Pour commencer avec XCAS

Thomas Rey
http://reymarlioz.free.fr

19 octobre 2011

1 Présentation

Xcas est un logiciel multi-fonctions de mathématiques. Il permet d'effectuer des calculs numériques, du calcul formel (c'est-à-dire avec des lettres !), de la géométrie, des représentations de courbes et surfaces, du tableur, des statistiques mais aussi de programmer. Il s'agit d'un logiciel gratuit et multi-plateforme (c'est-à-dire que vous pouvez l'installer sous Windows, Linux et Mac OSX). Vous pouvez le télécharger [1] ou même le tester en ligne [2]. Il est développé depuis 2000 par Bernard Parisse et documenté par René De Graeve.
Ce document est très loin d'être exhaustif, il permet juste de se familiariser avec le logiciel et ses commandes de base. Il est recommandé d'utiliser l'aide du logiciel (par l'index ou la recherche) ou encore de consulter les nombreux documents disponibles sur le site d'Xcas
Au démarrage, vous obtenez un fenêtre qui ressemble à la figure 1.

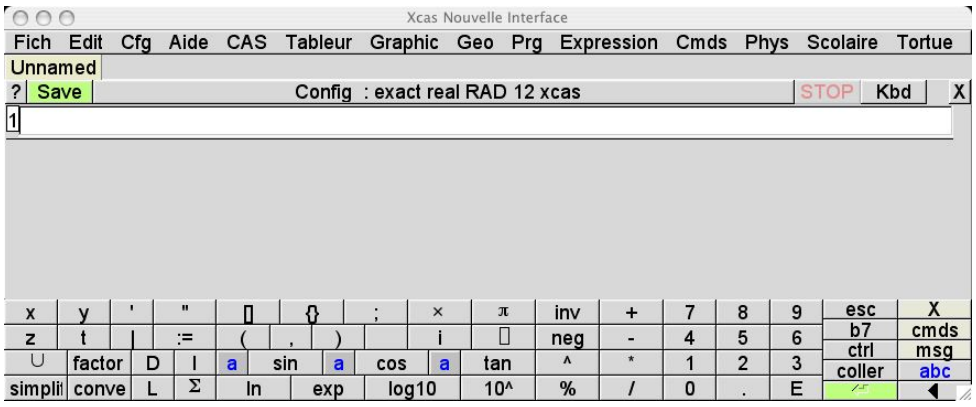

Figure 1 – Fenêtre de démarrage

Chaque calcul, programme, ... est effectué sur une ligne blanche numérotée. On peut retourner sur une ligne précédente pour modifier une instruction, un calcul, ... Pour créer une nouvelle feuille de calcul de tableur, une nouvelle figure de géométrie, un nouveau programme, ..., on clique [3] sur le menu correspondant puis sur nouveau tableur, nouvelle figure, ... Le plus simple pour apprendre étant de manipuler, allons-y...

1. www-fourier.ujf-grenoble.fr/~parisse/install_fr
2. http://vds1100.sivit.org/giac/giac_online/xcasenligne.html
3. Ou on apprend les raccourcis claviers, c'est encore plus rapide !

2 Calculs

La plupart des commandes décrites dans ce paragraphe sont accessible dans les menus `CAS` et `Cmds`. Elles y sont rangées par thème, il suffit souvent de chercher dans ces menus lorsqu'on cherche une commande particulière. En cas de doute, penser à l'aide en recherchant un mot (touche F12) ou par l'index.

2.1 Calcul numérique

Xcas est capable d'effectuer toutes les opérations courantes, bien sûr en respectant les priorités opératoires. Ainsi en saisissant `10-4*3`, la réponse proposée est `-2`.
Plus intéressant maintenant, Xcas calcule en valeur exacte ; par exemple en saisissant dans une nouvelle ligne `(1/3+5/4)/(7/4)`, le résultat donné est $\frac{19}{21}$. Si on souhaite une valeur approchée de ce résultat, on peut cliquer sur le petit `M` à droite de la ligne du résultat et `Sélectionner tout` puis à nouveau sur le `M` et `Evalf` ; on pouvait aussi saisir directement `evalf((1/3+5/4)/(7/4))` ou même plus simplement `(1./3+5/4)/(7/4))` : la présence d'un seul nombre décimal (le « `1.` ») indique à Xcas qu'on souhaite une valeur approchée.
Toujours plus intéressant, les calculs avec radicaux. Saisir `(sqrt(2)+3)*(sqrt(2)-5)` (`sqrt` signifie *square root* : racine carrée) ; sélectionner à nouveau tout (avec le « `M` ») puis `Normale` ou `Simplifier`. Il affiche la réponse sous la forme $a\sqrt{b}+c$.
Autre calcul : `5*sqrt(8)-3*sqrt(2)+sqrt(32)`, sélectionner tout puis simplifier : le résultat donne $11\sqrt{2}$.
Attention : le « `*` » situé entre les nombres et le `sqrt` est obligatoire : on ne peut pas écrire `2sqrt(2)`.

