac{\sqrt{5}+1}{4}$

2.8 Comment trouver le maximum d'une liste

Il suffit d'appliquer la commande max à une liste grâce à la fonction apply :

```
(C1) apply(max,[5,7,2,3]);
```

(D1) 7

2.9 Comment calculer des lignes trigonométriques

Par défaut, Maxima connait la valeur de quelques lignes trigonométriques. Le package ntrig permet d'obtenir des résultats supplémentaires :

```
(C1) cos(%pi/4);
```

(D1) $\dfrac{\sqrt{2}}{2}$

```
(C2) cos(%pi/5);
```

(D2) $\cos\left(\dfrac{\pi}{5}\right)$

```
(C3) load(ntrig);
```

(D3) /usr/share/maxima/5.9.0/share/trigonometry/ntrig.mac

```
(C5) cos(%pi/5);
```

(D5) $\dfrac{\sqrt{5}+1}{4}$

3 Arithmétique

3.1 Quotient et diviseur

La commande divide(m,n) renvoie le diviseur et le reste de la division de l'entier m par n. Cette commande fonctionne également avec les polynômes.

```
(C1) divide(100,7);
```

(D1) $[14, 2]$

```
(C2) divide(3693,3);
```

(D2) $[1231, 0]$

```
(C3) divide(X^3-1,X-1);
```

(D3) $\left[X^2+X+1, 0\right]$

```
(C4) divide(X^4+X^2-3,X^2+1);
```

(D4) $\left[X^2, -3\right]$

3.2 Comment obtenir le reste d'une division

La commande mod(a,b) renvoie le reste de la division de a par b :

```
(%i1) mod(25,4);
```

(%o1) 1

```
(%i2) mod(2345,23);
```

(%o3) 22

3.3 Comment changer de base

La base de départ est définie par le paramètre ibase, celle d'arrivée par le paramètre obase. Il suffit d'affecter les valeurs désirées à ces deux paramètres pour que Maxima effectue les changements de base attendus :

```
(C1) ibase:10;
```

(D1) 10

```
(C2) obase:6;
```

(D2) 10

```
(C3) [5,6,7,8];
```

(D4) $[5, 10, 11, 12]$

```
(C5) obase:12;
```

(D5) 10

```
(C6) [10,11,12,13,2576];
```

(D10) $[A, B, 10, 11, 15A8]$

Il semble qu'un bug empêche de convertir un nombre écrit en base $b > 10$ en un nombre écrit en base 10.

4 Calcul algébrique

4.1 Comment résoudre une inéquation ?

La version 5.9 de Maxima (et antérieures) ne sait pas résoudre les inéquations. Il est à souhaiter que cette fonctionnalité soit implémentée dans les futures versions, notamment la version 6.0.

4.2 Comment décomposer une fraction en éléments simples ?

On utilise la commande partfrac(expression, variable) :

```
(C1) partfrac((x^2+8*x+4)/(x^2-4),x);
```

(D2) $\frac{2}{x+2}+\frac{6}{x-2}+1$

4.3 Comment simplifier une somme de logarithmes

On emploie la fonction logcontract(expression) :

```
(C1) logcontract(log(x)+log(x+1));
```

(D1) $\log\left(x^2+x\right)$

```
(C2) logcontract(log(x)-log(x+1));
```

(D2) $\log\left(\frac{x}{x+1}\right)$

```
(C3) logcontract(10*log(x));
```

(D4) $\log x^{10}$

4.4 Comment ordonner les termes d'un polynôme

On utilise la commande declare(x,mainvar), qui déclare x comme variable principale. Dans ce cas, le polynôme s'ordonnera selon les puissances de x. Par exemple :

```
(C1) expand((x+y)^3);
```

(D1) $y^3+3\,x\,y^2+3\,x^2\,y+x^3$

```
(C2) declare(x,mainvar);
```

(D2) DONE

```
(C3) expand((x+y)^3);
```

(D3) $x^3+3\,y\,x^2+3\,y^2\,x+y^3$

Autre solution proposée par Richard Fateman, l'utilisation de la commande ratexpand(polynôme,variable) qui développe en respectant la variable x :

```
(C1) rat(expand(x+y)^3,x);
```

(D1) $x^3+3\,y\,x^2+3\,y^2\,x+y^3$

```
(C2) rat(expand(x+y+z)^3);
```

(D2) $z^3+(3\,y+3\,x)\,z^2+\left(3\,y^2+6\,x\,y+3\,x^2\right)z+y^3+3\,x\,y^2+3\,x^2\,y+x^3$

```
(C3) rat(expand(x+y+z)^3,x);
```

(D3) $x^3+(3\,z+3\,y)\,x^2+\left(3\,z^2+6\,y\,z+3\,y^2\right)x+z^3+3\,y\,z^2+3\,y^2\,z+y^3$

Le package expandwrt peut aussi contribuer à ordonner les termes d'un polynôme :

4.5 Comment obtenir le coefficient du terme d'un polynome

La commande ratcoeff(polynome,x^n) donne le coefficient du terme en x^n :

```
(C6) ratcoeff((z+1)^2*(t+y)^2,t^2);
```

(D11) $z^2+2\,z+1$

```
(C12) ratcoeff((z+1)^2*(t+y)^2,t);
```

(D12) $2\,y\,z^2+4\,y\,z+2\,y$

4.6 Comment générer un polynome à partir des coefficients

```
(C1) coefficients:[5,4,3,2];
```

(D2) $[5,4,3,2]$

```
(C3)  sum(coefficients[i+1]*x^i,i,0,length(coefficients)-1);
```

(D4) $2\,x^3+3\,x^2+4\,x+5$

4.7 Linéariser un polynome trigonométrique

La fonction trigrat(expression) permet de transformer de linéariser :

```
(%i1) trigrat(sin(x)^2);
```

(%o1) $-\dfrac{\cos\left(2\,x\right)-1}{2}$

```
(%i2) trigrat(cos(x)^4+sin(x)^3);
```

(%o2) $\dfrac{\cos\left(4\,x\right)-2\sin\left(3\,x\right)+4\cos\left(2\,x\right)+6\sin\left(x\right)+3}{8}$

4.8 Transformer avec la forme exponentielle

Pour exprimer une fonction trigonométrique ou trigométrique réciproque à l'aide de la fonction inverse, il faut utiliser la commande exponentialize :

```
(C1) exponentialize(cos(x));
```

