Tutoriel Maxima pour enseigner les mathématiques en BTS

Thomas BRÉLIVET

15 juin 2014

Document en cours d'élaboration.

Pour toute question ou remarque, envoyez un email : thomas.brelivet@ac-creteil.fr

Table des matières

Chapitre 1

Première prise en main de Maxima

Maxima (http ://maxima.sourceforge.net/) est un logiciel de calcul formel, descendant du logiciel Macsyma développé dès 1968 au MIT (institut de recherche et université américaine). Il est disponible sous linux, Mac OS X et Windows. C'est un logiciel libre distribué sous licence GNU GPL, programmé dans le langage Lisp.

Ce logiciel possède une interface graphique qui est wxmaxima. C'est cette interface graphique avec maxima que l'on utilise ici.

Le logiciel wxmaxima se présente graphiquement en deux parties : la zone de menu et la zone d'exécution des commandes. La zone de menu, comme beaucoup de logiciels comporte les sous menus « Fichiers », « Edition » et « Aide ». Dans la zone d'exécution des commandes, on peut écrire du texte comme ce que vous êtes en train de lire et aussi bien sûr exécuter des commandes !

Les commandes sont toujours précédées d'une flêche lorsqu'elles sont vides ou de %i et un nombre entouré par des parenthèses lorsquelles contiennent une commande qui a été exécutée. Si la commande (ci-dessous) n'a pas été exécutée, il faut mettre le curseur dessus et appuyer simultanément les touches Majuscule et entrée (Maj+Entrée).

```
(%i1) 2+5;
```

(%o1) 7

La lettre i signifie input (entrée) et le o output (sortie). Le pourcentage sert à dire que i1 est un nom, c'est le nom de la première entrée. On peut les utiliser dans la suite des commandes. Un poucentage seul est le nom du dernier résultat obtenu.

```
(%i2) %;
```

(%o2) 7

```
(%i3) %o1;
```

(%o3) 7

```
(%i4) %i1;
```

(%o4) 7

Le point virgule « ; » marque la fin de la commande à éxécuter. À la place du point virgule, on peut mettre un dollar pour ne pas afficher le résultat. Cela peut être utile si l'on ne souhaite pas faire apparaître de résultats intermédiaires.

```
(%i5) 2+5$
```

```
(%i6) %i5+2;
```

(%o6) 9

On peut aussi écrire plusieurs commandes dans une même cellule :

```
(%i7) 2+7;
      7/8;
```

(%o7) 9

(%o8) $\frac{7}{8}$

Si on éxécute plusieurs fois une commande, la numérotation augmente et certains numéros disparaissent. On peut remettre à jour la numérotation en utilisant "Menu/Maxima/Redémarrer Maxima" puis "Menu/Cell/Réévaluer toutes les cellules". Avec le clavier on peut utiliser : Alt+M+R puis CTRL+R.

Chapitre 2

Les nombres

2.1 Opérations sur les nombres

2.1.1 Somme et différence

```
(%i1)  2+3;
       2-3;
```

(%o1) 5

(%o2) -1

2.1.2 Produit et division

```
(%i3)  5*45;
```

(%o3) 225

```
(%i4)  -63/49;
```

(%o4) $-\frac{9}{7}$

2.1.3 Arrondis

Si on veut un arrondi, on utilise la commande float :

```
(%i5)  float(%);
```

(%o5) -1.285714285714286

On peut l'avoir directement si on écrit un des nombres sous forme décimale :

```
(%i6)  -63.0/49;
       -63/49.0;
```

(%o6) -1.285714285714286

(%o7) -1.285714285714286

2.2 Des fonctions sur les entiers

2.2.1 Factorisation

```
(%i8)  factor(6);
       factor(45454566780);
```

(%o8) $2\,3$

(%o9) $2^2\,3^2\,5\,7^2\,19\,271241$

2.2.2 Racine carrée

La racine est un entier

```
(%i10) sqrt(4);
```

(%o10) 2

La racine est non entière et le résultat donne la racine carrée :

```
(%i11) sqrt(2);
```

(%o11) $\sqrt{2}$

Les racine est non entière et on obtient une valeur approchée :

```
(%i12) sqrt(2.0);
```

(%o12) 1.414213562373095

Valeur exacte écrite sous forme de puissance

```
(%i13) sqrt(8);
```

(%o13) $2^{\frac{3}{2}}$

Valeur approchée

```
(%i14) sqrt(8.0);
```

(%o14) 2.82842712474619

2.2.3 Puissances

```
(%i15) 2^3;
```

(%o15) 8

```
(%i16) 45^3;
```

(%o16) 91125

2.2.4 Valeur absolue

Valeur absolue :

```
(%i17) abs(-4);
```

(%o17) 4

2.2.5 Factorielle

La factorielle d'un nombre

```
(%i18) 3!;
```

(%o18) 6

```
(%i19) 50!;
```

(%o19) 30414093201713378043612608166064768844377641568960512000000000000

2.3 Représentation des nombres

La notation scientifique

```
(%i20) sqrt(23.0)^2-23;
```

(%o20) $-3.5527136788005009\,10^{-15}$

Si l'on souhaite avoir d'avantage de précision, on indique la précision souhaitée, par exemple avec 100 chiffres après la décimale :

```
(%i21) fpprec:100;
```

(%o21) 100

On utilise ensuite la commande bfloat :

```
(%i22) bfloat(sqrt(4700000));
```

(%o22) $2.167948338867879941898962448 0[43 digits]41853792930268502274725809 43b3$

où b3 signifie que le résultat est multiplié par 10^3 (il y a 43 chiffres qui ne sont pas affichés, ils sont remplacés par [43 digits]). On peut aussi écrire la commande en une seule ligne :

```
(%i23) sqrt(8), bfloat, fpprec:50;
```

(%o23) $2.8284271247461900976033774484193961571393437507539b0$

Chapitre 3

Les variables

3.1 Les variables

Déclaration d'une variable

```
(%i1) a:2;
```

(%o1) 2

Maintenant la variable a, contient la valeur 2 :

```
(%i2) a;
```

(%o2) 2

Si l'on souhaite désaffecter la variable a :

```
(%i3) kill(a);
```

(%o3) *done*

```
(%i4) a;
```

(%o4) a

On peut développer :

```
(%i5) expand((a+1)^2);
```

(%o5) $a^2 + 2\,a + 1$

On peut aussi factoriser :

```
(%i6) factor(a^2+5*a+6);
```

(%o6) $(a + 2)\,(a + 3)$

Les identités remarquables avec la commande expand :

```
(%i7) expand((a+b)^2);
```

(%o7) $b^2 + 2\,a\,b + a^2$

```
(%i8) expand((a-b)^2);
```

(%o8) $b^2 - 2\,a\,b + a^2$

```
(%i9)  expand((a-b)*(a+b));
```

(%o9) $a^2 - b^2$

Les identités remarquables avec la commande factor :

```
(%i10) factor(a^2+2*a*b+b^2);
```

(%o10) $(b + a)^2$

```
(%i11) factor(a^2-2*a*b+b^2);
```

(%o11) $(b - a)^2$

```
(%i12) factor(a^2-b^2);
```

(%o12) $-(b - a)\ (b + a)$

Après utilisation de declare(a,mainvar) ; afin d'avoir la variable a en premier :

```
(%i13) declare(a,mainvar);
```

(%o13) *done*

```
(%i14) expand((a+b)^2);
       expand((a-b)^2);
       expand((a-b)*(a+b));
```

(%o14) $a^2 + 2\,b\,a + b^2$

(%o15) $a^2 - 2\,b\,a + b^2$

(%o16) $a^2 - b^2$

```
(%i17) factor(a^2+2*a*b+b^2);
       factor(a^2-2*a*b+b^2);
       factor(a^2-b^2);
```

(%o17) $(a + b)^2$

(%o18) $(a - b)^2$

(%o19) $(a - b)\ (a + b)$

3.2 Les constantes

Le nombre PI

```
(%i20) %pi;
```

(%o20) π

```
(%i21) float(%pi);
```

(%o21) 3.141592653589793

Le nombre e

```
(%i22) %e;
```

(%o22) e

```
(%i23) float(%e);
```

(%o23) 2.718281828459045

3.3 Substitutions de variables

Il peut parfois être utilise de remplacer une variable par une expression ou par une valeur. La fonction subst permet de réaliser cette action :

```
(%i24) subst(x^2, a,(a+1)^2);
```

(%o24) $\left(x^2+1\right)^2$

```
(%i25) subst(2, a,(a+1)^2);
```

(%o25) 9

Chapitre 4

Les équations

4.1 Les équations

4.1.1 Définition d'une équation

On commence par définir une équation et on lui donne un nom :

```
(%i1) equation1:x^2-1=0;
```

(%o1) $x^2 - 1 = 0$

4.1.2 Résolution exacte d'une équation

On résout l'équation suivante, où l'inconnue est x (dans le menu : Menu/Equations/Résoudre...) :

```
(%i2) solve(equation1,x);
```

(%o2) $[x = -1, x = 1]$

On peut aussi l'écrire directement :

```
(%i3) solve(x^2-1=0,x);
```

(%o3) $[x = -1, x = 1]$

4.1.3 Utilisation des solutions obtenues

Les résultats précédents sont donnés sous forme d'équation $x = \ldots$.