Exemple 1
Quelques exemples à retenir :

Saisie dans Xcas	Résultat affiché
`1/3+4/5`	$\frac{17}{15}$
`evalf(1/3+4/5)`	`1.13333333333`
`1./3+4/5`	`1.13333333333`
`simplifier(2*sqrt(2)-5*sqrt(8))`	$-8\sqrt{2}$
`evalf(sqrt(2),20)`	`1.41421356237309504880`
`evalf(pi,5)`	`3.14159`
`cos(pi/6)`	$\frac{\sqrt{3}}{2}$
`acos(1/2)`	$\frac{1}{3}\pi$

Pour terminer, la fonction `ans(-1)` donne le dernier résultat trouvé, `ans(-2)` l'avant-dernier, ... et `ans(0)`, donne la première réponse depuis l'ouverture de la session, `ans(1)` la deuxième, ...
Attention : en saisissant `2+2` comme première instruction (et en validant), puis en modifiant la ligne 1 en `2+3`, `ans(-1)` donnera 5 et `ans(-2)` donnera 4. De même, `ans(0)` et `ans(1)` donneront respectivement 4 et 5 : il ne faut donc pas se fier au numéro de la ligne !

2.2 Calcul littéral

2.2.1 Pour tous

Dans cette partie nous allons exploiter les capacités de calcul formel du logiciel Xcas.
La première fonction à connaître est la fonction `developper` ; par exemple si vous oubliez les identités remarquables, `developper((a+b)^2)` donne sa forme développée.
Attention : `developper((x+1)(2x+5))` n'est pas une écriture correcte : il faut utiliser le `*` entre les deux paires de parenthèses : `developper((x+1)*(2x+5))`. Par contre le `2x` est correct.
Xcas peut développer, il peut aussi factoriser : `factoriser(2x^2-12x+18)` donne $2(x-3)^2$.
La résolution d'équations n'est pas plus compliquée : `resoudre(x^2-4x-1)` donne $-\sqrt{5}+2$ et $\sqrt{5}+2$.
Encore plus fort, la résolution d'une équation où figure un paramètre par exemple on souhaite trouver les abscisses des points d'intersection de la parabole d'équation $y = x^2$ avec la droite d'équation $y = x+m$ où m est un réel quelconque. Il s'agit donc de résoudre l'équation d'inconnue x : $x^2 = x + m$. On saisit donc `resoudre(x^2=x+m,x)` ; le « `,x` » signifie que l'inconnue est x. On obtient $\frac{1-\sqrt{4m+1}}{2}$ et $\frac{1-\sqrt{4m+1}}{2}$ (bien entendu si $4m+1 \geq 0$, sinon il n'y a pas de solution).
On peut aussi définir une fonction en saisissant `f(x):=2x^2-x-1` ; par la suite on peut calculer des images : `f(3)` ou trouver des antécédents : `resoudre(f(x)=1)`.
Attention : pour définir une fonction, ou attribuer une valeur à une variable (une lettre par exemple), il faut bien écrire `f(x):=2x+1`, ou `a:=5` (ne pas oublier le « : »).

2.2.2 À partir de la première

La nouveauté principale du programme de première concernant les études de fonctions est la dérivation. Tout élève de première connaît parfaitement ses formules de dérivation, néanmoins, pour vérifier Xcas a la commande `derive` ou encore pour pouvoir travailler sur la dérivée la commande `fonction_derivee`. Par exemple `derive(2x^2-5x+1,x)` nous donne bien `4x-5`. À noter le « `,x` qui signifie qu'on dérive par rapport à x, il joue le rôle du $\frac{\cdot}{\mathrm{d}x}$ des physiciens.
Xcas est aussi capable de calculer des limites : `limite((2x-3)/(x-1),x,+infinity)` calcule $\lim\limits_{x\to+\infty} \dfrac{2x-3}{x-1} (= 2)$.
$\lim\limits_{\substack{x\to1\\x<1}} \dfrac{2x-3}{x-1}$ s'écrit : `limite((2x-3)/(x-1),x,1,-1)`
$\lim\limits_{\substack{x\to1\\x>1}} \dfrac{2x-3}{x-1}$ s'écrit : `limite((2x-3)/(x-1),x,1,1)`

Exemple 2
Étudions la fonction $f : x \mapsto \frac{x^2+3}{x+1}$.
On définit f dans Xcas par `f(x):=(x^2+3)/(x+1)`.
On cherche les valeurs interdites : `resoudre(x+1=0)`.
Nommons g sa dérivée : `g:=fonction_derivee(f)` et écrivons $g(x)$ sous forme factorisée : `factoriser(g(x))`.
On résout $f'(x) = 0$: `resoudre(g(x)=0,x)` et on trouve deux solutions : -3 et 1.

On résout $f'(x) > 0$: `resoudre(g(x)>0,x)` et on trouve `[x<(-3) x>1]` ce qui signifie que $g(x)$ est strictement positif pour $x \in]-\infty\,;\,-3[\cup]1\,;\,+\infty[$.
On complète par le calcul des valeurs de $f(-6)$ et de $f(1)$ grâce à `f(-3)` puis `f(1)` qui donnent respectivement -6 et 2.
On calcule également les quatre limites :
`limite(f(x),x,-infinity)` et `limite(f(x),x,+infinity)` qui donnent $-\infty$ et $+\infty$ respectivement, ainsi que `limite(f(x),x,-1,-1)` et `limite(f(x),x,-1,1)` qui donnent $-\infty$ et $+\infty$ respectivement.
Attention : le `+` du `+infinity` est indispensable (on peu aussi remplacer `+infinity` par `inf`).
Avec toutes ces informations on complète donc facilement le tableau de variation de f :

x	$-\infty$		-3		-1		1		$+\infty$
$f'(x)$		$+$	0	$-$	‖	$-$	0	$+$	
f	$-\infty$	↗	-6	↘	‖ $+\infty$	↘	2	↗	$+\infty$

Nous verrons dans la partie 3 comment tracer la courbe représentative de f dans un repère.
Pour les profs de maths qui utilisent LaTeXpour écrire leurs cours, il existe même les extensions « Professor » et « Tablor » créée par G. Connan et D. LeFur permettant d'intégrer directement le tableau de variation dans le document en ne précisant que l'expression algébrique de la fonction et l'intervalle d'étude...