(D1) $\dfrac{e^{ix}+e^{-ix}}{2}$

```
(C2) exponentialize(tanh(x));
```

(D2) $\dfrac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$

4.9 Comment obtenir un développement en série

La commande taylor(fonction,variable,point,ordre) donne la partie principale du développement en série de Taylor d'une fonction :

```
(C1) taylor(cos(x),x,0,5);
```

(D1) $1-\frac{x^2}{2}+\frac{x^4}{24}+\cdots$

```
(C2) taylor(log(x),x,1,4);
```

(D2) $x-1-\frac{(x-1)^2}{2}+\frac{(x-1)^3}{3}-\frac{(x-1)^4}{4}+\cdots$

4.10 Comment calculer une somme double ordonnée

On veut par exemple calculer $\sum_{1\leqslant i<j\leqslant n} a_i\,b_j$:

```
(%i1) n:5;s:0;
```

(%o1) 5
(%o2) 0

```
(%i5) for i from 1 thru n do for j from i+1 thru n do s:s+a[i]*b[j];
```

(%o5) **done**

```
(%i6) s;
```

(%o6) $a_4\,b_5+a_3\,b_5+a_2\,b_5+a_1\,b_5+a_3\,b_4+a_2\,b_4+a_1\,b_4+a_2\,b_3+a_1\,b_3+a_1\,b_2$

4.11 Comment éliminer une racine carrée d'un dénominateur

On peut utiliser la commande ratsimp(expression), à condition de positionner l'option algebraic à true :

```
(C1) x:(1+sqrt(5))/(2-sqrt(5));
```

(D1) $\frac{\sqrt{5}+1}{2-\sqrt{5}}$

```
(C2) algebraic:true;
```

(D2) **true**

```
(C3) ratsimp(x);
```

(D3) $-3\sqrt{5}-7$

4.12 Substituer à une variable une valeur

La fonction subst(valeur,variable,expression) remplace dans expression la variable par valeur.

```
(C1) eq:2*x^2+1/x-(x+1)^3/(1-x)
```

(D1) $-\frac{(x+1)^3}{1-x}+2\,x^2+\frac{1}{x}$

```
(C2) subst(3,x,eq);
```

KostasSum(exp, [i,i1,i2], [j,j1,j2], ...)

(D2) $\frac{151}{3}$

4.13 Substituer complètement à une variable une expression

Lorsque l'on désire remplacer une variable par une autre (en éliminant la première variable), il faut utiliser à la place de subst la commande ratsubst :

```
(C3) eq;
```

(D3) $-\frac{(x+1)^3}{1-x}+2\,x^2+\frac{1}{x}$

```
(C4) subst(y,1-x,eq);
```

(D5) $-\frac{(x+1)^3}{y}+2\,x^2+\frac{1}{x}$

```
(C6) ratsubst(y,1-x,eq)
```

(D6) $\frac{3\,y^4-13\,y^3+24\,y^2-23\,y+8}{y^2-y}$

4.14 Substituer complètement et simplifier une expression

Si a est tel que $a^2=1+a$, on veut simplifier l'expression $X=(a+1)^5$. Pour cela, la commande fullratsubst(expression), du package lrats, est à utiliser :

```
(C4) Y:(a+1)^5;
```

(D13) $(a+1)^5$

```
(C14) load(lrats);
```

(D14) /usr/share/maxima/5.9.0/share/simplification/lrats.mac

```
(C15) fullratsubst(1+a,a^2,Y);
```

(D15) $55\,a+34$

4.15 Manipuler une équation

Les opérations élémentaires (somme, multiplication) s'appliquent à chacun des membres d'une équation donnée. Ce comportement ne fonctionne pas si l'on utilise une fonction :

```
(C1) eq:2*x-5=5*x+3;
```

(D1) $2\,x-5=5\,x+3$

```
(C2) eq+5;
```

(D2) $2\,x=5\,x+8$

```
(C3) eq/2;
```

(D3) $\frac{2\,x-5}{2}=\frac{5\,x+3}{2}$

```
(C4) eq^2;
```

(D4) $(2\,x-5)^2=(5\,x+3)^2$

```
(C5) sin(eq);
```

(D5) $\sin\left(2\,x-5=5\,x+3\right)$

4.16 Factoriser une expression

La commande factor est faite pour cela :

```
(%i8) factor((x-1)*(2*x+3)-(x^2-2*x+1));
```

(%o8) $(x-1)\,(x+4)$

Pour des factorisations plus complexes, il existe la commande scanmap(commande,expression), qui permet d'appliquer *récursivement* la commande indiquée à l'expression donnée en argument :

```
(%i11) factor(sin(x^2-1));
```

(%o11) $\sin\left(x^2-1\right)$

```
(%i12) scanmap(factor,sin(x^2-1));
```

(%o12) $\sin\left((x-1)\,(x+1)\right)$

4.17 Isoler une solution d'un ensemble de solutions

Si par exemple sol est l'ensemble des solutions, sol[n] permet d'accéder à la nième valeur de la liste :

```
(%i1) sol:linsolve([C1*3-2,C2*1+4],[C1,C2]);
```

(%o1) $\left[\mathrm{C1}=\frac{2}{3},\mathrm{C2}=-4\right]$

```
(%i2) sol[1];
```

(%o2) $\mathrm{C1}=\frac{2}{3}$

```
(%i2) subst(sol[1],4*C1^2)
```

(%o3) $\frac{16}{9}$

On utilise dans ce cas la valeur de C1 pour calculer $4C1^2$.

