

Fiche Xcas l'interface

1 Installation de Xcas

Le programme Xcas est un logiciel libre écrit en C++, (disponible sous licence GPL). La version à jour se récupère sur :
http://www-fourier.ujf-grenoble.fr/~parisse/giac_fr.html
 CAS (Computer Algebra System) veut dire calcul exact, formel ou symbolique.

2 Interface, objets.

Description de l'interface	
Fich Edit Cfg...	est une barre de menu cliquable
session1.xws ou Unnamed	est le nom de la session ou si la session n'a pas été sauvée
?	ouvre l'index de l'aide
Sauve	sauvegarde la session
Config : exact reel...	ouvre la configuration du CAS
STOP	interrompt un calcul trop long
Kbd	fait apparaître un clavier
X	ferme la session
1	est une ligne de commande

Vous pouvez taper votre première commande (cliquez si nécessaire dans cette ligne de commande pour y faire apparaître le curseur) : par exemple $1+1$, suivi de la touche "Entrée" ("Enter" ou "Return" selon les claviers). Le résultat apparaît au-dessous dans un éditeur d'expressions, et une nouvelle ligne numérotée 2, s'ouvre.

Xcas manipule différents types de données : les entiers (2), les fractions ($3/2$), les nombres flottants ($2.0, 1.5$), les paramètres formels (x, t), les variables ($a := 2$), les expressions ($x^2 - 1$), les fonctions ($f(x) := x^2 - 1$), les listes ($[1, 2, 3]$), les séquences (1, 2, 3), les chaînes de caractères ("na") et les objets géométriques.

Une expression est une combinaison de nombres et de variables reliés entre eux par des opérations alors qu'une fonction associe à une variable une expression. Par exemple $a := x^2 + 2x + 1$ définit une expression alors que $b(x) := x^2 + 2x + 1$ définit une fonction et on a $b(0) = \text{subst}(a, x=0) = 1$.

Une matrice est une liste de listes de même longueur, une séquence ne peut pas contenir de séquences.

Signification des signes de ponctuation	
.	sépare la partie entière de la partie décimale
,	sépare les éléments d'une liste ou d'une séquence
;	termine chaque instruction d'un programme
::	termine les instructions lorsqu'on ne veut pas l'affichage de la réponse
!	$n!$ est la factorielle de n ($4! = 1 \cdot 2 \cdot 3 \cdot 4 = 24$)
:=	$a := 2$ instruction d'affectation qui stocke 2 dans la variable a
[]	délimiteurs d'une liste ($L := [0, 2, 4]$ et $L[1]$ renvoie 2)
" "	délimiteurs d'une chaîne de caractères ($C := "ba"$ et $C[1]$ renvoie "a")

3 Les configurations

Configurer permet de définir les paramètres de votre environnement.

Les différentes configurations possibles	
Cfg►Configuration du CAS	ouvre la configuration du CAS
Cfg►Configuration graphique	ouvre la configuration graphique par défaut
Cfg►Configuration generale	ouvre la configuration générale
bouton cfg (Graphe)	ouvre la configuration du niveau graphique
bouton Config : exact...	ouvre la configuration du CAS
bouton Config tableur :	ouvre la configuration du tableur

Vous pouvez modifier l'aspect de l'interface et sauvegarder vos modifications pour les utilisations futures (menu Cfg).







4 Les niveaux

Chaque session est composée de niveaux numérotés qui peuvent être de différentes natures : ligne de commandes pour le calcul formel, écran de géométrie dynamique (2-d et 3-d), tableur formel, dessin tortue, éditeur de programmes etc...

Les différents niveaux possibles	
Alt+c	ouvre une ligne de commentaires
Alt+d	ouvre un niveau de dessin tortue
Alt+e	ouvre un éditeur d'expressions
Alt+g	ouvre un niveau de géométrie 2-d
Alt+h	ouvre un niveau de géométrie 3-d
Alt+n	ouvre une ligne de commandes
Alt+p	ouvre un éditeur de programmes
Alt+t	ouvre un tableur

5 Les aides

Toutes les commandes sont classées par ordre alphabétique dans l'index de l'aide (Aide►Index). Vous avez aussi plusieurs manuels disponibles avec des exercices corrigés (Aide►Manuels►...) et des exemples (Aide►Exemples).