Autre nouveauté en première, la forme canonique d'un polynôme du second degré ; elle s'obtient en saisissant `forme_canonique(x^2+x+1)` par exemple.

2.2.3 À partir de la terminale

En terminale, des nouvelles fonctions apparaissent : l'exponentielle et les logarithmes. La définition de l'exponentielle dans Xcas est `exp`, le logarithme népérien est `ln`, le logarithme de base 10 est `log10`.
Autre nouveauté de la classe de terminale : les primitives. Une primitive de la fonction f est obtenue par la saisie de `integrer(f(x),x)`.
Enfin, $\int_0^1 f(x)\mathrm{d}x$ s'obtient par `integrer(f(x),x,0,1)`.

Pour la série S :
Les nombres complexes s'écrivent avec la lettre i (minuscule). Ainsi `(2+i)*(1-i)` donne `3-i`.
`conj(z)` renvoie le conjugué deu nombre complexe z.
`arg(z)` renvoie un argument du nombre complexe z (dans $]-\pi\,;\,+\pi]$).
`abs(z)` renvoie le module du nombre complexe z.
`re(z)` et `im(z)` renvoient respectivement les parties réelle et imaginaire de z.
`evalc(z)` renvoie l'écriture algébrique de z.

Enfin, `csolve(x^2+x+1=0,x)` résout dans $\mathbf{C}$ l'équation $x^2+x+1=0$ et renvoie donc $\frac{-1+i\sqrt{3}}{2}$ et $\frac{-1-i\sqrt{3}}{2}$. On peut aussi utiliser `resoudre_dans_C(x^2+x+1=0)` mais c'est plus long à écrire.

2.2.4 Récapitulons...

Exemple 3
Quelques exemples à retenir :

Définition mathématique	Saisie dans Xcas	Résultat affiché
Définir une fonction	`f(x):=2x^2-5x+1`	$x \mapsto 2x^2-5x+1$
L'image de 2 par f	`f(2)`	-1
Développer	`developper((x+3)*(x-1)^2)`	x^3+x^2-5x+3
Réduire, ...	`simplifier(2x-5x^2+4x*(x+1))`	$-x^2+6x$
Factoriser	`factoriser(x^2-x-6)`	$(x-3)(x+2)$
Forme canonique	`forme_canonique(x^2+x+1)`	$\left(x+\frac{1}{2}\right)^2+\frac{3}{4}$
Résoudre une équation	`resoudre(2x^2-3x=2x-3,x)`	1 et $\frac{3}{2}$
Résoudre une inéquation	`resoudre(x^2-1>0)`	$x<(-1)$ ou $x>1$
Dériver une fonction	`derive(x^2+x+1+cos(x),x)`	$2x+1-\sin(x)$
$\lim_{x\to+\infty} \frac{2x-4}{x+5}$	`limite((2x-4)/(x+5),x,+infinity)`	2
$\lim_{\substack{x\to-5\\x>-5}} \frac{2x-4}{x+5}$	`limite((2x-4)/(x+5),x,-5,1)`	$-\infty$
$\lim_{\substack{x\to-5\\x<-5}} \frac{2x-4}{x+5}$	`limite((2x-4)/(x+5),x,-5,-1)`	$+\infty$
$f(x)=\mathrm{e}^x$	`f(x):=exp(x)`	$x \mapsto \exp(x)$
$g(x)=\ln(x)$	`g(x):=ln(x)`	
$h(x)=\log_{10}(x)$	`h(x):=log10(x)`	$x \mapsto \ln(x)$
Une primitive de ln	`F(x):=integrer(ln(x),x)`	$x\ln(x)-x$
$\int_1^2 \frac{1}{x}\mathrm{d}x$	`integrer(1/x,x,1,2)`	$\ln(2)$
$z=2\sqrt{3}+2\mathrm{i}$	`z:=2*sqrt(3)+2*i`	$2\sqrt{3}+2\mathrm{i}$
$\overline{z}$	`conj(z)`	$2\sqrt{3}+-2\mathrm{i}$
$\arg(z)$	`simplifier(arg(z))`	$\frac{\pi}{6}$
$\lvert z\rvert$	`simplifier(abs(z))`	4
$\mathrm{Re}(z)$	`re(z)`	$2\sqrt{3}$
$\mathrm{Im}(z)$	`im(z)`	2
Résoudre $x^2+1=0$ dans $\mathbf{C}$	`csolve(x^2+1=0,x)`	$-\mathrm{i}$ et i

2.3 Arithmétique

Quelques commandes de base dans la tableau ci-dessous :

Définition mathématique	Saisie dans Xcas
pgcd(a,b)	`gcd(a,b)`
ppcm(a,b)	`lcm(a,b)`
quotient de la div. eucl. de a par b	`iquo(a,b)`
reste de la div. eucl. de a par b	`irem(a,b)`
coefficients de Bezout	`bezout_entiers(a,b)`

`bezout_entier(12,16)` renvoie -1, 1 et 4 car $-1 \times 12 + 1 \times 16 = 4$;
`bezout_entier(16,21)` renvoie 4, -3 et 1 car $4 \times 16 + (-3) \times 21 = 1$ (16 et 21 sont premiers entre eux).