5 Nombres complexes

5.1 Définition d'un nombre complexe

Le complexe $z=a+i\,b$ se définit avec Maxima par z:a+%i*b;

```
(C1) z1:sqrt(3)+%i;
```

(%o9) $\mathrm{i}+\sqrt{3}$

```
(%i10) z2:sqrt(2)-%i*sqrt(2);
```

(%o10) $\sqrt{2}-\sqrt{2}\,\mathrm{i}$

5.2 Module d'un nombre complexe

La fonction cabs(nombre complexe) renvoie le module :

```
(C5) cabs(z1);
```

(D5) 2

```
(C6) cabs(z2);
```

(D6) 2

5.3 Argument d'un nombre complexe

La fonction carg(nombre complexe) renvoie un argument :

```
(C7) carg(z1);
```

(D7) $\frac{\pi}{6}$

```
(C8) carg(z2);
```

(D8) $-\frac{\pi}{4}$

5.4 Forme algébrique d'un nombre complexe

La commande rectform(z) renvoie la forme algébrique de z :

```
(C1) z1:(1+%I)/(3-2*%I);
```

(D1) $\frac{i+1}{3-2\,i}$

```
(C2) rectform(z1);
```

(D2) $\frac{5\,i}{13}+\frac{1}{13}$

```
(C3) z2:2*exp(1+%I);
```

(D3) $2\,e^{i+1}$

```
(C4) rectform(z2);
```

(D4) $2\,e\,i\sin 1+2\,e\cos 1$

5.5 Forme exponentielle

La commande polarform(z) renvoie la forme exponentielle de z :

```
(C5) z3:1-%i;
```

(D5) $1-i$

```
(C6) polarform(z3);
```

(D6) $\sqrt{2}\,e^{-\frac{i\,\pi}{4}}$

5.6 Formules d'Euler

La fonction exponentialize[nombre complexe] permet l'application des formules d'Euler sur un nombre complexe :

```
(C2) exponentialize(cos(x));
```

(D3) $\frac{e^{ix}+e^{-ix}}{2}$

```
(C4) exponentialize(sin(2*x)+sin(x));
```

$$(D4) \quad -\frac{i\left(e^{2ix}-e^{-2ix}\right)}{2}-\frac{i\left(e^{ix}-e^{-ix}\right)}{2}$$

```
(C5) exponentialize(cos(x)^2);
```

$$(D5) \quad \frac{\left(e^{ix}+e^{-ix}\right)^2}{4}$$

```
(C6) expand(%);
```

$$(D6) \quad \frac{e^{2ix}}{4}+\frac{e^{-2ix}}{4}+\frac{1}{2}$$

5.7 Transformer une exponentielle complexe

Pour transformer une exponentielle complexe en nombre complexe, on utilise la fonction Demoivre :

```
(C7) demoivre(%e^(%i*x)+%e^(-%i*x));
```

$$(D9) \quad 2\cos x$$

6 Analyse

6.1 Comment calculer une limite ?

La fonction limit(expression,variable,point,direction) répond à cette question. La direction (facultative) est donnée par plus pour la limite par valeur supérieure, et minus pour la limite par valeur inférieure :

```
(%i1) 'limit(sin(x)/x,x,0)=limit(sin(x)/x,x,0)
```

$$(\%o7) \quad \lim_{x\to 0}\frac{\sin(x)}{x}=1$$

```
(%i8) 'limit(exp(x)/x^2,x,inf)=limit(exp(x)/x^2,x,inf)
```

$$(\%o10) \quad \lim_{x\to\infty}\frac{e^x}{x^2}=\infty$$

```
(%i11) 'limit(1/(x^2+1),x,minf)=limit(1/(x^2+1),x,minf)
```

$$(\%o11) \quad \lim_{x\to-\infty}\frac{1}{x^2+1}=0$$

```
(%i12) 'limit(1/t,t,0,plus)=limit(1/t,t,0,plus)
```

$$(\%o12) \quad \lim_{t\to 0}\frac{1}{t}=\infty$$

```
(%i13) 'limit(1/t,t,0,minus)=limit(1/t,t,0,minus)
```

$$(\%o13) \quad \lim_{t\to 0}\frac{1}{t}=-\infty$$

La fonction tlimit effectue aussi le calcul d'une limite, mais en utilisant les développements en série de Taylor. Cela peut quelquefois donner une solution que la fonction limit est incapable de trouver :

```
(%i6) limit(sqrt(9*x^2+6*x+3)+3*x+1,x,minf)
```
Is x positive or negative?`negative`

(%o14) $-\infty$

Cette réponse est fausse. Par contre, on obtient la bonne réponse avec tlimit :
```
(%i15) tlimit(sqrt(9*x^2+6*x+3)+3*x+1,x,minf)
```
(%o15) 0

6.2 Comment calculer une dérivée d'ordre n ?

La commande diff(f(x),x,n) renvoie l'expression de la dérivée n-ième de la fonction f par rapport à la variable x.

```
(C1) f(x):=x^5+1/x;
```
(D1) $f(x):=x^5+\frac{1}{x}$

```
(C2) diff(f(x),x,2);
```
(D2) $20\,x^3+\frac{2}{x^3}$

```
(C3) diff(f(x),x,6);
```
(D3) $\frac{720}{x^7}$

6.3 Calcul d'intégrales

6.3.1 Comment intégrer par parties ?

Le code suivant permet d'effectuer une intégration par parties. u est la fonction à intégrer, et v est la fonction à différentier :

```
(C1) intpart(u,v,a,b):=subst(x=b,integrate(u,x))*subst(x=b,v)-
     subst(x=a,integrate(u,x))*subst(x=a,v)-
     integrate(integrate(u,x)*diff(v,x),x,a,b);
```
(D1) intpart$(u,v,a,b):=$ SUBSTITUTE$(x=b,$INTEGRATE$(u,x))$ SUBSTITUTE$(x=b,v)-$ SUBSTITUTE$(x=a,$INTEGRATE$(u,x))$ SUBSTITUTE$(x=a,v)-$ INTEGRATE(INTEGRATE(u,x) DIFF$(v,x),x,a,b)$

```
(C2) 'integrate(x*log(x),x,1,%e)=intpart(x,log(x),1,%e);
```
(D3) $\int_1^e x\log x\,dx=\frac{e^2}{2}-\frac{\frac{e^2}{2}-\frac{1}{2}}{2}$

```
(C4) 'integrate(x*exp(2*x),x,0,1)=intpart(x,exp(2*x),0,1);
```
(D5) $\int_0^1 x\,e^{2x}\,dx=\frac{e^2}{4}+\frac{1}{4}$

```
(C6) 'integrate(x^n*sin(x),x,0,%pi/2)=intpart(x^n,sin(x),0,%pi/2);
```
Is $n+1$ zero or nonzero? `nonzero;`

(D10) $\int_0^{\frac{\pi}{2}} x^n\sin x\,dx=\frac{2^{-n-1}\,\pi^{n+1}}{n+1}-\frac{\int_0^{\frac{\pi}{2}} x^{n+1}\cos x\,dx}{n+1}$