Les aides possibles	
Aide►Index	ouvre l'index de l'aide des commandes
Aide►Manuels►...	ouvre un des manuels dans votre navigateur
	ouvre l'index de l'aide des commandes
ce 	ouvre l'index de l'aide sur ceil
ce 	ouvre l'index de l'aide sur ceil
ce 	ouvre l'index de l'aide sur ceil
?ceil	ouvre l'aide détaillée sur ceil
Cmds►Reel►Base►ceil	ouvre l'aide sur ceil dans msg visible avec
Cfg►Montrer►msg ou  ► 	

Fiche Xcas calcul formel de base

L'exécution d'une ligne de commandes se fait par la touche "Entrée". Les nombres peuvent être exacts ou approchés (flottants). Les nombres exacts sont les constantes prédéfinies, les entiers, les fractions d'entiers et toutes expressions ne contenant que des entiers et des constantes. Les nombres approchés sont notés avec la notation scientifique standard : partie entière suivie du point de séparation et partie fractionnaire optionnellement suivie de e et d'un exposant.

Opérations	Constantes prédéfinies
+ addition	pi $\pi \simeq 3.14159265359$
- soustraction	e $e \simeq 2.71828182846$
* mutiplication	i $i = \sqrt{-1}$
/ division	infinity ∞
^ puissance	+infinity ou inf $+\infty$
	-infinity ou -inf $-\infty$
	euler_gamma constante d'Euler

Séquences et listes ou vecteurs	
S:=a,b,c	S est une séquence de 3 éléments
S:=[a,b,c]	S est une liste de 3 éléments
S:=NULL	S est une séquence de 0 élément
S:=[]	S est une liste de 0 élément
dim(S)	renvoie le nombre d'éléments de S
S[0]	renvoie le premier élément de S
S[n]	renvoie le $n + 1$ unième élément de S
S[dim(S)-1]	renvoie le dernier élément de S
S:=S,d	ajoute l'élément d à la fin de la séquence S
S:=append(S,d)	ajoute l'élément d à la fin de la liste S

Chaînes de caractères	
S:="abc "	<i>S</i> est une chaîne de 3 caractères
S:=" "	S est une chaîne de 0 caractère
dim(<i>S</i>)	renvoie le nombre de caractères de <i>S</i>
S[0]	renvoie le premier caractère de <i>S</i>
S[<i>n</i>]	renvoie le <i>n</i> + 1 unième caractère de <i>S</i>
S[dim(<i>S</i>)-1]	renvoie le dernier caractère de <i>S</i>
S:=S+"d"	ajoute le caractère d à la fin de la chaîne <i>S</i>
"ab"+"def "	concaténe les deux chaînes et renvoie "abdef"

Fractions	
propfrac	partie entière + partie fractionnaire
getNum [numer]	numérateur de la fraction [simplifiée]
getDenom [denom]	dénominateur de la fraction [simplifiée]
f2nd	[numer, denom] de la fraction simplifiée
simp2	simplification d'un couple
dfc	développe en fraction continue un réel
dfc2f	transforme une fraction continue en réel

Fonctions classiques			
evalf(<i>t</i> , <i>n</i>)	évalue <i>t</i> avec <i>n</i> décimales	sign	signe (-1,0,+1)
max	maximum	min	minimum
round	arrondi	frac	partie fractionnaire
floor	plus grand entier \leq	ceil	plus petit entier \geq
re	partie réelle	im	partie imaginaire
abs	module ou valeur absolue	arg	argument
conj	conjugué	affixe	affixe
factorial	factorielle	binomial	coefficients binomiaux
exp	exponentielle	sqrt	racine carrée
ln log	logarithme naturel	log10	logarithme en base 10
sin	sinus	cos	cosinus
tan	tangente	cot	cotangente
asin	arc sinus	acos	arc cosinus
atan	arc tangente	acot	arc cotangente
sinh	sinus hyperbolique	cosh	cosinus hyperbolique
asinh	arc sinus hyperbolique	acosh	arc cosinus hyperbolique
tanh	tangente hyperbolique	atanh	arc tangente hyperbolique