2.4 Statistiques et probabilités

Pour créer une liste de nombres, de caractères, ... on utilise les crochets :
`MaListe:=[1,4,7,9]`.
On peut ensuite obtenir un certain nombre de paramètres pour cette liste alors assimilée à une série statistique à une variable :

Commande Xcas	Résultat
`size(MaListe)`	donne le nombre d'éléments de la liste
`moyenne(MaListe)`	donne la moyenne de la série
`mediane(MaListe)`	donne la médiane de la série
`quartile1(MaListe)`	donne le premier quartile de la série
`quartile3(MaListe)`	donne le troisième quartile de la série
`ecart_type(MaListe)`	donne l'écart-type de la série
`moustache(MaListe)`	trace la boîte à moustaches de la série

Remarques :
– pour obtenir les valeurs approchées des paramètres il suffit d'écrire une valeur de la série sous forme décimale par exemple `2.0` pour `2` ;
– dans le cas de séries données par un tableau d'effectifs par exemple :

valeur	4	5	7	10
effectif	2	6	1	7

on saisit : `Valeurs:=[4,5,7,10]` et `Effectifs:=[2,6,1,7]`. La moyenne s'obtient par la saisie de `moyenne(Valeurs,Effectifs)` et il en est de même pour les autres paramètres statistiques.

Pour les séries à deux variables stockées dans deux listes `A` et `B` on a :

Commande Xcas	Résultat
`covariance(A,B)`	donne la covariance des séries A et B
`linear_regression(A,B)`	donne les coefficients a et b de la droite d'ajustement obtenue par la méthode des moindres carrés
`linear_regression_plot(A,B)`	trace la droite précédente
...	

Pour effectuer des simulations d'expérience aléatoires, on peut utiliser les commandes de nombres aléatoires suivantes :

Commande Xcas	Résultat
`alea()`	donne un entier aléatoire inférieur à $< 10^{32}$
`alea(n)`	donne un entier compris entre 0 et $n-1$ inclus
`alea(3,4,25)`	donne une liste de 3 entiers compris entre 4 et 25
`alea(5,A)`	donne 5 éléments distincts de la liste A (où A est une liste d'au moins 5 éléments)
`alea(4,5)`	donne un « réel » aléatoire compris entre 4 et 5

En probabilités, les coefficients binomiaux C_n^p qu'on note aussi $\begin{pmatrix} n \\ p \end{pmatrix}$ s'obtient par la commande `comb(n,p)`. De même `perm(n,p)` donne le nombre d'arrangement de p éléments parmi n. Enfin, `factorial(n)` donne $n!$.

3 Représentations graphiques

Si on a plusieurs courbes représentatives ou plusieurs surfaces de l'espace à tracer successivement on pourra utiliser avec profit le module `Geo` en cliquant sur le menu du même nom (voir la partie 5 page 12).

3.1 Dans le plan

`graphe([cos(x),sin(x)],x=-pi..pi,color=[rouge,bleu])` trace les représentations graphiques des fonctions cosinus et sinus restreintes à l'intervalle $[-\pi\,;\,\pi]$. On obtient alors une fenêtre semblable à la figure 2.

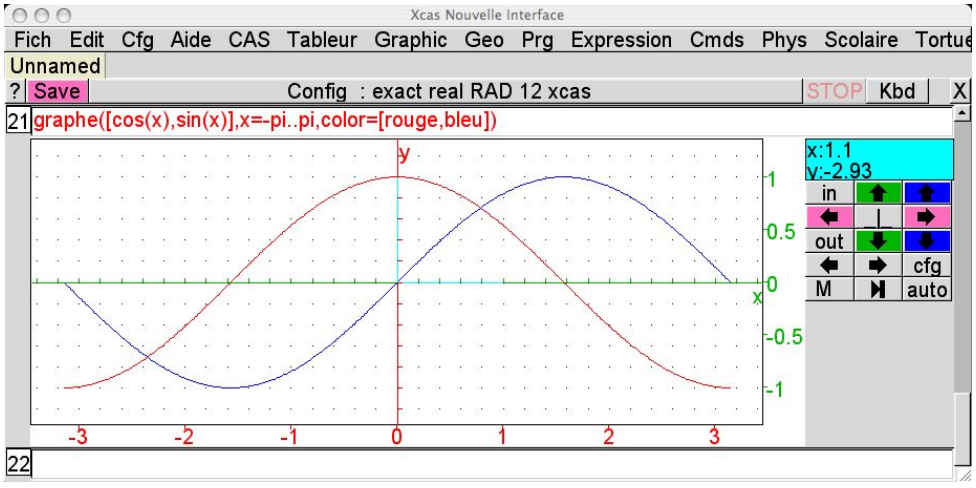

FIGURE 2 – Représentations graphiques des fonctions sinus et cosinus

Intéressons nous aux boutons situés à droite du repère :
- le bouton situé au centre des quatre flèches permet de rendre le repère orthonormé ;
- les quatre flèches permettent de déplacer la fenêtre graphique (beaucoup plus pratique qu'à la calculatrice !) ;

– les boutons `in` et `out` permettent de zoomer vers l'avant ou vers l'arrière ;
– en cliquant sur le `M` on obtient un menu `Export print` permettant de sauvegarder le graphique sous forme d'une image ou de l'imprimer ;
– le bouton `cfg` permet de régler la fenêtre graphique « à la main » ainsi que d'autres paramètres ;
– enfin, en promenant le curseur de la souris sur le graphique, on peut lire les coordonnées du point correspondant au dessus des boutons décrits ci-dessus.