Pour obtenir la première équation, on écrit :

```
(%i4) solve(x^2-1=0,x)[1];
```

(%o4) $x = -1$

Pour la deuxième :

```
(%i5) solve(x^2-1=0,x)[2];
```

(%o5) $x = 1$

Pour récupérer le membre de droite, on utilise rhs (right hand side) :

```
(%i6) rhs(solve(x^2-1=0,x)[1]);
      rhs(solve(x^2-1=0,x)[2]);
```

(%o6) -1

(%o7) 1

Si on souhaite récupérer la variable, on utilise lhs (left hand side) :

```
(%i8) lhs(solve(x^2-1=0,x)[1]);
      lhs(solve(x^2-1=0,x)[2]);
```

(%o8) x

(%o9) x

Pour vérifier qu'un nombre annule une expression :

```
(%i10) subst(-1, x, x^2-1);
```

(%o10) 0

4.1.4 Résolution approchée d'une équation

Si une solution exacte de l'équation n'existe pas, on obtient l'équation :

```
(%i11) solve(x^7-2*x+5=0,x);
```

(%o11) $[0 = x^7 - 2\,x + 5]$

Dans ce cas on utilise une méthode de résolution numérique, il faut donner un intervalle où la fonction change une seule fois de signe.

```
(%i12) find_root(x^7-2*x+5=0, x, -2, 0);
```

(%o12) -1.337973214895086

En changeant l'intervalle, on peut obtenir toutes les solutions :

```
(%i13) find_root(x^5+20*x^2+2*x-1=0, x, -10, 10);
       find_root(x^5+20*x^2+2*x-1=0, x, -10, -1);
       find_root(x^5+20*x^2+2*x-1=0, x, -1, 0);
```

(%o13) 0.17910867241289

(%o14) -2.673637481699788

(%o15) -0.27931420183824

4.1.5 Résolution d'équations polynomiales

Si l'équation est polynomiale, on dispose de realroots :

```
(%i16) realroots(x^5+20*x^2+2*x-1);
```

(%o16) $[x = -\frac{89712387}{33554432}, x = -\frac{9372229}{33554432}, x = \frac{6009889}{33554432}]$

Pour obtenir les solutions réelles approchées, on utilise float :

```
(%i17) float(realroots(x^5+20*x^2+2*x-1));
```

(%o17) $[x = -2.673637479543686, x = -0.27931419014931, x = 0.17910864949226]$

Par contre solve ne trouve pas ces solutions !

```
(%i18) solve(x^5+20*x^2+2*x-1=0, x);
```

(%o18) $[0 = x^5 + 20\,x^2 + 2\,x - 1]$

4.2 Les inéquations

Pour les inéquations, il existe le package solve_rat_ineq qui fonctionne pour les expressions rationnelles.

```
(%i19) load(solve_rat_ineq)$
```

```
(%i20) solve_rat_ineq((2*x+3)*(x)>0);
```

(%o20) $[[x < -\frac{3}{2}], [x > 0]]$

```
(%i21) solve_rat_ineq((x-1)^2*(x+1)^2>0);
```

(%o21) $[[x < -1], [x > -1, x < 1], [x > 1]]$

Chapitre 5

Les fonctions

5.1 Définition d'une fonction

(%i1) `f(x):=x^2;`

(%o1) $f(x) := x^2$

Avec define :

(%i2) `define(g(x), x^3-1);`

(%o2) $g(x) := x^3 - 1$

5.2 Tableau de valeurs d'une fonction

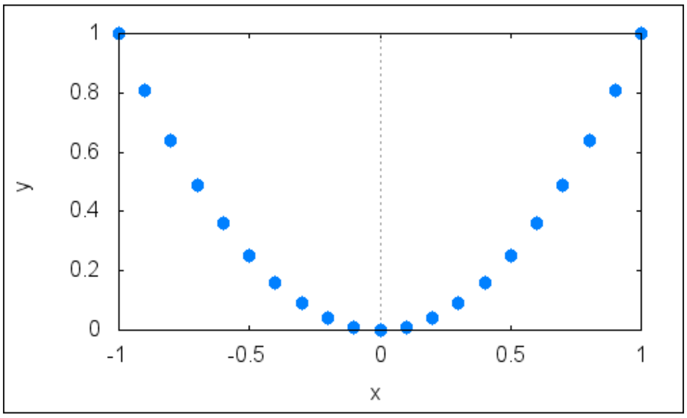
On utilise les listes :

(%i3)
```
valeursdex:makelist(-1+k/10,k,0, 20);
imagesdesvaleursdex:map(f,valeursdex);
```

(%o3) $[-1, -\frac{9}{10}, -\frac{4}{5}, -\frac{7}{10}, -\frac{3}{5}, -\frac{1}{2}, -\frac{2}{5}, -\frac{3}{10}, -\frac{1}{5}, -\frac{1}{10}, 0, \frac{1}{10}, \frac{1}{5}, \frac{3}{10}, \frac{2}{5}, \frac{1}{2}, \frac{3}{5}, \frac{7}{10}, \frac{4}{5}, \frac{9}{10}, 1]$

(%o4) $[1, \frac{81}{100}, \frac{16}{25}, \frac{49}{100}, \frac{9}{25}, \frac{1}{4}, \frac{4}{25}, \frac{9}{100}, \frac{1}{25}, \frac{1}{100}, 0, \frac{1}{100}, \frac{1}{25}, \frac{9}{100}, \frac{4}{25}, \frac{1}{4}, \frac{9}{25}, \frac{49}{100}, \frac{16}{25}, \frac{81}{100}, 1]$

(%i5) `wxplot2d([discrete, valeursdex,imagesdesvaleursdex],[style, points]);`

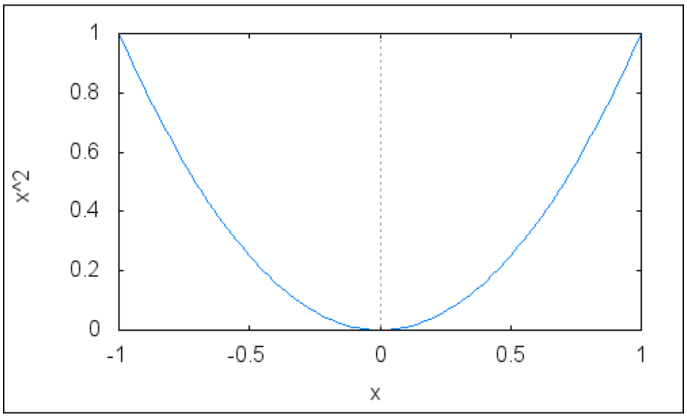
(%t5) 

(%o5)

5.3 Courbe représentative d'une fonction

On donne l'expression de f(x) et l'intervalle sur lequel on souhaite représenter la fonction. L'intervalle des coordonnées se calcule automatiquement.

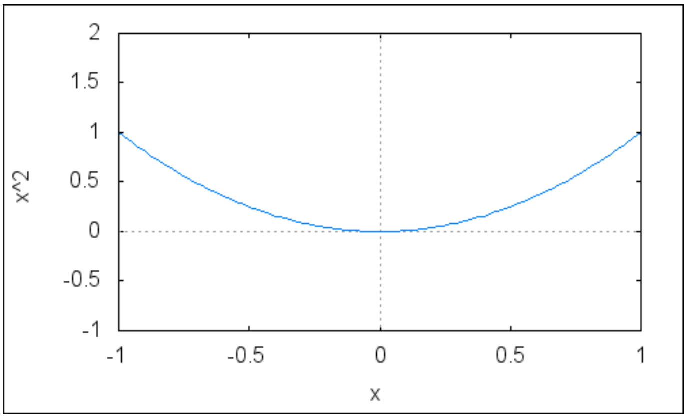
(%i6)
```
wxplot2d(f(x),[x,-1,1]);
```

(%t6) 

(%o6)

On peut aussi donner cet intervalle.