Fonctions d'arithmétique	
a%p	<i>a</i> modulo <i>p</i>
powmod(<i>a</i> , <i>n</i> , <i>p</i>)	a^n modulo <i>p</i>
irem	reste de la division euclidienne
iquo	quotient de la division euclidienne
iquorem	quotient et reste
ifactor	décomposition en facteurs premiers
ifactors	liste des facteurs premiers
idivis	liste des diviseurs
gcd	plus grand diviseur commun
lcm	plus petit multiple commun
iegcd	identité de Bezout
iabcuv	renvoie [<i>u</i> , <i>v</i>] tels que $au + bv = c$
ichinrem	restes chinois
is_prime	teste si l'entier est premier
nextprime	prochain entier pseudo-premier
previousprime	entier pseudo-premier précédent

Transformations			
simplify	simplifie	tsimplify	simplifie (- puissant)
normal	forme normale	ratnormal	forme normale (- puissant)
expand	développe	partfrac	décomposition en éléments simples
factor	factorise	convert	transforme en le format spécifié

Transformations en trigonométrie			
tlin	linéarise	tcollect	linéarise et regroupe
texpand	développe les exp, ln et trig	trig2exp	trig vers exp
hyp2exp	hyperbolique vers exp	exp2trig	exp vers trig

Fiche Xcas les proba-stats et le tableur

Probabilités	
<code>comb(n,k)</code>	nombre de combinaisons de p objets pris parmi n
<code>binomial(n,k,[p])</code>	renvoie $\text{comb}(n,k) * p^k(1-p)^{n-k}$ ou <code>comb(n,k)</code>
<code>perm(n,p)</code>	nombre d'arrangements de p objets pris parmi n
<code>factorial(n), n!</code>	$n!$
<code>rand(n)</code>	entier aléatoire uniformément distribué dans $0..n-1$
<code>rand(p,q)</code>	réel aléatoire uniformément distribué dans $[p,q]$
<code>randnorm(mu,sigma)</code>	réel aléatoirement distribué selon la loi normale $N(\mu, \sigma)$

Statistiques 1-d	
moyenne	moyenne d'une liste pondérée par le deuxième argument
median	médiane d'une liste pondérée par le deuxième argument
quartiles	[min,quartile1, médiane,quartile3,max]
moustache	boite à moustache d'une série statistique
variance	variance d'une liste pondérée par le deuxième argument
ecart_type	écart-type d'une liste pondérée par le deuxième argument
histogramme	trace l'histogramme de l'argument

Statistiques 2-d	
<code>polygonplot</code>	ligne polygonale
<code>scatterplot</code>	nuage de points
<code>polygonscatterplot</code>	ligne polygonale pointée
<code>covariance</code>	covariance des éléments de l'argument
<code>correlation</code>	corrélation des éléments de l'argument
<code>exponential_regression</code>	(m,b) où $y = be^{mx}$ approche l'argument
<code>exponential_regression_plot</code>	graphe de $y = be^{mx}$ approchant l'argument
<code>linear_regression</code>	(a,b) où $y = ax + b$ approche l'argument
<code>linear_regression_plot</code>	graphe de $y = ax + b$ approchant l'argument
<code>logarithmic_regression</code>	(m,b) où $y = m \ln(x) + b$ approche l'argument
<code>logarithmic_regression_plot</code>	graphe de $y = m \ln(x) + b$, approchant l'argument
<code>polynomial_regression</code>	$(a_n, ..a_0)$ où $y = a_n x^n + ..a_0$ approche l'argument
<code>polynomial_regression_plot</code>	graphe de $y = a_n x^n + ..a_0$ approchant l'argument
<code>power_regression</code>	(m,b) où $y = bx^m$ approche l'argument
<code>power_regression_plot</code>	graphe de $y = bx^m$ approchant l'argument