3.2 Dans l'espace

Une fonction comparable permet de représenter des surfaces de l'espaces définies par une fonction de deux variables ou par des équations paramétriques. Deux exemples non détaillés ci-dessous :

Exemple 4
Avec une fonction de deux variables définie par $f(x, y) = x^2 + y^2 - 5$:
`graphe3d(x^2+y^2-5,x=-9..9,y=-9..9)` donne la figure 3 de la page 8.
Avec `plotfunc(x^2+y^2-5,[x=-9..9,y=-9..9],xstep=0.5,ystep=0.5,display=cyan+rempli)` on a même la couleur !

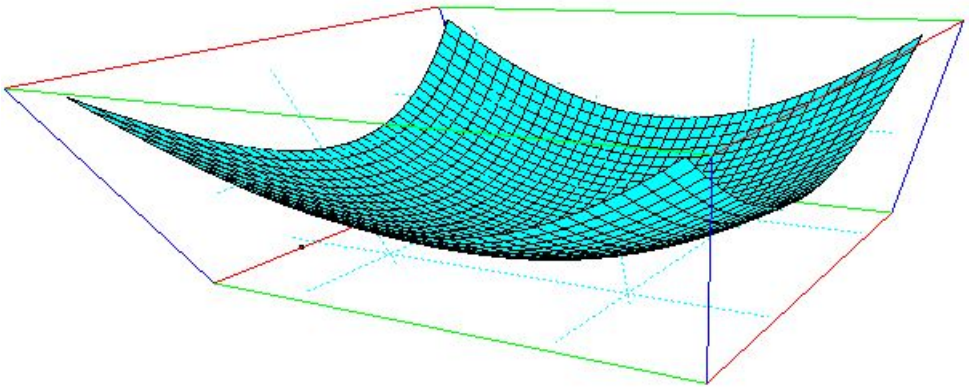

FIGURE 3 – Un paraboloïde

Exemple 5
Avec un système d'équations paramétriques :

$$\begin{cases} x = \cos(u)\sin(v) \\ y = \sin(u)\sin(v) \\ z = \cos(v) \end{cases} , u \in [-\pi\,;\,\pi], v \in [0\,;\,\pi]$$

`graphe3d([cos(u)*sin(v),sin(u)*sin(v),cos(v)],u=-pi..pi,v=0..pi)` donne la figure 4 de la page 9, et en couleur : `plotparam([cos(u)*sin(v),sin(u)*sin(v),cos(v)],[u=-pi..pi,v=0..pi],display=magenta+rempli)`

Pour chacune de ces figures, on peut sélectionner à la souris un coin du parallélépipède entourant la surface et faire tourner cette surface pour obtenir plusieurs vues différentes.

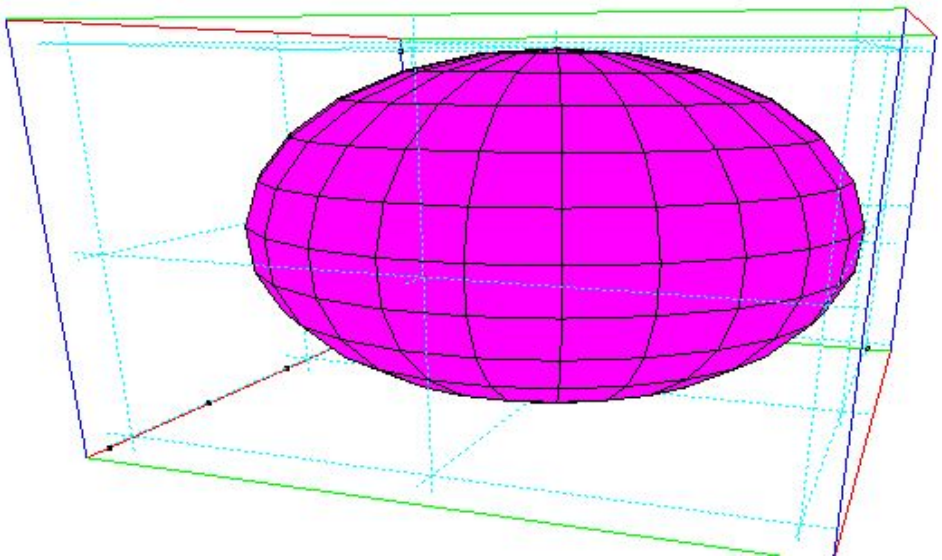

FIGURE 4 – Une surface paramétrée

On peut également effectuer un certain nombre de réglages grâce aux boutons situés à droite de la figure.

4 Programmation

Pour écrire un programme, il faut se placer sur une ligne vierge, et cliquer sur `Prg` puis sur `Nouveau Programme` ; effacer le « : ; » s'il existe. Vous êtes prêts à programmer.

4.1 Échangisme...