6.3.2 Un package pour l'intégration par parties

Le package bypart permet d'effectuer une intégration par parties sans bornes définies. On dispose alors de la fonction byparts(intégrande,variable,u,dv) :

```
(C1) load(bypart);
```

(D1) /usr/share/maxima/5.9.0/share/integration/bypart.mac

```
(C2) byparts(log(x),x,log(x),1);
```

(D2) $x \log x - x$

```
(C3) byparts(x*exp(x),x,x,exp(x));
```

(D6) $x\, e^x - e^x$

6.4 Intégrer numériquement avec la méthode de Romberg

La commande romberg(fonction, variable, borne inférieure, borne supérieure) est à utiliser dans ce cas :

```
(C14) f(x):=sqrt(1+sin(1+x^2));
```

(D14) $f(x) := \sqrt{1+\sin\left(1+x^2\right)}$

```
(C15) integrate(f(x),x,0,1);
```

(D15) $\displaystyle\int_0^1 \sqrt{\sin\left(x^2+1\right)+1}\, d\, x$

```
(C16) romberg(f(x),x,0,1);
```

(D16) 1.388657330137038

6.5 La fonction dilogarithme

Il s'agit de la fonction définie par $x \longmapsto \int_1^x \frac{\ln(t)}{1-t}\mathrm{dt}$ pour $0 < x < 1$. Elle est définie en standard dans Maxima et se note li[2] :

```
(C10) li[2](1/2);
```

(D10) $\dfrac{\pi^2}{12} - \dfrac{\log^2 2}{2}$

```
(C11) expand(integrate(log(t)/(1-t),t,1,1/2));
```

(D13) $\dfrac{\pi^2}{12} - \dfrac{\log^2 2}{2}$

```
(C14) diff(li[2](x),x);
```

(D14) $-\dfrac{\log\left(1-x\right)}{x}$

7 Statistiques

7.1 Comment étudier des séries statistiques

Le package descriptive.mac permet l'étude des séries statistiques. Il se télécharge à l'adresse
http://www.telefonica.net/web2/biomates/maxima/descriptive/descriptive.htm

8 Algèbre linéaire

8.1 Vecteurs

8.1.1 Comment noter un vecteur

On définit un vecteur par ses coordonnées que l'on note entre deux crochets. Par exemple, la commande u:[a,b,c]; définit le vecteur $\vec{u}$ de coordonnées (a, b, c).

```
(C1) u:[1,2,3];
```

(D1) $[1, 2, 3]$

```
(C2) v:[-1,2,3];
```

(D2) $[-1, 2, 3]$

```
(C3) u+v;
```

(D3) $[0, 4, 6]$

8.1.2 Comment calculer un produit scalaire

Le produit scalaire se note avec un point, précédé et suivi d'un espace :

```
(C13) a:[1,2,3];
```

(D13) $[1, 2, 3]$

```
(C14) b:[0,-1,5];
```

(D14) $[0, -1, 5]$

```
(C15) a . b;
```

(D16) 13

8.1.3 Comment calculer un produit vectoriel

Il faut charger le package vect par la commande load("vect"); Ensuite, le produit vectoriel se note ~, et le calcul effectif du produit vectoriel s'effectue par la commande express(vecteur);

```
maxima] load("vect");
```

(D10) /usr/share/maxima/5.9.0/share/vector/vect.mac

```
(C11) [x,y,z]~[x1,y1,z1];
```

(D11) $([x, y, z], [\mathrm{x1}, \mathrm{Y1}, \mathrm{z1}])$

```
(C12) express(%);
```

(D12) $[y\,\mathrm{z1} - \mathrm{Y1}\,z, \mathrm{x1}\,z - x\,\mathrm{z1}, x\,\mathrm{Y1} - \mathrm{x1}\,y]$

8.2 Matrices

8.2.1 Comment écrire une matrice

On définit une matrice A par la commande A:matrix([1,2,3],[-1,5,2],[4,3,0]);

```
(C8) A:matrix([1,2,3],[-1,5,2],[4,3,0]);
```

$$(D9)\quad \begin{pmatrix} 1 & 2 & 3 \\ -1 & 5 & 2 \\ 4 & 3 & 0 \end{pmatrix}$$

8.2.2 Comment extraire une ligne d'une matrice

On entre la matrice et entre crochets le numéro de la ligne désirée.

```
(C11) A[1];
```

$$(D11)\quad [1,2,3]$$

8.2.3 Comment extraire une colonne d'une matrice

La fonction col(matrice,numéro de colonne) renvoie la colonne désirée :

```
(C12) col(A,1);
```

$$(D12)\quad \begin{pmatrix} 1 \\ -1 \\ 4 \end{pmatrix}$$

8.2.4 Comment transposer une matrice

La fonction transpose(matrice) effectue cette opération.

```
(C13) A;
```

$$(D13)\quad \begin{pmatrix} 1 & 2 & 3 \\ -1 & 5 & 2 \\ 4 & 3 & 0 \end{pmatrix}$$

```
(C14) transpose(A);
```

$$(D14)\quad \begin{pmatrix} 1 & -1 & 4 \\ 2 & 5 & 3 \\ 3 & 2 & 0 \end{pmatrix}$$

8.2.5 La fonction echelon

Elle renvoie une matrice triangulaire supérieure obtenue avec des combinaisons élémentaires de lignes et de colonnes, avec des éléments diagonaux égaux à 1.

```
(C26) A;
```

$$(D26)\quad \begin{pmatrix} 1 & 2 & 3 \\ -1 & 5 & 2 \\ 4 & 3 & 0 \end{pmatrix}$$

```
(C27) echelon(A);
```

$$(D27)\quad \begin{pmatrix} 1 & \frac{3}{4} & 0 \\ 0 & 1 & \frac{8}{23} \\ 0 & 0 & 1 \end{pmatrix}$$

8.2.6 Transformer une matrice en vecteur

Soit la matrice $A=\begin{pmatrix} 3 & 2 \\ 4 & 5 \\ 6 & 1 \end{pmatrix}$. On souhaite la transformer en un vecteur ligne de 6 éléments :

```
(C16) A:matrix([3,2],[4,5],[6,1]);
```

(D16) $\begin{pmatrix} 3 & 2 \\ 4 & 5 \\ 6 & 1 \end{pmatrix}$

```
(C17) flatten_matrix(m):= apply(append,args(transpose(m)))$
(C18) flatten_matrix(A);
```