Les commandes de statistiques s'utilisent :

- dans le tableur avec le menu Maths : le tableur se remplit automatiquement grâce à une boite de dialogue qui demande de préciser les paramètres de la commande choisie (le curseur doit être dans un niveau de type tableur qui s'obtient avec Alt+t).
- dans des lignes de commandes : soit on les tape, soit on les sélectionne dans le menu Cmds►Proba_stats, soit on utilise le menu Graphic►Stats et ses boites de dialogues.

Le tableur de Xcas est un tableur formel dans lequel on peut utiliser toutes les commandes de Xcas ainsi que les variables définies auparavant. Il a :

- ses menus Table Edit Maths (un click droit et on obtient ses menus),
- ses boutons eval val init 2-d 3-d,
- sa case de sélection qui est une case interactive : soit on sélectionne à la souris une cellule ou une plage et son nom s'y inscrit (par exemple A0 : B3), soit on y tape le nom des cellules à sélectionner, (par exemple A0 . . 3 , C),
- sa ligne de commande qui permet de remplir les cellules,
- un bouton pour configurer le tableur et qui rappelle cette configuration.

Au tableur est associé un écran graphique 2D visible si on a coché Graphe dans la configuration du tableur (visible en dessous du tableur ou à droite selon que Paysage est coché ou non). On peut aussi utiliser le bouton 2-d (ou 3-d).

Le tableur est une matrice dont on numérote les lignes par des nombres (0,1...) et les colonnes par une ou plusieurs lettres (A,B...). Le nom d'une cellule se fait par référence relative (C2) ou absolue (\$C\$2) ou mixte (\$C2 ou C\$2). Dans une cellule, on met soit une liste ou une formule de calcul commençant par =, soit une constante, soit une chaîne de caractères entourées de guillemets ("essai").

Exemple : Bézout et le tableur calculant $[u,v,\text{pgcd}(a,b)]$ avec $a*u+b*v=\text{pgcd}(a,b)$

- Dans une ligne de commande ou dans Init sheet de la configuration du tableur on met $a:=78$; $b:=6$, et on appuie sur init,
- La colonne A est "la suites des restes" r_n . On met a dans A0, b dans A1, $=\text{irem}(A0,A1)$ dans A2 et on remplit vers le bas (Ctrl+d),
- Les colonnes B et C, sont les deux suites u_n et v_n de façon qu'à chaque étape on ait : $r_n = u_n a + v_n b$. On met 1 dans B0, 0 dans C0, 0 dans B1, 1 dans C1, $B0-E2*B1$ dans B2, $C0-E2*C1$ dans C2 et on remplit vers le bas,
- La colonne D est $u_n a + v_n b$ ($D=A$). On met $=B0*\$A\$0+C0*\$A\1 dans D0 et on remplit vers le bas
- La colonne E est "la suites des quotients" q_n . On met $=\text{iquo}(A0,A1)$ dans E2 et on remplit vers le bas,
- La colonne F est la réponse $[u,v,\text{pgcd}(a,b)]$. On met $=\text{si } A0==0 \text{ alors } [B0,C0,D0] ; \text{sinon } 0 ; \text{fsi}$ dans F0 et on remplit vers le bas.

On vérifie avec $\text{iegcd}(78,56)$ et on obtient :