Pour créer un programme, il faut lui donner un nom, par exemple `Echange` et lui indiquer les variables dont il a besoin (ici deux nombres a et b à « échanger »). Pour cela, on écrit : `Echange(a,b):={`
On écrira notre programme après l'accolade ouvrante `{` et on le conclura par une accolade fermante `}` (qu'on peut écrire tout de suite pour ne pas l'oublier...).
Avec Xcas, *l'affectation* c'est-à-dire le « remplissage d'une variable » par une information (ce qu'on indique généralement dans les algorithmes par « ← ») se fait avec « `:=` ». Ainsi pour affecter à la variable c le nombre 5 on écrit `c:=5;` (**Chaque ligne du programme doit se terminer par un « ; »**).

Traduisons l'algorithme ci-dessous en langage Xcas :

Entrées :
nombres a et b ;
Sorties :
le contenu de a et b après échange ;
début
    $a \leftarrow a+b$;
    $b \leftarrow a-b$;
    $a \leftarrow a-b$;
    Afficher a, b
fin

Algorithme 1 : Échange du contenu des variables a et b

Le programme Xcas :

```
Echange(a,b):={
// Tout ce qui est placé après deux "/" est ignoré par Xcas
a:=a+b;// Il est important de commenter ses programmes.
b:=a-b;
a:=a-b;
return a,b;// l'instruction return permet de renvoyer un
   résultat à l'utilisateur du programme
}
```

Une fois le programme écrit, il faut le *compiler*[4] en cliquant sur `OK`. Si tout se passe bien, un message « `Success compiling Echange` » apparaît. Dans une nouvelle ligne, taper `Echange(5,3)` et constater le résultat.
Pour enregistrer ce programme, Cliquer sur `Prog` (juste à côté du numéro de la ligne où vous avez écrit votre programme), puis sur `Sauver comme`. Sélectionner le dossier `Mes documents` et saisir le nom `Echange`.
Remarque : ce programme n'est qu'un exemple d'utilisation des affectations car avec Xcas, il suffit d'écrire `a,b:=b,a` pour que les contenus des variables a et b soient échangés.

4.2 Tests, boucles, . . .

Écrivons un programme qui teste si deux nombres contenus dans les variables a et b sont égaux.
Pour programmer la condition **Si. . .** , cliquer sur `ajouter` puis sur `test` et enfin sur `si alors sinon`. Il reste à compléter les « blancs ».
Si l'option en français n'est pas disponible, il faut sélectionner `If [Then] Else`, mettre la condition dans les parenthèses, les instructions si la condition est remplie dans les premières accolades et les instructions du sinon dans les deuxièmes accolades.
Les deux programmes possibles sont donc :
En français :

4. La compilation est la traduction du programme écrit dans le language de programmation en instructions que l'ordinateur peut exécuter.

```
Egalite(a,b):={
si a==b alors // le test d'égalité est ''==''
  afficher("ils sont égaux");
sinon
  afficher("ils sont différents");
fsi; // Fin du ''si''
}// Fin du programme
```

En anglais :

```
Egalite(a,b):={
if (a==b) { // début du ''oui''
  afficher("ils sont égaux")} // fin du ''oui''
else { // début du ''sinon''
  afficher("ils sont différents");
}// fin du ''sinon''
}// fin du programme
```

L'utilisation de boucles n'est pas plus difficile. Par exemple calculons la somme des entiers naturels inférieurs ou égaux à un un entier n saisi par l'utilisateur du programme.
Les instructions de boucles s'obtiennent en cliquant sur `Ajouter` puis sur `Loop`.
Pour demander une information à l'utilisateur, on utilise la commande `saisir`.
Enfin, dans ce programme, on utilise les variables n, k et S qu'on déclare comme variables *locales* c'est-à-dire qu'elle ne sont utilisées que dans le programme et sont « oubliées » après. C'est le rôle du `local S,n,k;`.
Voici le programme :

```
SommeEntiers():={
local S,n,k; // Déclaration des variables locales
saisir("Donner un entier naturel non nul :",n);
pour k de 0 jusque n faire // la boucle en français
S:=S+k;  // on augmente S de k
fpour;   // fin de la boucle
return S;// on renvoie la réponse.
}
```

On l'exécute en saisissant dans une nouvelle ligne : `SommeEntiers()`

4.3 Résumé

Mot clé ou exemple Xcas	Utilisation
// Commentaires	Tout ce qui suit (sur la ligne) les « // » n'est pas traité par Xcas
`x:=3;` `k:=k+1;`	Affectation de la valeur 3 à la variable x Augmentation de 1 de la variable k (incrémentation)

Mot clé ou exemple Xcas	Utilisation

`x:=saisir("entrer x : ");`
`afficher("la valeur de x :",x);`
`return a;`
	Saisie d'une variable
	Afficher
	renvoie la valeur a à l'utilisateur du programme

`si` <*test*> `alors` <*instructions*> `sinon` <*instructions*> `fsi;`	Test conditionnel ; le `sinon` est facultatif.
`pour` <*var*>`de` <*num*>`jusque` <*num*> `pas` <*num*>`faire` <*instructions*>`fpour;`	Boucle « pour » ; `pas` est facultatif : il s'agit de la valeur d'incrémentation de la variable.
`tantque` <*condition*>`faire` <*instructions*> `ftantque;`	Boucle « tant que ».
`repeter` <*instructions*> `jusqua` <*condition*>	Boucle ; les instructions sont répétées jusqu'à ce que la condition soit réalisée.

`continue;`
 	
`break;`
	Dans une boucle, passe à l'itération suivante.
	Pour quitter une boucle

Attention : l'utilisation de la lettre `i` est impossible comme nom de variable car elle est réservée pour le nombre imaginaire dont le carré vaut -1.
Remarque : on peut faire appel à un programme déjà compilé dans un autre programme.