(D18) $[3, 4, 6, 2, 5, 1]$

8.2.7 Comment générer automatiquement une matrice

On définit une suite double, par exemple f[i,i], qui va générer les coefficients. On termine avec la fonction genmatrix(suite utilisée,nombre de lignes, nombre de colonnes) :

```
(C2) f[i,j]:=i^2+j^2;
```

(D7) $f_{i,j} := i^2 + j^2$

```
(C8) genmatrix(f,4,5);
```

(D9) $\begin{pmatrix} 2 & 3 & 4 & 5 & 26 \\ 3 & 4 & 5 & 6 & 29 \\ 4 & 5 & 6 & 7 & 34 \\ 5 & 6 & 7 & 8 & 41 \end{pmatrix}$

On peut utiliser cette possibilité pour écrire une matrice sous forme générale :

```
(C10) genmatrix(a,3,2);
```

(D1) $\begin{pmatrix} a_{1,1} & a_{1,2} \\ a_{2,1} & a_{2,2} \\ a_{3,1} & a_{3,2} \end{pmatrix}$

8.2.8 Comment multiplier deux matrices

La multiplication des matrices se note avec le point (.), et non pas avec l'étoile (*) :

```
(C1) M:matrix([a,b],[c,d]);
```

(D1) $\begin{pmatrix} a & b \\ C & d \end{pmatrix}$

```
(C2) M.M;
```

(D2) $\begin{pmatrix} b\,C + a^2 & b\,d + a\,b \\ C\,d + a\,C & d^2 + b\,C \end{pmatrix}$

8.2.9 Comment inverser une matrice

La matrice inverse de A se note A^^(-1) :

```
(C1) M:matrix([a,b],[c,d]);
```

(D1) $\begin{pmatrix} a & b \\ C & d \end{pmatrix}$

```
(C2) M^^(-1);
```

(D3) $\begin{pmatrix} \frac{d}{a\,d - b\,C} & -\frac{b}{a\,d - b\,C} \\ -\frac{C}{a\,d - b\,C} & \frac{a}{a\,d - b\,C} \end{pmatrix}$

8.2.10 Comment calculer une puissance de matrices

La matrice A^n se note A^^n :

```
(C4) M:matrix([u,v],[w,t]);
```

(D4) $$\begin{pmatrix} u & v \\ w & t \end{pmatrix}$$

```
(C5) M^^2;
```

(D5) $$\begin{pmatrix} v\,w+u^2 & u\,v+t\,v \\ u\,w+t\,w & v\,w+t^2 \end{pmatrix}$$

```
(C6) M^^3;
```

(D6) $$\begin{pmatrix} u\left(v\,w+u^2\right)+\left(u\,v+t\,v\right)w & v\left(v\,w+u^2\right)+t\left(u\,v+t\,v\right) \\ w\left(v\,w+t^2\right)+u\left(u\,w+t\,w\right) & t\left(v\,w+t^2\right)+v\left(u\,w+t\,w\right) \end{pmatrix}$$

```
(C7) expand(%);
```

(D7) $$\begin{pmatrix} 2\,u\,v\,w+t\,v\,w+u^3 & v^2\,w+u^2\,v+t\,u\,v+t^2\,v \\ v\,w^2+u^2\,w+t\,u\,w+t^2\,w & u\,v\,w+2\,t\,v\,w+t^3 \end{pmatrix}$$

8.3 Comment résoudre une récurrence linéaire

Le package solve_rec permet dans certains cas d'exprimer, dans le cas d'une suite récurrente linéaire, le terme d'ordre n en fonction de n. Pour cela, on dispose de la fonction solve_rec(définition suite récurrence,suite,valeur d'un terme) :

```
(%i16) load(solve_rec)
```

(%o16) /usr/share/maxima/5.9.3/share/contrib/solve$_{\text{r}}$ec/solve$_{\text{r}}$ec.mac

```
(%i17) solve_rec(u(n)=n*u(n-1)/(1+n),u(n))
```

(%o17) $$u(n)=\frac{\%\mathrm{k}_1}{n+1}$$

```
(%i18) solve_rec(u(n)=n*u(n-1)/(1+n),u(n),u(1)=3)
```

(%o18) $$u(n)=\frac{6}{n+1}$$

```
(%i19) solve_rec(u(n)=u(n-1)+u(n-2),u(n))
```

(%o19) $$u(n)=\frac{\left(\sqrt{5}-1\right)^n\%\mathrm{k}_1\,(-1)^n}{2^n}+\frac{\left(\sqrt{5}+1\right)^n\%\mathrm{k}_2}{2^n}$$

```
(%i20) solve_rec(u(n)=u(n-1)+u(n-2),u(n),u(0)=1,u(2)=5)
```

(%o20) $$u(n)=\frac{\left(\sqrt{5}+1\right)^n\left(7\sqrt{5}+5\right)}{10\,2^n}-\frac{\left(\sqrt{5}-1\right)^n\left(7\sqrt{5}-5\right)(-1)^n}{10\,2^n}$$

9 Graphisme

9.1 Courbes planes cartésiennes

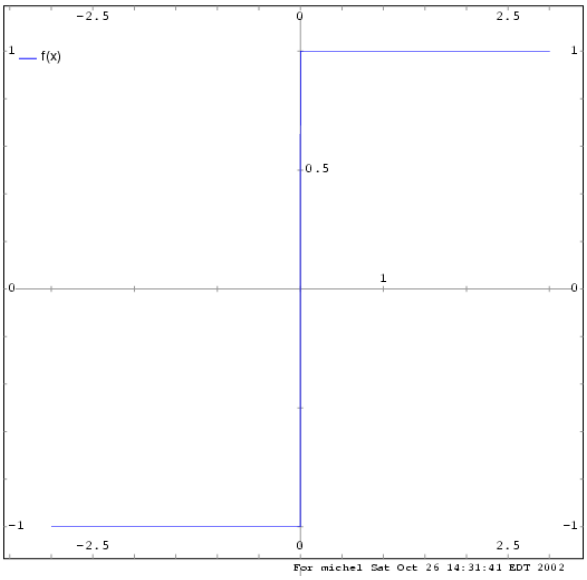
9.1.1 Représenter une fonction en escalier