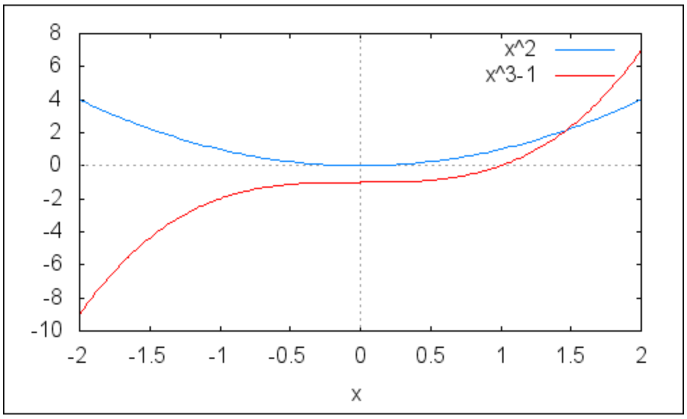
(%i7)
```
wxplot2d(f(x),[x,-1,1], [y, -1,2]);
```

(%t7) 

(%o7)

Pour tracer plusieurs courbes à la fois, on donne la liste des expressions des fonctions à représenter.

(%i8)
```
wxplot2d([f(x),g(x)],[x,-2,2]);
```



(%t8)

(%o8)

5.4 Dérivée d'une fonction

```
(%i9) define(f(x),x^20+x^15);
      diff(f(x),x);
      define(df(x),diff(f(x),x));
```

(%o9) $\mathrm{f}(x) := x^{20} + x^{15}$

(%o10) $20\,x^{19} + 15\,x^{14}$

(%o11) $\mathrm{df}(x) := 20\,x^{19} + 15\,x^{14}$

Dérivée seconde :

```
(%i12) diff(f(x),x,2);
```

(%o12) $380\,x^{18} + 210\,x^{13}$

Dérivées d'ordre supérieur :

```
(%i13) diff(f(x),x,5);
```

(%o13) $1860480\,x^{15} + 360360\,x^{10}$

Remarque : pour définir la dérivée, il faut utiliser define et non := Voici deux exemples illustrant une différence :

```
(%i14) g(x):=x^2;
       h(x):=x^3;
```

(%o14) $\mathrm{g}(x) := x^2$

(%o15) $\mathrm{h}(x) := x^3$

```
(%i16) dg(x):=diff(g(x),x);
       define(dh(x), diff(h(x),x));
```

(%o16) $\mathrm{dg}(x) := \mathrm{diff}(\mathrm{g}(x), x)$

(%o17) $\mathrm{dh}(x) := 3\,x^2$

Jusqu'ici pas de différence, en effet :

```
(%i18) dg(x);
       dh(x);
```

(%o18) $2x$

(%o19) $3x^2$

mais pour le calcul en une valeur := donne une erreur :

```
(%i20) dg(2);
       dh(2);
```

$diff : secondargumentmustbeavariable; found"2\#0" : dg(x = 2) - -anerror.Todebugthistry : debugmode(true);$

(%o21) 12

Lorsque l'on demande dg(2), Maxima exécute la commande diff(g(2), 2) ce qui n'est pas possible car le deuxième paramètre de diff doit-être une variable. On dérive par rapport à une variable.

5.5 Limites d'une fonction

En plus l'infini

```
(%i22) limit(f(x),x,inf);
```

(%o22) ∞

Lorsque le signe n'est pas précisé, c'est plus l'infini. Lorsque la limite est indéfinie et infinie en valeur absolue, on obtient le mot "infinity" :

```
(%i23) limit(1/x,x,0);
```

(%o23) $infinity$

En 0^+ et 0^- :

```
(%i24) limit(1/x, x, 0, plus);
```

(%o24) ∞

```
(%i25) limit(1/x, x, 0, minus);
```

(%o25) $-\infty$

En moins l'infini

```
(%i26) limit(f(x),x,minf);
       limit(f(x),x,-inf);
```

(%o26) ∞

(%o27) ∞

En une valeur

```
(%i28) limit(f(x),x,2);
```

(%o28) 1081344

Limite indéfinie

```
(%i29) limit(sin(x), x, inf);
```

(%o29) *ind*

5.6 Fonctions affines

```
(%i30) kill(a)$
       kill(b)$
```

```
(%i32) h(x):=a*x+b;
```

(%o32) $\mathrm{h}\left(x\right) := a\,x+b$

Zéro d'une fonction affine

```
(%i33) solve(h(x)=0,x);
```

(%o33) $[x=-\frac{b}{a}]$

Dérivée d'une fonction affine

```
(%i34) diff(h(x),x);
```

(%o34) a

Limites

```
(%i35) limit(f(x),x,-inf);
       limit(f(x),x,x[0]);
       limit(f(x),x,inf);
```

(%o35) ∞

(%o36) $x_0^{20}+x_0^{15}$

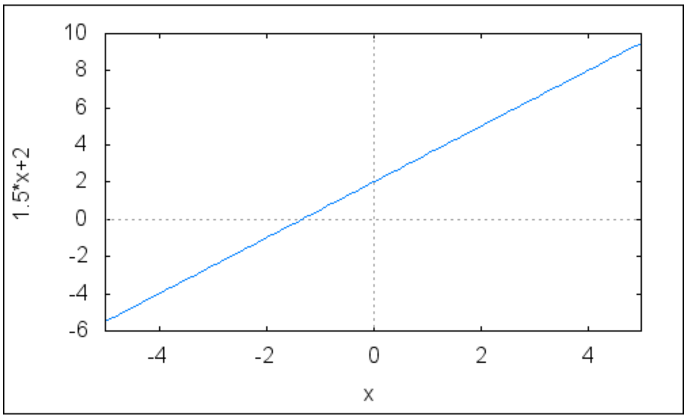
(%o37) ∞

Courbe d'une fonction affine : droite

```
(%i38) a:1.5$
       b:2$
       h(x);
```

(%o40) $1.5\,x+2$

```
(%i41) wxplot2d([h(x)], [x,-5,5])$
```

(%t41) 

5.7 Fonctions du second degré

```
(%i42) kill(a)$
       kill(b)$
       kill(c)$
```

```
(%i45) f(x):=a*x^2+b*x+c;
```

(%o45) $f(x) := a\,x^2 + b\,x + c$

Zéros d'un polynôme du second degré

```
(%i46) solutions:solve(f(x)=0,x);
```

(%o46) $[x = -\frac{\sqrt{b^2 - 4\,a\,c} + b}{2\,a}, x = \frac{\sqrt{b^2 - 4\,a\,c} - b}{2\,a}]$

Attention cette expression est valable pour toutes les valeurs réelles de Δ. Si Delta est négatif le nombre %i apparaît, voir les exemples suivants. Dans le cas où Delta = b^2-4*a*c est positif :

```
(%i47) subst(1, c,subst(4, b, subst(1, a, solutions)));
```

(%o47) $[x = -\frac{2\sqrt{3} + 4}{2}, x = \frac{2\sqrt{3} - 4}{2}]$

Dans le cas où Delta = b^2-4*a*c est nul :

```
(%i48) subst(1, c,subst(2, b, subst(1, a, solutions)));
```

(%o48) $[x = -1, x = -1]$

Dans le cas où Delta = b^2-4*a*c est négatif :

```
(%i49) subst(4, c,subst(2, b, subst(1, a, solutions)));
```

(%o49) $[x = -\frac{2\sqrt{3}\,i + 2}{2}, x = \frac{2\sqrt{3}\,i - 2}{2}]$

Dérivée d'une fonction polynôme

```
(%i50) define(df(x),diff(f(x),x));
```

(%o50) $\mathrm{df}(x) := 2\,a\,x + b$

Zéro de la dérivée

```
(%i51) solve(df(x)=0,x);
```

(%o51) $[x = -\frac{b}{2\,a}]$

```
(%i52) a:1.5$
       b:2$
       c:-1$
       f(x);
```

(%o55) $1.5\,x^2 + 2\,x - 1$

Limites

```
(%i56) limit(f(x),x,-inf);
       limit(f(x),x,x[0]);
       limit(f(x),x,inf);
```