5 Géométrie

En ouvrant une fenêtre de géométrie, on obtient une fenêtre ressemblant à la figure 5. La fenêtre est alors divisée en trois : d'abord, une partie « lignes de saisie » où on saisira des commandes, puis la fenêtre graphique et enfin, on reconnaît le bloc de boutons de déplacement de la fenêtre, de zoom, ... déjà vus dans la partie 3 de la page 7.

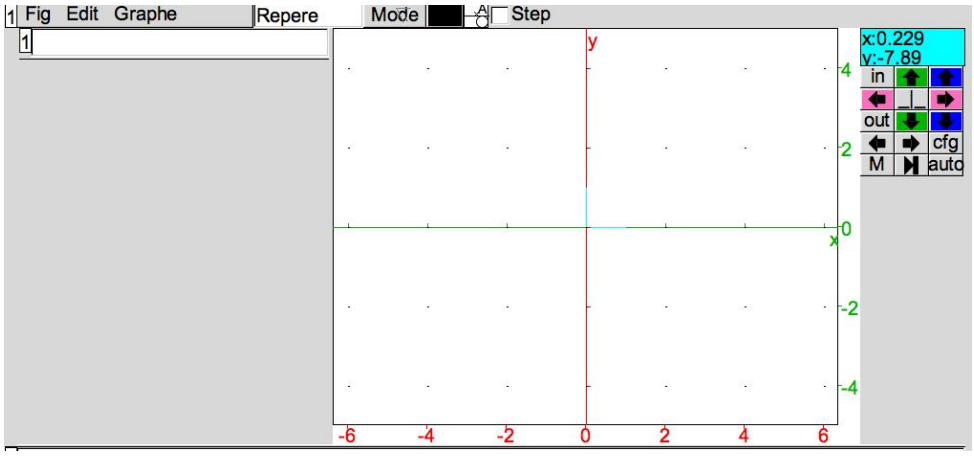

FIGURE 5 – Fenêtre géométrie

5.1 Équations cartésiennes, ...

Dans le menu `Fig` :

– on peut enregistrer toutes les lignes de commandes qui ont permis la création de la figure, pour pouvoir les insérer par la suite dans une autre figure ;
– on peut aussi exporter la figure dans divers formats (image, fichier LaTeX, ...).

Dans le menu `Edit` :
– on peut créer un paramètre (en donnant ses valeurs min et max ainsi que le pas) ; ce paramètre apparaît alors sous les boutons de zoom, déplacement, ... dans la partie droite de l'écran. On fait alors varier le paramètre en cliquant de chaque côté de sa valeur ;
– on peut activer la trace d'un objet variable : chacune de ses positions est marquée à l'écran (utile pour observer un lieu de points par exemple) ;
– on peut changer le format d'affichage (portrait / paysage) ;
– ...

Dans le menu `Graphe` :
– on peut tracer la représentation d'une fonction définie par son équation cartésienne, son équation en coordonnées polaires ou par des équations paramétriques ;
– on peut aussi visualiser l'aire sous une courbe ;
– on peut représenter les premiers termes d'une suite définie par une relation de récurrence du type $u_{n+1} = f(u_n)$;

5.2 Géométrie « classique »

Dans une fenêtre géométrie, on peut aussi construire des figures de géométrie « classique » en plaçant des points (par la donnée de leurs coordonnées ou en les positionnant à la souris), en traçant des segments, des droites (médiatrices, bissectrices, ...), des polygones, des cercles ou en représentant des vecteurs, ... On peut aussi créer des transformations (symétries, rotations, translations, ...) et construire des images par ces transformations. Toutes ces fonctions du logiciel ne seront pas détaillées ici car elles sont trop nombreuses et se trouvent assez « naturellement » en cliquant dans le menu `Geo` de la barre de menu principale.

5.3 Quelques commandes « en ligne »

On résume dans le tableau ci-dessous quelques commandes usuelles qu'on peut écrire sur une ligne de saisie :

Commande	Explication
`A:=point(2,1)`	crée et place le point A de coordonnées $(2;1)$
`I:=milieu(A,B)`	crée et place le milieu I de $[AB]$
`H:=orthocentre(A,B,C)`	crée et place l'orthocentre H de ABC
`G:=isobarycentre(A,B,C)`	crée et place l'isobarycentre(centre de gravité) G de ABC
`G:=barycentre([A,1],[B,2])`	crée et place le point G barycentre des points pondérés $(A,1)$ et $(B,2)$
`d:=droite(A,B)`	crée et trace la droite (AB)
`d:=mediatrice(A,B)`	crée et trace la médiatrice d de $[AB]$

Commande	Explication
`d:=bissectrice(A,B,C)`	crée et trace la bissectrice de $(\overrightarrow{AB}, \overrightarrow{AC})$ (ou $\widehat{BAC}$)
`d:=hauteur(A,B,C)`	crée et trace la hauteur issue de A du triangle ABC
`d:=mediane(A,B,C)`	crée et trace la médiane issue de A du triangle ABC
`d2:=parallele(A,d)`	crée et trace d_2 la parallèle à d passant par A
`d2:=perpendiculaire(A,d)`	crée et trace d_2 la perpendiculaire à d passant par A
...	...
`C:=cercle(A,2)`	crée et trace le cercle de centre A et de rayon 2
`C:=cercle(A,B)`	crée et trace le cercle de diamètre $[AB]$

Bien d'autres commandes existent, elles se trouvent facilement par le menu `Geo` ou par l'aide en choisissant l'index ou la recherche « par mots ».