Maxima renvoie un message d'erreur si l'on essaye de représenter une fonction en escalier. Pour y arriver, il faut utiliser une évaluation différée. Par exemple, représentons la fonction f définie sur $\mathbb{R}$ par $f(x)=-1$ si $x<0$ et $f(x)=1$ si $x\geqslant 0$:

```
maxima] f(t):=if t>=0 then 1.0 else -1.0;
```

(D4) $f(t) := \textbf{if } t \geq 0 \textbf{ then } 1.0 \textbf{ else } -1.0$

```
(C5) plot2d('(f(x)),[x,-3,3]);
```



9.1.2 Représenter la fonction partie entière

```
(C1) plot2d('(float(entier(x))),[x,0,10]);
```

9.1.3 Représenter une famille de fonctions

Pour représenter la famille de fonctions $x \to \cos(n\,x)$, pour n variant de 1 à 3, on entre la commande :

```
(C1) plot2d(makelist(cos(n*x),n,1,3),[x,0,2*%pi]);
```

9.2 Comment représenter un cercle

Pour tracer le cercle d'équation $x^2 + y^2 = 1$, on utilise la représentation paramétrique $x = \cos(t)$ et $y = \sin(t)$:

```
(C3) plot2d([parametric,cos(t),sin(t),[t,-%pi*2,%pi*2],[nticks,80]]);
```

9.3 Courbes planes polaires

9.3.1 Comment tracer une courbe en polaire

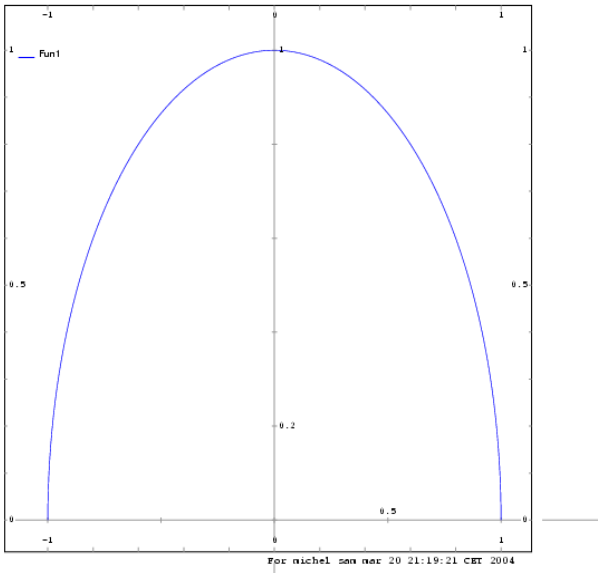
Cette solution est proposée par Stavros Macrakis : on entre le programme suivant :

```
(C1) plot_polar(expr,range) := block([theta_var: range[1]], plot2d( ['parametric,
     cos(theta_var)*expr, sin(theta_var)*expr, range]))$
(C2) plot_polar(1,[t,0,%pi])$
(C3) plot_polar( 1 + 2 * cos(t), [t,0,2*%pi])$
```

```
(C4)
```

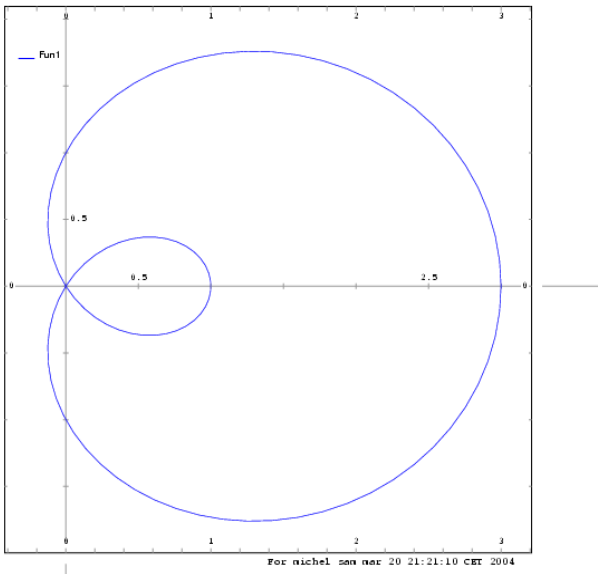
Le tracé de la courbe d'équation $\rho = 1$ sur $[0, \pi]$ s'obtient par :

```
(C2) plot_polar(1,[t,0,%pi])$
```



Le tracé d'une cardioide s'obtient par :

```
(C2) plot_polar( 1 + 2 * cos(t), [t,0,2*%pi])$
```



9.4 Courbes planes paramétriques

9.4.1 Comment tracer une courbe en paramétrique ?

Le tracé d'une courbe définie par un système d'équations paramétriques n'a pas été prévu dans le code de Maxima. Cependant, le docteur Schelter a implémenté dans les dernières versions de Maxima une fonction remplissant ce rôle. Le code suivant ne fonctionne donc qu'avec la version 5.9. Dans ce cas, voici des exemples :

(C1) plot2d([parametric,cos(t),sin(t),[t,-%pi*2,%pi*2]]);
(C2) plot2d([parametric,cos(t),sin(t),[t,-%pi*2,%pi*2], [nticks,8]]);
(C3) plot2d([x^3+2,[parametric,cos(t),sin(t),[t,-5,5]]], [x,-3,3]);
Function: PLOT2D (expr,range,...,options,..)
Function: PLOT2D ([expr1,expr2,..,exprn],xrange,...,options,..)
Function: PLOT2D (parametric_expr)
Function: PLOT2D ([..,expr,..,parametric_expr,..],xrange,...,options)
donnent les différentes possibilités de tracer des courbes en paramétrique.

10 Entrées et sorties

10.1 Variables réservées

10.1.1 Variables réservées à Maxima

Les noms de variable C1, D1, C2, D2, ... sont réservées à Maxima. Elles désignent les entrées (Ci) et les sorties (Di). Par exemple (merci à Stavros Macrakis) :

```
(C1) x^3-1;
```

(D1) x^3-1

```
(C2) solve(d1,x);
```

(D2) $\left[x=\frac{\sqrt{3}\,i-1}{2},x=-\frac{\sqrt{3}\,i+1}{2},x=1\right]$

```
(C3) part(d2,1,2);
```

(D3) $\frac{\sqrt{3}\,i-1}{2}$

```
(C4) pickapart(d3,2);
```

(E4) $\sqrt{3}\,i$

(D4) $\frac{\mathrm{E4}-1}{2}$

```
(C5) c2;
```

(D5) SOLVE(D1, x)

```
(C6) d3;
```

(D6) $\frac{\sqrt{3}\,i-1}{2}$

```
(C7) e4;
```

(D7) $\sqrt{3}\,i$

Attention : à compter de la version 5.9.2, la numérotation des entrées/sorties a été modifiée. L'entrée d'une commande est notée %in, où n est un entier qui s'incrémente d'une unité après chaque commande, tandis que les sorties se numérotent %on. Les variables c et d sont donc libres dorénavant.