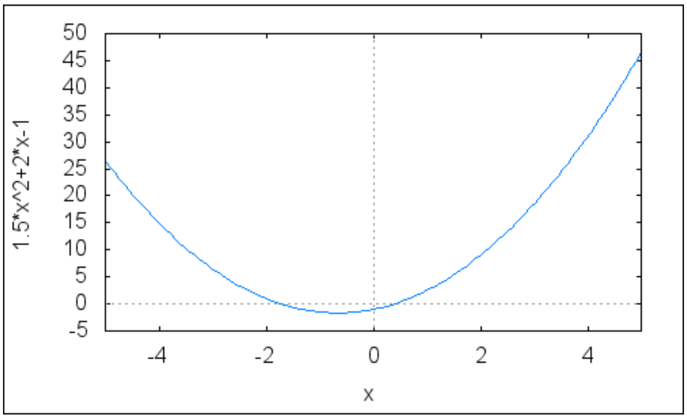
(%o56) ∞

(%o57) $1.5\,x_0^2 + 2\,x_0 - 1$

(%o58) ∞

Courbe d'une fonction du second degré

```
(%i59) wxplot2d([f(x)], [x,-5,5])$
```

(%t59) 

5.8 Fonction exponentielle

Définition

```
(%i60) define(f(x),exp(x));
```

(%o60) $\mathrm{f}(x) := e^x$

Valeurs particulières

```
(%i61) exp(0);
```

(%o61) 1

```
(%i62) exp(1);
```

(%o62) e

Dérivée

```
(%i63) define(df(x),diff(f(x),x));
```

(%o63) $\mathrm{df}(x) := e^x$

Limites

```
(%i64) limit(f(x),x,-inf);
       limit(f(x),x,x[0]);
       limit(f(x),x,inf);
```

(%o64) 0

(%o65) e^{x_0}

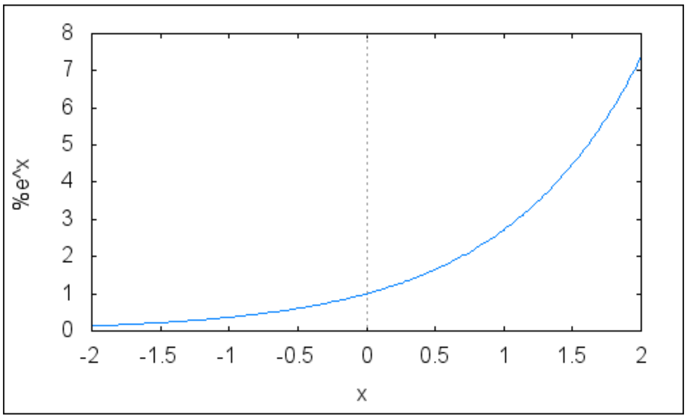
(%o66) ∞

```
(%i67) limit(exp(x)/x,x,inf);
```

(%o67) ∞

Courbe

```
(%i68) wxplot2d([f(x)], [x,-2,2])$
```

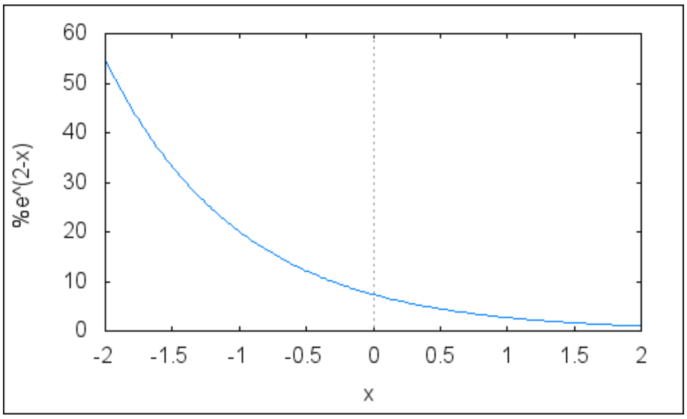
(%t68) 

Composée avec une fonction affine

```
(%i69) kill(a)$
       kill(b)$
       define(f(x),exp(a*x+b));
       define(df(x),diff(f(x),x));
       a:-1$ b:2$
       wxplot2d([f(x)], [x,-2,2])$
```

(%o71) $\mathrm{f}(x) := e^{a\,x+b}$

(%o72) $\mathrm{df}(x) := a\,e^{a\,x+b}$

(%t75) 

Relations fonctionnelles

```
(%i76) kill(a)$
       kill(b)$
```

```
(%i78) exp(a)*exp(b);
```

(%o78) e^{b+a}

```
(%i79) 1/exp(a);
```

(%o79) e^{-a}

```
(%i80) exp(a)/exp(b);
```

(%o80) e^{a-b}

```
(%i81) (exp(a))^b;
```

(%o81) $e^{a\,b}$

```
(%i82) sqrt(exp(a));
```

(%o82) $e^{\frac{a}{2}}$

5.9 Fonction logarithme népérien

Definition

```
(%i83) define(f(x),log(x));
```

(%o83) $\mathrm{f}(x) := \log(x)$

Valeurs particulières

```
(%i84) log(1);
       log(1.0);
```

(%o84) 0

(%o85) 0.0

```
(%i86) log(%e);
       log(2.718);
```

(%o86) 1

(%o87) 0.99989631572895

Dérivée

```
(%i88) define(df(x),diff(f(x),x));
```

(%o88) $\mathrm{df}\left(x\right) := \frac{1}{x}$

Limites

```
(%i89) limit(f(x),x,-inf);
       limit(f(x),x,3);
       limit(f(x),x,inf);
```

(%o89) $infinity$

(%o90) $\log\left(3\right)$

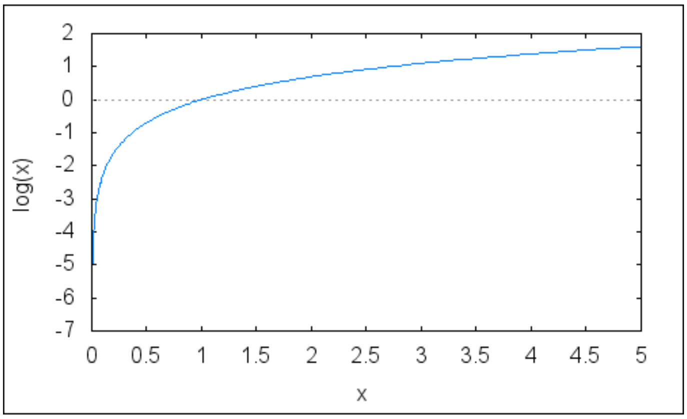
(%o91) ∞

```
(%i92) limit(log(x)/x,x,inf);
```

(%o92) 0

Courbe

```
(%i93) wxplot2d([f(x)], [x,0.001,5])$
```

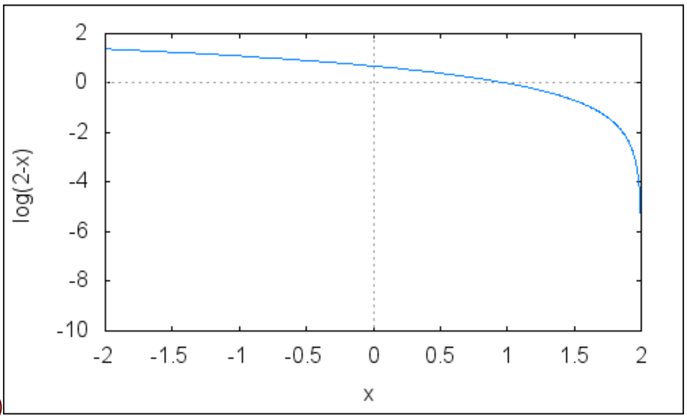
(%t93) 

Composée avec une fonction affine

```
(%i94) kill(a)$
       kill(b)$
       define(f(x),log(a*x+b));
       define(df(x),diff(f(x),x));
       a:-1$ b:2$
       wxplot2d([f(x)], [x,-2,1.9999])$
```

(%o96) $\mathrm{f}\left(x\right) := \log\left(a\,x+b\right)$

(%o97) $\mathrm{df}\left(x\right) := \frac{a}{a\,x+b}$

(%t100) 

Logarithme et exponentielle

```
(%i101)log(exp(x));
```

(%o101) x

```
(%i102)exp(log(x));
```

(%o102) x

Relations fonctionnelles

```
(%i103)kill(a)$
       kill(b)$
```

```
(%i105)logcontract(log(a)+log(b));
```

(%o105) $\log(a\,b)$

```
(%i106)logcontract(-log(a));
```

(%o106) $-\log(a)$

```
(%i107)logcontract(log(a)-log(b));
```

(%o107) $\log\left(\frac{a}{b}\right)$

```
(%i108)log(a^b);
```

(%o108) $\log(a)\ b$

```
(%i109)log(sqrt(a));
```

(%o109) $\frac{\log(a)}{2}$

5.10 Fonctions trigonométriques

Définition

```
(%i110)define(f(x),cos(x));
       define(g(x),sin(x));
```

(%o110) $f(x) := \cos(x)$

(%o111) $g(x) := \sin(x)$

Valeurs particulières entre 0 et $2*\pi$

```
(%i112)valeursdex:sort(append(makelist(k*%pi/6,k,0, 12),[%pi/4, 3*%pi/4, 5*%pi/4, 7*%pi/4]));
       valeursdecosx:map(f,valeursdex);
       valeursdesinx:map(g,valeursdex);
```