6 Tortue

Pour ceux qui ne connaissent pas, la tortue graphique faisait partie du langage de programmation Logo créé à la fin des années 1960. Il s'agit en fait d'une « pointe de crayon » à qui on donne des ordres simples du style, avance de 10 unités, tourne de 45°à droite, change de couleur, ...
Un module tortue est présent dans Xcas, on y accède en cliquent sur le menu `Tortue`. On obtient alors une fenêtre qui ressemble à la figure 6 de la page 14.

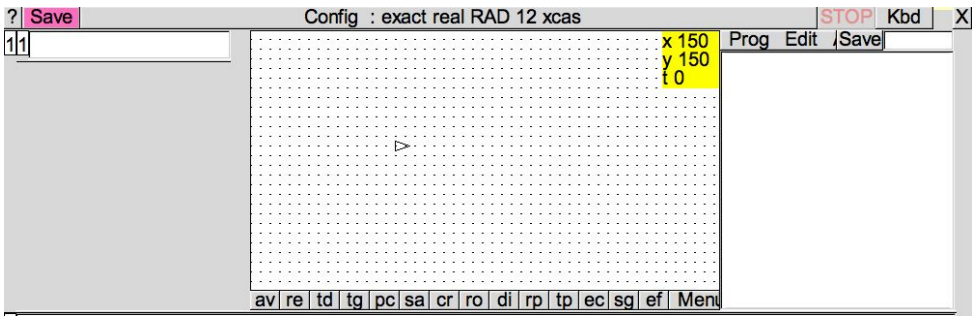

FIGURE 6 – Fenêtre du module tortue

On peut écrire dans les lignes de saisie (à gauche de la figure) des instructions simples par exemple `avance(30)` ou `tourne_droite(60)`, ... et la tortue exécute les ordres ; dans le même temps, les instructions sont recopiées dans la fenêtre de programmation (à droite). On peut alors compléter par des boucles, des tests, comme dans le module `Prog` (voir page 9).
L'ensemble des ordres possibles est accessible facilement sous la fenêtre graphique par un code à deux lettres `av` pour avance, `re` pour recule, ... (la description s'affiche en « info-bulle » lorsqu'on survole le bouton à l'aide de la souris).

Par exemple, pour tracer un carré un programme possible est le suivant (à saisir à droite) :

```
pour k de 1 jusque 4 faire
avance(30);
tourne_droite(90);
fpour;
```

On l'exécute en cliquant dans le menu `Edit` sur `Exec all`. Le carré se trace.

Autre exemple qui donne la figure 7 :

```
efface ();// Efface l'écran
etapes:=36;// nombre de cercles "intérieurs"
p:=20;// longueur d'un côté du polygone à étapes côtés
h:=0.5*p/(tan(pi/etapes));// rayon d'un cercle intérieur
pas_de_cote 10;// pour centrer la fig
saute 60;// idem
pour k de 1 jusque etapes faire// boucle
crayon irem(k,5);// choix de la couleur du crayon
tourne_gauche (360/etapes);// construction d'un côté
avance (p/2);// suite
rond (h);// trace un cercle
fpour;// fin de boucle
```

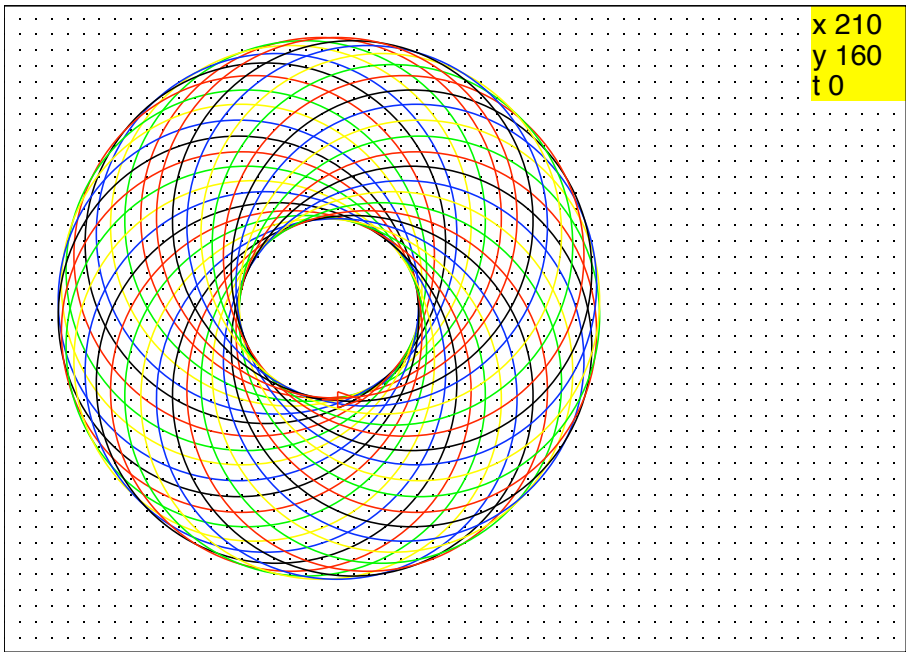

FIGURE 7 – Exemple de figure obtenue avec la tortue

Table des figures

Table des matières