10.1.2 Comment noter plus l'infini ?

Maxima utilise la variable réservée inf :

```
(C1) limit((x+1)/(x+5),x,inf)
```

(D1) 1

```
(C2) integrate(exp(-x^2),x,0,inf)
```

(D2) $\frac{\sqrt{\pi}}{2}$

10.1.3 Comment noter moins l'infini ?

Maxima utilise la variable réservée minf :

```
(C3) limit(exp(x),x,minf)
```

(D3) 0

10.2 Effacer la valeur d'une variable

La commande kill(variable) efface la définition précédemment entrée :

```
(C22) x:3+sqrt(5);
```

(D22) $\sqrt{5}+3$

```
(C23) x;
```

(D23) $\sqrt{5}+3$

```
(C24) kill(x);
```

(D24) DONE

```
(C25) x;
```

(D25) x

10.3 Fonctions usuelles

10.3.1 Comment écrire la fonction logarithme népérien

Elle se note log sous Maxima :

```
maxima] log(1);
```

(D5) 0

```
(C6) log(exp(x));
```

(D6) x

10.3.2 Comment définir la fonction logarithme décimal

```
(C3) log10(x):=log(x)/log(10);
```

(D3) $\log10(x):=\frac{\log x}{\log 10}$

```
(C4) log10(10);
```

(D4) 1

```
(C9) diff(log10(x),x);
```

(D9) $\dfrac{1}{\log 10\, x}$

10.3.3 Comment définir une fonction dérivée

On veut définir la fonction dérivée d'une fonction f, pour pouvoir la réutiliser :

```
(C1) expression:x^2+1/x;
```

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

```
(C2) define(f(x),expression);
```

(D2) $f(x):=x^2+\dfrac{1}{x}$

```
(C3) define(df(x),diff(f(x),'x));
```

(D3) $\mathrm{df}(x):=2\,x-\dfrac{1}{x^2}$

```
(C4) df(1);
```

(D4) 1

10.4 Comment transformer une expression en fonction

Si on a définit une expression expr, on peut définir une fonction à l'aide de la commande f(x):=''(expr); On utilise deux fois la simple quote. Une autre syntaxe possible est define('f(x),expr); Par exemple :

```
maxima] expr:logcontract(integrate(2/(x^2-1),x));
```

(D7) $\log\left(\dfrac{x-1}{x+1}\right)$

```
(C8) f(x):=''(expr);
```

(D8) $f(x):=\log\left(\dfrac{x-1}{x+1}\right)$

```
(C9) f(5);
```

(D9) $\log\left(\dfrac{2}{3}\right)$

```
(C10) expand(factor(diff(f(x),x)));
```

(D12) $\dfrac{2}{x^2-1}$

```
(C13) define('g(x),D12);
```

(D13) $g(x):=\dfrac{2}{x^2-1}$

```
(C14) g(5);
```

(D14) $\dfrac{1}{12}$

10.5 Définir une fonction formellement

Il est possible de définir une fonction f en précisant uniquement la variable dont elle dépend. Cela permet d'effectuer notamment du calcul différentiel dans le cas général. La commande à utiliser est dans ce cas depends(fonction, variable) :

```
(C5) depends(f,x);
```

(D5) $[f(x)]$

```
(C6) depends(g,x);
```

(D6) $[g(x)]$

```
(C7) diff(f*g,x);
```

(D7) $f\left(\frac{d}{d\,x}\,g\right)+\frac{d}{d\,x}\,f\,g$

```
(C8) diff(f/g,x);
```

(D8) $\frac{\frac{d}{d\,x}\,f}{g}-\frac{f\left(\frac{d}{d\,x}\,g\right)}{g^2}$

```
(C9) ratsimp(%);
```

(D9) $-\frac{f\left(\frac{d}{d\,x}\,g\right)-\frac{d}{d\,x}\,f\,g}{g^2}$

10.6 Comment récupérer un élément d'une liste

Les commandes first(liste), second(liste), third(liste) renvoient respectivement les premier, deuxième et troisième élément de la liste nommée liste. Par exemple :

```
maxima] first([a,b,c,d]);
```

(D15) a

```
(C16) second([a,b,c,d]);
```

(D16) b

```
(C17) third([a,b,c,d]);
```

(D17) C

10.7 Comment additionner les éléments d'une liste

```
(C1) apply("+",[1,5,7,-1]);
```

(D1) 12

10.8 Comment simplifier une expression trigonométrique

On peut utiliser la fonction trigsimp(expression) :

```
maxima] trigsimp(tan(x));
```

(D19) $\frac{\sin x}{\cos x}$

```
(C20) tan(x);
```

(D20) $\tan x$

10.9 Comment tester une égalité

La fonction is(equal(expression1,expression2) renvoie vrai ou faux :

```
(C15) is(equal(4,2+2));
```

(D15) **true**

```
(C16) is(equal((x+1)^2,x^2+2*x+1));
```

(D16) **true**

```
(C17) is(equal((x+1)^2,x^2+1));
```

(D18) **false**

Attention, cette fonction n'est pas très sûre et semble renvoyer un résultat faux sur certains tests :

```
(C21) is(equal(sqrt(x^2),x));
```

(D22) **true**

10.10 Comment déclarer qu'un nombre est entier

La commande declare(n,integer) le réalise facilement.

```
(C1) sin(n*%PI);
```

(D1) $\sin(\pi\, n)$

```
(C2) declare(n,integer);
```

(D2) DONE

```
(C3) sin(n*%PI);
```

(D3) 0

10.11 Comment déclarer qu'un nombre est positif

La commande assume(x>0) permet à Maxima de savoir que le réel x est positif. Le renseignement sera utilisé dans les calculs :

```
(C1) 'abs(x)=abs(x);
```