(%o112) $[0, \frac{\pi}{6}, \frac{\pi}{4}, \frac{\pi}{3}, \frac{\pi}{2}, \frac{2\pi}{3}, \frac{3\pi}{4}, \frac{5\pi}{6}, \pi, \frac{7\pi}{6}, \frac{5\pi}{4}, \frac{4\pi}{3}, \frac{3\pi}{2}, \frac{5\pi}{3}, \frac{7\pi}{4}, \frac{11\pi}{6}, 2\pi]$

(%o113) $[1, \frac{\sqrt{3}}{2}, \frac{1}{\sqrt{2}}, \frac{1}{2}, 0, -\frac{1}{2}, -\frac{1}{\sqrt{2}}, -\frac{\sqrt{3}}{2}, -1, -\frac{\sqrt{3}}{2}, -\frac{1}{\sqrt{2}}, -\frac{1}{2}, 0, \frac{1}{2}, \frac{1}{\sqrt{2}}, \frac{\sqrt{3}}{2}, 1]$

(%o114) $[0, \frac{1}{2}, \frac{1}{\sqrt{2}}, \frac{\sqrt{3}}{2}, 1, \frac{\sqrt{3}}{2}, \frac{1}{\sqrt{2}}, \frac{1}{2}, 0, -\frac{1}{2}, -\frac{1}{\sqrt{2}}, -\frac{\sqrt{3}}{2}, -1, -\frac{\sqrt{3}}{2}, -\frac{1}{\sqrt{2}}, -\frac{1}{2}, 0]$

Dérivées

```
(%i115)define(df(x),diff(f(x),x));
       define(dg(x),diff(g(x),x));
```

(%o115) $\mathrm{df}(x) := -\sin(x)$

(%o116) $\mathrm{dg}(x) := \cos(x)$

Limites

L'affichage de ind veut dire indéfinie, la limite n'existe pas !

```
(%i117)limit(f(x),x,-inf);
       limit(f(x),x,3);
       limit(f(x),x,inf);
       limit(g(x),x,-inf);
       limit(g(x),x,3);
       limit(g(x),x,inf);
```

(%o117) ind

(%o118) $\cos(3)$

(%o119) ind

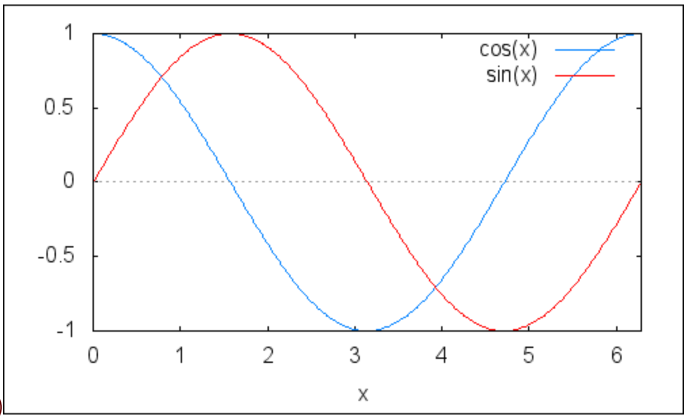
(%o120) ind

(%o121) $\sin(3)$

(%o122) ind

Courbes

```
(%i123)wxplot2d([f(x),g(x)], [x,0,2*%pi])$
```

(%t123) 

Composée avec une fonction affine

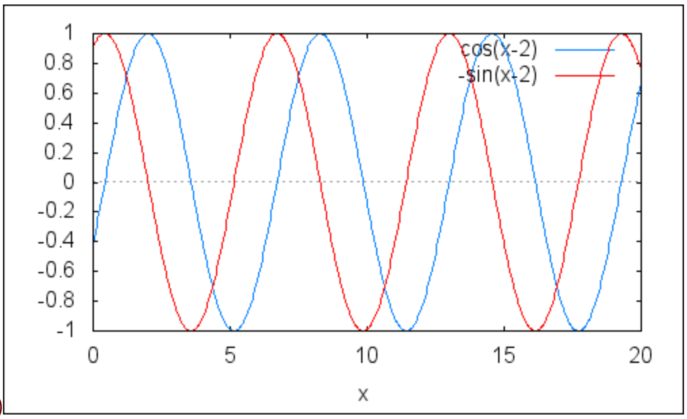
```
(%i124)kill(a)$
       kill(b)$
       define(f(x),cos(a*x+b));
       define(df(x),diff(f(x),x));
       define(g(x),sin(a*x+b));
       define(dg(x),diff(f(x),x));
       a:-1$ b:2$
       wxplot2d([f(x),g(x)], [x,0,20])$
```

(%o126) $\mathrm{f}(x) := \cos(a\,x+b)$

(%o127) $\mathrm{df}(x) := -a\sin(a\,x+b)$

(%o128) $\mathrm{g}(x) := \sin(a\,x+b)$

(%o129) $\mathrm{dg}(x) := -a\sin(a\,x+b)$

(%t132) 

Relations

```
(%i133)kill(a)$
       kill(b)$
```

```
(%i135)trigexpand(cos(a+b));
```

(%o135) $\cos(a)\,\cos(b) - \sin(a)\,\sin(b)$

```
(%i136)trigexpand(cos(a-b));
```

(%o136) $\sin(a)\,\sin(b) + \cos(a)\,\cos(b)$

```
(%i137)trigexpand(sin(a+b));
```

(%o137) $\cos(a)\,\sin(b)+\sin(a)\,\cos(b)$

```
(%i138)trigexpand(sin(a-b));
```

(%o138) $\sin(a)\,\cos(b)-\cos(a)\,\sin(b)$

5.11 Primitives et intégrales d'une fonction

5.11.1 Primitives

```
(%i139)kill(all)$
```

```
(%i1) integrate(a*x+b, x);
```

(%o1) $\frac{a\,x^2}{2}+b\,x$

```
(%i2) integrate(a*x^2+b*x+c,x);
```

(%o2) $\frac{a\,x^3}{3}+\frac{b\,x^2}{2}+c\,x$

```
(%i3) integrate(exp(a*x+b),x);
```

(%o3) $\frac{e^{a\,x+b}}{a}$

```
(%i4) integrate(log(a*x+b),x);
```

(%o4) $\frac{(a\,x+b)\,\log(a\,x+b)-a\,x-b}{a}$

```
(%i5) integrate(cos(a*x+b),x);
```

(%o5) $\frac{\sin(a\,x+b)}{a}$

```
(%i6) integrate(sin(a*x+b),x);
```

(%o6) $-\frac{\cos(a\,x+b)}{a}$

5.11.2 Intégrales

```
(%i7) integrate(2*x+3, x, 0, 1);
```

(%o7) 4

```
(%i8) integrate(2*x^2+4*x-5,x, 1, 4);
```

(%o8) 57

```
(%i9) integrate(exp(-x+3),x, 3, 5);
```

(%o9) $1-e^{-2}$

```
(%i10) integrate(log(x-5),x, 6, 7);
```

(%o10) $2\log(2) - 1$

```
(%i11) integrate(cos(2*x),x, 0, %pi/3);
```

(%o11) $\frac{\sqrt{3}}{4}$

```
(%i12) integrate(sin(2*x),x, 0, %pi/3);
```

(%o12) $\frac{3}{4}$

Chapitre 6

Les suites

6.1 Définition d'une suite

Suites définies explicitement, comme une fonction de n :

```
(%i1) a(n):=n^2;
```

(%o1) $\mathrm{a}(n) := n^2$

ou comme une liste infinie :

```
(%i2) a[n]:=n^2;
```

(%o2) $a_n := n^2$

Suites définies par une relation de récurrence, comme une fonction de n :

```
(%i3) a(n):= if n = 0 then 4 else 2*a(n-1)-3$
      a(110);
```

(%o4) 1298074214633706907132624082305027

ou comme une liste infinie :

```
(%i5) a[n]:= if n = 0 then 4 else 2*a[n-1]-3$
      a[110];
```

(%o6) 1298074214633706907132624082305027

6.2 Expression explicite d'une suite définie par récurrence

On charge le module « solve_rec » :

```
(%i7) load("solve_rec");
```

(%o7) $C : /PROGRA\,2/MAXIMA\,1.0{-}2/share/maxima/5.28.0{-}2/share/solve\_rec/solve\_rec.mac$

```
(%i8) solve_rec(u[n]=2*u[n-1]-3, u[n]);
```