(D1) $\mathrm{ABS}(x) = |x|$

```
(C2) assume(x>0);
```

(D2) $[x > 0]$

```
(C3) 'abs(x)=abs(x);
```

(D3) $\mathrm{ABS}(x) = x$

10.12 Comment savoir si une expression contient une variable

La commande freeof(x,expression) renvoie true si expression contient x, et false dans le cas contraire.

```
(C1) freeof(x,2*a+5*y^2);
```

(D1) **true**

```
(C2) freeof(x,x^2-5*y+6);
```

(D2) **false**

Attention, la commande n'est pas récursive, et donc le résultat renvoyé est faux si l'expression a été définie précédemment :

```
(C3) x:5*a+2;
```

(D3) $5\,a+2$

```
(C4) freeof(x,a);
```

(D4) **true**

10.13 Isoler une sous-expression

La commande pickapart(expression, profondeur) isole les sous-expressions d'une expression donnée. Par exemple :

```
(C10) u2:(sqrt(7)-1)/(1+sqrt(3));
```

(D18) $$\frac{\sqrt{7}-1}{\sqrt{3}+1}$$

```
(C19) pickapart(u2,1);
```

(E19) $\sqrt{7}-1$

(E20) $\sqrt{3}+1$

(D20) $$\frac{\mathrm{E19}}{\mathrm{E20}}$$

```
(C20) pickapart(u2,2);
```

(E21) $\sqrt{7}$

(E22) $\sqrt{3}$

(D22) $$\frac{\mathrm{E21}-1}{\mathrm{E22}+1}$$

10.14 Comment exporter en tex

La commande tex(expression) renvoie l'expresssion codée en tex :

```
(C1) tex(sqrt(2)+1/2);
```

```
$$\sqrt{2}+{{1}\over{2}}$$
```
(D1) **false**

(C4) `'integrate(1/(x+1),x)=integrate(1/(x+1),x);`

(D5) $$\int \frac{1}{x+1}\,dx = \log\left(x+1\right)$$

(C6) `tex(%);`

```
$$\int {{{1}\over{x+1}}}{\;dx}=\log \left(x+1\right)$$
```
(D6) **false**

10.15 Comment choisir un fichier d'exportation

On dispose d'un fichier dont le nom est contenu dans une variable. Pour l'utiliser avec la commande tex, on utilise les syntaxes suivantes :

(C2) `nomfichier:"test.tex";`

(D3) test.tex

(C4) `tex(sqrt(3)+sqrt(2),''nomfichier);`

(D4) **false**

(C5) `nomfichier2:"/home/michel/temp/test.tex";`

(D6) /home/michel/temp/test.tex

(C7) `tex(%pi,''nomfichier2);`

(D7) **false**

Dans le premier cas, un fichier test.tex est créé dans le répertoire courant, tandis que dans le deuxième cas le fichier test.tex est situé dans le dossier /home/michel/temp/.

10.16 Comment charger des commandes au lancement de maxima

Il suffit de les mettre dans le fichier :

/usr/share/maxima/5.9.0/share/maxima-init.mac

qui est lu automatiquement au démarrage de maxima. On peut aussi rajouter des fonctions lisp personnalisées dans le fichier :

/usr/share/maxima/5.9.0/share/maxima-init.lisp

10.17 Comment utiliser une fonction comme argument

Voici les codes proposés par Barton Willis pour le calcul de :

$$\mathrm{xsum}(f,n) = \sum_{k=1}^{n} f(k)$$

où f est une fonction donnée et n un entier. Les tests se font avec la fonction identité puis la fonction carrée.

(C1) `xsum(f,n):=block([s:0],for k:1 thru n do (s:s + apply(f,[k])),s)$`

```
(C6) f(x):=x$
(C7) g(x):=x^2$
(C8) xsum(f,2);
```

(D8) 3

```
(C9) xsum(g,3);
```

(D9) 14

Autre solution, utilisant la commande translate :

```
(C10) ysum(f,n):=block([s:0],for k:1 thru n do (s:s+f(k)),s)$
(C11) ysum(f,2);
```

(D11) 3

```
(C12) ysum(g,3);
```

(D12) 6 (On constate ici une erreur)

```
(C13) translate(ysum)$
(C14) ysum(g,3);
```

(D14) 14

10.18 Comment composer n fois une fonction

On définit une procédure qui permet de composer n fois une fonction (astuce de Stavros Macrakis) :

```
(C4)  composen(f,n,x):=if n=0 then x else f(composen(f,n-1,x))$
(C5) composen(sin,3,y) ;
```

(D5) $\sin\sin\sin y$

```
(C6) composen(f,2,x);
```

(D6) $f(f(x))$

10.19 Comment collecter des termes d'une expression

On veut exprimer un polynôme de plusieurs variables en fonction des puissances de l'une d'entre elles (astuce de Barton Willis) :

```
(C7) p : expand((a+b+c)^3);
```

(D7) $C^3+3\,b\,C^2+3\,a\,C^2+3\,b^2\,C+6\,a\,b\,C+3\,a^2\,C+b^3+3\,a\,b^2+3\,a^2\,b+a^3$

```
(C8) collectterms(p,c);
Warning - you are redefining the MACSYMA function INTERSECTION
```

(D8) $C^3+\left(3\,b+3\,a\right)C^2+\left(3\,b^2+6\,a\,b+3\,a^2\right)C+b^3+3\,a\,b^2+3\,a^2\,b+a^3$

```
(C9) facsum(p,c);
```

(D9) $C^3+3\left(b+a\right)C^2+3\left(b+a\right)^2C+\left(b+a\right)^3$

10.20 Exporter en Latex

Par défaut, Maxima dispose d'une fonction pour générer du TeX. Pour obtenir du Latex, il suffit d'utiliser le package mactex-utilities :

```
(C10) tex(x/y);

$${{x}\over{y}}$$
(D11) false

(C12) load("mactex-utilities");

(D12) /usr/share/maxima/5.9.0/share/utils/mactex-utilities.lisp

(C13) tex(x/y);

$$\frac{x}{y}$$
(D13) false
```

10.21 Firewall et port sous Windows XP

Sous Windows XP, Maxima communique avec le moteur de calcul formel par le port 4040. Il faut donc ouvrir ce port en cas d'utilisation d'un firewall pour que Maxima puisse fonctionner.