(%o8) $u_n = \%k_1\,2^n - 3\,2^n + 3$

Avec un terme initial :

```
(%i9) solve_rec(u[n]=2*u[n-1]-3, u[n], u[0]=4);
```

(%o9) $u_n = 2^n + 3$

On peut définir une suite de cette façon :

```
(%i10) define(v[n], rhs(%o9));
```

(%o10) $v_n := 2^n + 3$

```
(%i11) v[9];
```

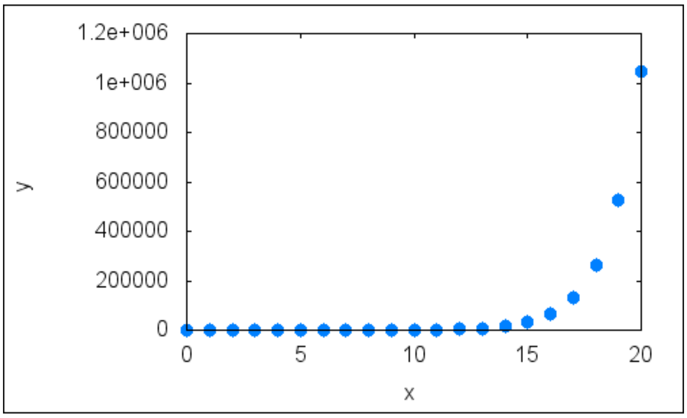
(%o11) 515

6.3 Représentation géométrique d'une suite

```
(%i12) entiers:makelist(n,n,0, 20);
       termesdelasuite:map(a,entiers);
       wxplot2d([discrete, entiers,termesdelasuite],[style, points]);
```

(%o12) [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]

(%o13) [4, 5, 7, 11, 19, 35, 67, 131, 259, 515, 1027, 2051, 4099, 8195, 16387, 32771, 65539, 131075, 262147, 524291, 1048579]

(%t14) 

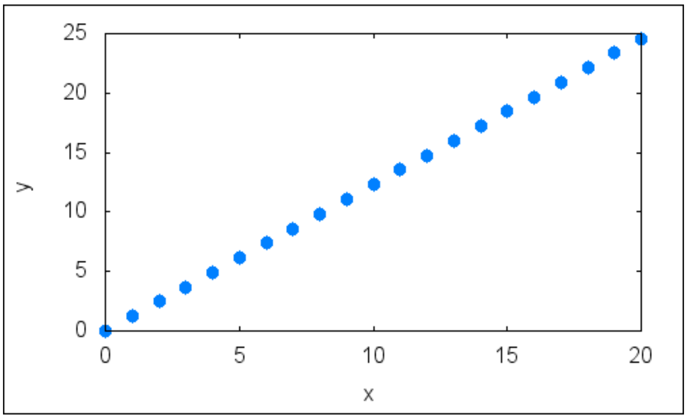
(%o14)

6.4 Suites arithmétiques

```
(%i15) r:1.23$
       a(n):= if n < 1 then 0 else a(n-1)+r$
       a(110);
```

(%o17) 135.3000000000001

```
(%i18) entiers:makelist(n,n,0, 20)$
       termesdelasuite:map(a,entiers)$
       wxplot2d([discrete, entiers,termesdelasuite],[style, points])$
```

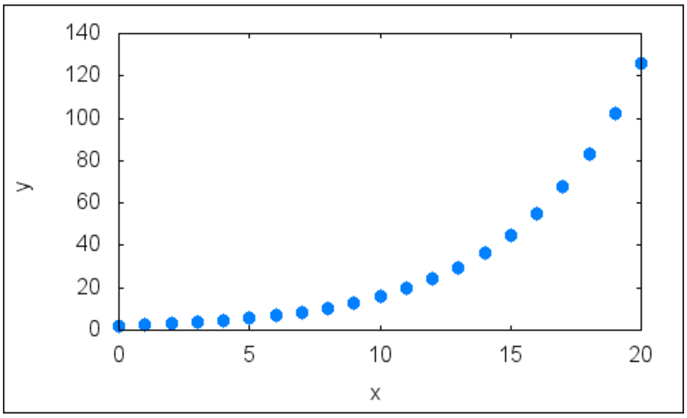
(%t20) 

6.5 Suites géométriques

```
(%i21) q:1.23$
       a(n):= if n < 1 then 2 else a(n-1)*q$
       a(10);
```

(%o23) 15.85189219210378

```
(%i24) entiers:makelist(n,n,0, 20)$
       termesdelasuite:map(a,entiers)$
       wxplot2d([discrete, entiers,termesdelasuite],[style, points])$
```

(%t26) 

6.6 Limite d'une suite

```
(%i27) a(n):=n^2;
       limit(a(n),n,inf);
```

(%o27) $a(n) := n^2$

(%o28) ∞

```
(%i29) a(n):=1/n^2;
       limit(a(n),n,inf);
```

(%o29) $a(n) := \frac{1}{n^2}$

(%o30) 0

6.7 Somme d'une suite

On utilise la commande sum :

```
(%i31) sum(a(k), k, 1, 10);
```

(%o31) $\frac{1968329}{1270080}$

6.8 Le module functs

```
(%i32) load("functs");
```

(%o32) $C:/PROGRA2/MAXIMA1.0-2/share/maxima/5.28.0-2/share/simplification/functs.mac$

Suites arithmétiques arithmetic(a, r, n) retourne le n-ième terme de la suite arithmétique de raison r et de premier terme a :

```
(%i33) arithmetic(1,2,3);
```

(%o33) 5

geometric(a, q, n) retourne le n-ième terme de la suite arithmétique de raison r et de premier terme a :

```
(%i34) geometric(1,2,3);
```

(%o34) 4

Somme des termes des suites arithmétiques et géométriques :

```
(%i35) arithsum(1,2,3);
```

(%o35) 9

```
(%i36) geosum(1,2,3);
```

(%o36) 7

Chapitre 7

Les nombres complexes

7.1 Forme algébrique d'un nombre complexe

Définition d'un nombre complexe

```
(%i1) z1:2+3*%i;
```

(%o1) $3\,i+2$

```
(%i2) z2:5+3*%i;
```

(%o2) $3\,i+5$

Somme et soustraction de deux nombres complexes

```
(%i3) z1+z2;
      z1-z2;
```

(%o3) $6\,i+7$

(%o4) -3

Produit de deux nombres complexes

Il faut développer l'expression avec expand si l'on souhaite avoir la forme algébrique :

```
(%i5) z1*z2;
      expand(z1*z2);
```

(%o5) $(3\,i+2)\,(3\,i+5)$

(%o6) $21\,i+1$

Pour avoir la forme algébrique d'un nombre complexe, on utilise la fonction rectform :

```
(%i7) z1/z2;
      rectform(z1/z2);
```

(%o7) $\frac{3\,i+2}{3\,i+5}$

(%o8) $\frac{9\,i}{34}+\frac{19}{34}$

Une nouvelle identité

(%i9)
```
declare(a,mainvar)$
expand((a-b*%i)*(a+b*%i));
```

(%o10) $a^2 + b^2$

Partie réelle

(%i11)
```
realpart(z1);
```

(%o11) 2

Partie imaginaire

(%i12)
```
imagpart(z1);
```

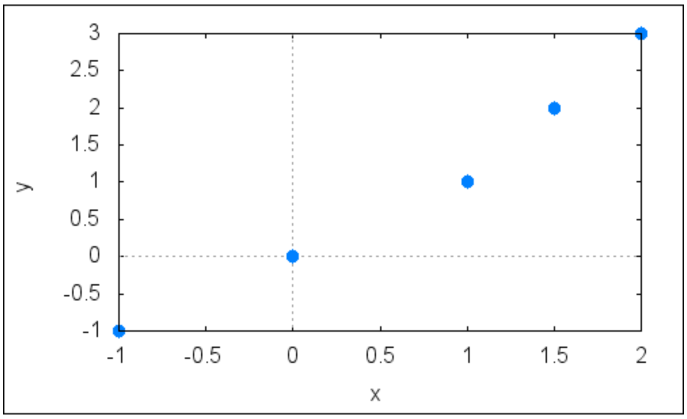
(%o12) 3

7.2 Représentation géométrique d'un nombre complexe

Définition d'une fonction utilisée pour placer les images des nombres complexes dans le plan complexe :

(%i13)
```
plotcomplex(Liste):=wxplot2d([discrete, makelist(realpart(Liste[k]), k, 1, length(Liste)),
makelist(imagpart(Liste[k]), k, 1, length(Liste))],[style, points])$
```

(%i14)
```
plotcomplex([1+%i, 2+3*%i, -1-%i, 1.5+2*%i, 0]);
```

(%t14) 

(%o14)

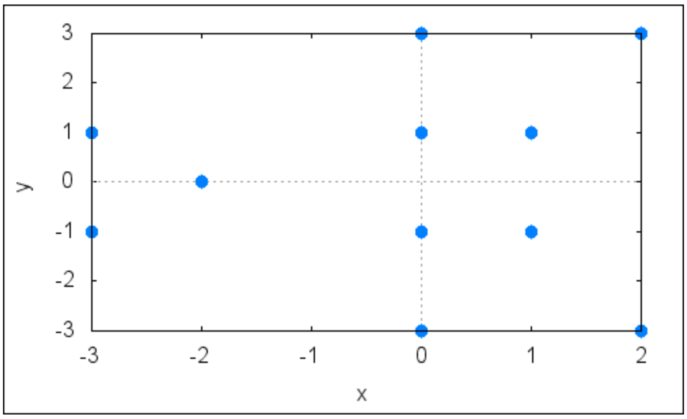
7.3 Nombre complexe conjugué

(%i15)
```
conjugate(1+%i);
```

(%o15) $1 - i$

Représentation géométrique d'un conjugué

(%i16)
```
liste:[1+%i, 2+3*%i, %i, -3+%i, 3*%i, -2]$
liste:append(liste, makelist(conjugate(liste[k]), k, 1, length(liste)))$
plotcomplex(liste);
```

(%t18) 

(%o18)

```
(%i19) conjugate[liste];
```

(%o19) $conjugate_{[i+1,3\,i+2,i,i-3,3\,i,-2,1-i,2-3\,i,-i,-i-3,-3\,i,-2]}$

7.4 Forme trigonométrique d'un nombre complexe

Définition d'un nombre complexe

```
(%i20) z:1+%i;
```

(%o20) $i+1$

Le module

```
(%i21) cabs(z);
```

(%o21) $\sqrt{2}$

Un argument

```
(%i22) carg(z);
```

(%o22) $\frac{\pi}{4}$

Définition de la forme trigonométrique

```
(%i23) formetrigo(z):=[cabs(z), carg(z)]$
```

```
(%i24) formetrigo(z);
```

(%o24) $[\sqrt{2},\frac{\pi}{4}]$

```
(%i25) formetrigo(1+%i*sqrt(3));
```

(%o25) $[2,\frac{\pi}{3}]$

7.5 Forme exponentielle

(%i26) `polarform(z);`

(%o26) $\sqrt{2}\, e^{\frac{i\,\pi}{4}}$

(%i27) `rectform(%);`

(%o27) $i+1$

(%i28) `z:exp(%i*theta);`

(%o28) $e^{i\,\theta}$

Définition de $e^{i\theta}$

(%i29) `rectform(z);`

(%o29) $i\sin(\theta)+\cos(\theta)$

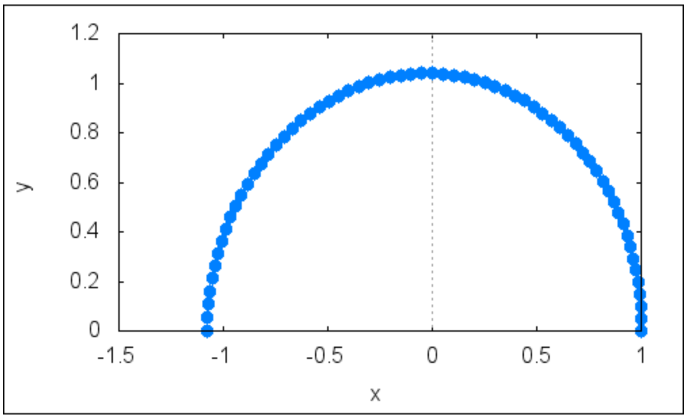
Cas particulier : $e^{i\pi}+1=0$

(%i30) `exp(%i*%pi);`

(%o30) -1

Illustration de l'égalité précédente et $\lim\limits_{n\to+\infty}\left(1+\frac{z}{n}\right)^n = e^z$

(%i31)
```
N:64$
liste:makelist((1+%i*%pi/N)^k, k, 0, N)$
plotcomplex(liste);
```

(%t33) 

(%o33)

Propriétés de la forme exponentielle

(%i34) `exp(%i*theta[1])*exp(%i*theta[2]);`

(%o34) $e^{i\,\theta_2+i\,\theta_1}$

(%i35) `1/exp(%i*theta);`

(%o35) $e^{-i\,\theta}$

```
(%i36) exp(%i*theta[1])/exp(%i*theta[2]);
```

(%o36) $e^{i\,\theta_1 - i\,\theta_2}$

```
(%i37) exp(%i*theta)^n;
```

(%o37) $e^{i\,n\,\theta}$

Formule de Moivre : $(i\sin(\theta) + \cos(\theta))^n = i\sin(n\theta) + cos(n\theta)$

```
(%i38) (cos(theta)+%i*sin(theta))^n;
```

(%o38) $(i\sin(\theta) + \cos(\theta))^n$

```
(%i39) trigrat((exp(%i*theta))^n);
```

(%o39) $i\sin(n\,\theta) + \cos(n\,\theta)$

En une seule commande

```
(%i40) trigrat((cos(theta)+%i*sin(theta))^n);
```

(%o40) $i\sin(n\,\theta) + \cos(n\,\theta)$

Formules d'Euler

```
(%i41) exponentialize(cos(theta));
```

(%o41) $\frac{e^{i\,\theta} + e^{-i\,\theta}}{2}$

```
(%i42) exponentialize(sin(theta));
```

(%o42) $-\frac{i\,\left(e^{i\,\theta} - e^{-i\,\theta}\right)}{2}$

7.6 Équations du second degré à coefficients constants

Maxima ne fait pas la distinction entre delta positif ou négatif :

```
(%i43) kill(z)$
```

```
(%i44) solutions:solve(a*z^2+b*z+c=0, z);
```

(%o44) $[z = -\frac{\sqrt{b^2 - 4\,c\,a} + b}{2\,a}, z = \frac{\sqrt{b^2 - 4\,c\,a} - b}{2\,a}]$

```
(%i45) solve(z^2+z-1=0,z);
```

(%o45) $[z = -\frac{\sqrt{5} + 1}{2}, z = \frac{\sqrt{5} - 1}{2}]$

```
(%i46) solve(z^2+2*z+1=0,z);
```

(%o46) $[z = -1]$

```
(%i47) solve(z^2+z+1=0,z);
```

(%o47) $[z = -\frac{\sqrt{3}\,i + 1}{2}, z = \frac{\sqrt{3}\,i - 1}{2}]$

Pour vérifier qu'un nombre complexe annule une expression :

```
(%i48) subst(-1, z, z^2+z+1);
```

(%o48) 1

Chapitre 8

Listes et statistiques descriptives

8.1 Listes

Définition d'une liste :

```
(%i1) L:[30, 14, 15, 16, 14, 15, 18, 19, 1 , 6 ,18 ];
```

(%o1) $[30, 14, 15, 16, 14, 15, 18, 19, 1, 6, 18]$

Longueur d'une liste :

```
(%i2) length(L);
```

(%o2) 11

Concaténer deux listes :

```
(%i3) append(L,[0, 2, 3]);
```

(%o3) $[30, 14, 15, 16, 14, 15, 18, 19, 1, 6, 18, 0, 2, 3]$

Ordonner la liste :

```
(%i4) sort(L);
```

(%o4) $[1, 6, 14, 14, 15, 15, 16, 18, 18, 19, 30]$

Créer un tableau de valeur d'une fonction :

```
(%i5) f(x):=x^2+1;
```

(%o5) $\mathrm{f}(x) := x^2 + 1$

On crée une liste de nombres régulièrement espacés, par exemple de 0,1 :

```
(%i6) nombres:makelist(k/10,k,0,10);
```

(%o6) $[0, \frac{1}{10}, \frac{1}{5}, \frac{3}{10}, \frac{2}{5}, \frac{1}{2}, \frac{3}{5}, \frac{7}{10}, \frac{4}{5}, \frac{9}{10}, 1]$

On crée la liste des images par f de ces nombres :

```
(%i7) fnombres:f(nombres);
```

(%o7) $[1, \frac{101}{100}, \frac{26}{25}, \frac{109}{100}, \frac{29}{25}, \frac{5}{4}, \frac{34}{25}, \frac{149}{100}, \frac{41}{25}, \frac{181}{100}, 2]$

Si on souhaite avoir les valeurs approchées :

```
(%i8) anombres:float(fnombres);
```

(%o8) $[1.0, 1.01, 1.04, 1.09, 1.16, 1.25, 1.36, 1.49, 1.64, 1.81, 2.0]$

8.2 Statistiques descriptives

```
(%i9) load(descriptive);
```

(%o9) $C:/PROGRA~2/MAXIMA~1.0-2/share/maxima/5.28.0-2/share/descriptive/descriptive.mac$

Valeur moyenne de L :

```
(%i10) mean(L);
```

(%o10) $\frac{166}{11}$

```
(%i11) float(mean(L));
```

(%o11) 15.09090909090909

Écart-type :

```
(%i12) std(L);
```

(%o12) $\frac{2\sqrt{1482}}{11}$

```
(%i13) float(std(L));
```

(%o13) 6.999409656334603

Minimum de L :

```
(%i14) mini(L);
```

(%o14) 1

Macimum de L :

```
(%i15) maxi(L);
```

(%o15) 30

Étendue :

```
(%i16) range(L);
```

(%o16) 29

Médiane :

```
(%i17) median(L);
```

(%o17) 15

Quantiles :

```
(%i18) quantile(L,1/4);
```

(%o18) 14

```
(%i19) quantile(L,2/4);
```

(%o19) 15

```
(%i20) quantile(L,3/4);
```

(%o20) 18

```
(%i21) quantile(L,1/10);
```

(%o21) 6

8.3 Listes avec pondérations

Liste des valeurs :

```
(%i22) L:[12,13,15,16];
```

(%o22) $[12, 13, 15, 16]$

Liste des effectifs ou des fréquences :

```
(%i23) F:[1,2,3,4];
```

(%o23) $[1, 2, 3, 4]$

```
(%i24) moyenne(L_1, L_2):=block ( [n:length(L_1)],

                       sum(L_1[k]*L_2[k],k,1,n)/sum(L_2[k],k,1,n)
                    )$
```

```
(%i25) variance(L_1, L_2):=block ( [n:length(L_1)],

                       moyenne(L_1^2,L_2)-moyenne(L_1, L_2)^2
                    )$
```

```
(%i26) ecartype(L_1, L_2):=sqrt(variance(L_1, L_2))$
```

```
(%i27) moyenne(L,F);
       variance(L,F);
       ecartype(L,F);
```

(%o27) $\frac{147}{10}$

(%o28) $\frac{201}{100}$

(%o29) $\frac{\sqrt{201}}{10}$

Chapitre 9

Probabilités

9.1 Simulations

9.1.1 Simulation du lancer de pièce et loi de Bernoulli

On utilise la commande random(2), pour avoir 0 ou 1 de façon aléatoire et équiprobable :

```
(%i1) random(2);
      random(2);
      random(2);
```

(%o1) 0

(%o2) 0

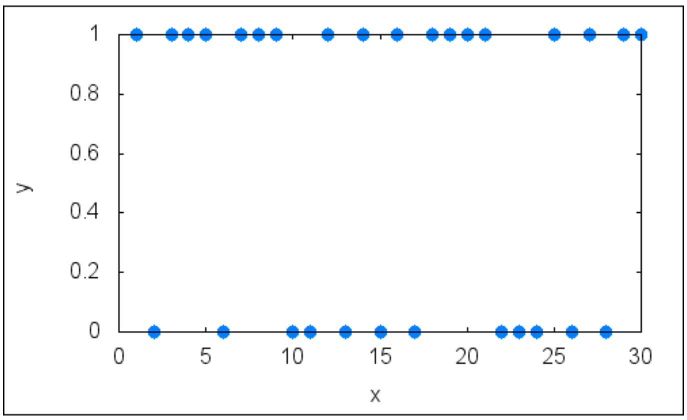
(%o3) 0

Liste de simulations du lancer de pièce

```
(%i4) Liste:[]$
      N:30$
      for i from 1 thru N do Liste:append(Liste,[random(2)])$
      Liste;
      entiers:makelist(n,n,1,N)$
      wxplot2d([discrete, entiers, Liste], [style, points]);
```

(%o7) [1, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 0, 1, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0, 1, 1]

(%t9)



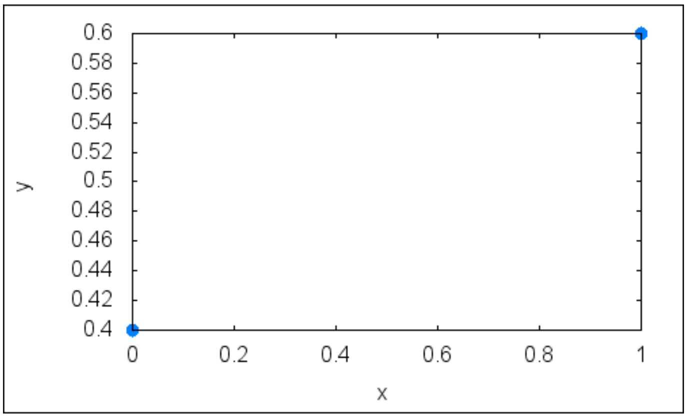
(%o9)

Affichage des fréquences

```
(%i10) zeroUn:[0, 1]$
       f1:sum(Liste[k],k, 1, N)/N$
       f0:1-f1$
       float(f0);
       float(f1);
       ListeFrequences:[f0, f1]$
       wxplot2d([discrete, zeroUn, ListeFrequences], [style, points]);
```

(%o13) 0.4

(%o14) 0.6

(%t16) 

(%o16)

La commande random(1.0) permet d'obtenir un nombre de l'intervalle $[0; 1[$.

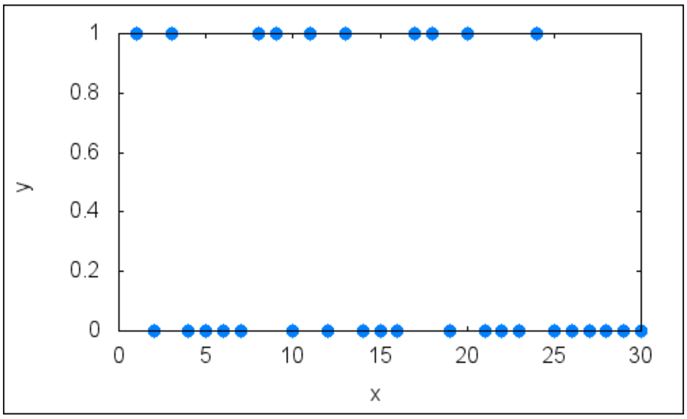
Pour obtenir un lancer de pièce truquée, où la probabilité d'obtenir le 1 est p, on utilise la commande ci-dessous :

```
(%i17) p:0.3$
       random(1.0)+ p;
```

(%o18) 1.152705109085575

```
(%i19) Liste:[]$
       N:30$
       for i from 1 thru N do Liste:append(Liste,[floor(random(1.0)+p)])$
       Liste;
       entiers:makelist(n,n,1,N)$
       wxplot2d([discrete, entiers, Liste], [style, points]);
```

(%o22) [1, 0, 1, 0, 0, 0, 0, 1, 1, 0, 1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0]

(%t24) 

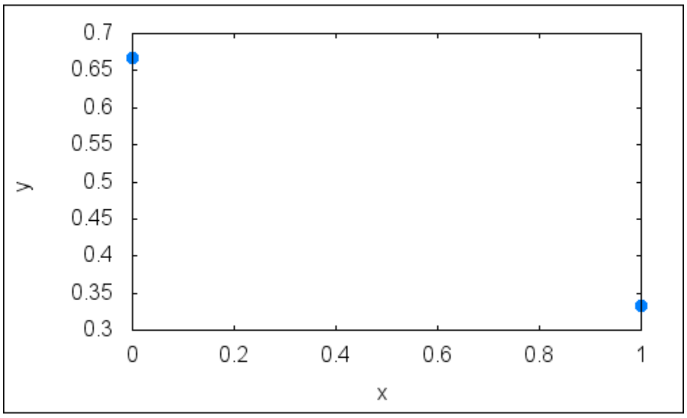
(%o24)

Affichage des fréquences

```
(%i25) zeroUn:[0, 1]$
       f1:sum(Liste[k],k, 1, N)/N$
       f0:1-f1$
       float(f0);
       float(f1);
       ListeFrequences:[f0, f1]$
       wxplot2d([discrete, zeroUn, ListeFrequences], [style, points]);
```

(%o28) 0.66666666666667

(%o29) 0.33333333333333

(%t31) 

(%o31)

9.1.2 Simulation du lancer de dé

Simulation du lancer de dé

```
(%i32) random(7);
       random(7);
       random(7);
       random(7);
       random(7);
       random(7);
       random(7);
```

(%o32) 0