Fich	Edit	Cfg	Aide	CAS	Expression	Cmds	Prg	Graphic	Geo	Tableur	Phys	Scolaire	Tortue
Unnamed													
? Sauvi Config : exact real RAD 12 xcas 14.969M STOP Kbd X													
1	Table	Edit	Maths	eval	val	init	2-d	3-d	Save B.tab				
D0 = \$A\$0*B0+\$A\$1*C0													
Sheet config: * Spreadsheet B R40C10 auto down fill													
	A	B	C	D	E	F	G	H	I				
0	78	1	0	78	0	0	0	0	0				
1	56	0	1	56	0	0	0	0	0				
2	22	1	-1	22	1	0	0	0	0				
3	12	-2	3	12	2	0	0	0	0				
4	10	3	-4	10	1	0	0	0	0				
5	2	-5	7	2	1	[-5,7,2]	0	0	0				
6	0	28	-39	0	5	0	0	0	0				
7	2	-5	7	2	0	[-5,7,2]	0	0	0				
8	0	28	-39	0	0	0	0	0	0				
9	2	-5	7	2	0	[-5,7,2]	0	0	0				
10	0	28	-39	0	0	0	0	0	0				
11	2	-5	7	2	0	[-5,7,2]	0	0	0				
12	0	28	-39	0	0	0	0	0	0				
	0	1	2	3	4	5	6	7	8				
2 iegcd(78,56) [-5, 7, 2] M													

Fiche Xcas Algèbre

Polynômes	
normal	forme normale (développée et réduite)
expand	forme développée
ptayl	forme de Taylor
peval horner	évaluation en un point par l'algorithme de Horner
genpoly	polynôme défini par sa valeur en un point
canonical_form	trinôme sous forme canonique
coeff	coefficient ou liste des coefficients
poly2symb	du polynôme au format Xcas à la forme symbolique
symb2poly	de la forme symbolique à un polynôme au format Xcas
pcoeff	polynôme décrit par ses racines
degree	degré
lcoeff	coefficient du terme de plus haut degré
valuation	degré du monôme de plus bas degré
tcoeff	coefficient du monôme de plus bas degré
factor	décomposition en facteurs irréductibles
cfactor	décomposition en facteurs irréductibles sur \mathbb{C}
factors	liste des facteurs irréductibles
divis	liste des diviseurs
collect	factorisation sur le corps des coefficients
froot	racines avec leurs multiplicités
proot	valeurs approchées des racines
sturmab	nombre de racines dans un intervalle
getNum	numérateur d'une fraction rationnelle non simplifiée
getDenom	dénominateur d'une fraction rationnelle non simplifiée
propfrac	isole partie entière et fraction propre
partfrac	décomposition en éléments simples
quo	quotient de la division euclidienne
rem	reste de la division euclidienne
gcd	plus grand diviseur commun
lcm	plus petit multiple commun
egcd	identité de Bezout
chinrem	restes chinois
randpoly	polynôme aléatoire
cyclotomic	polynômes cyclotomiques
lagrange	polynômes de Lagrange
hermite	polynômes de Hermite
laguerre	polynômes de Laguerre
tchebyshev1	polynômes de Tchebyshev 1ère espèce
tchebyshev2	polynômes de Tchebyshev 2nde espèce

Matrices	
<code>M:=[[a,b,c],[f,g,h]]</code>	M est une matrice de 2 lignes et 3 colonnes
<code>dim(M)</code>	est la liste [nbre_lignes, nbre_colonnes]
<code>M[0]</code>	renvoie la première ligne de M
<code>M[n]</code>	renvoie la $n + 1$ unième ligne de M
<code>row(M,n)</code>	renvoie la $n + 1$ unième ligne de M
<code>col(M,n)</code>	renvoie la $n + 1$ unième colonne de M
<code>M[dim(M)[0]-1]</code>	renvoie la dernière ligne de M
<code>M[n..p]</code>	renvoie la sous matrice de M de lignes [n..p]
<code>M:=append(M,[d,k,l])</code>	ajoute la ligne [d,k,l] à la fin de M
<code>M[dim(M)[0]]:=[d,k,l]</code>	ajoute la ligne [d,k,l] à la fin de M
<code>M:=border(M,[d,k])</code>	ajoute la colonne [d,k] à la fin de M

Opérations sur les vecteurs et matrices	
<code>v*w</code>	produit scalaire
<code>cross(v,w)</code>	produit vectoriel
<code>A*B</code>	produit matriciel
<code>A.*B</code>	produit terme à terme
<code>1/A</code>	inverse
<code>tran</code>	transposée
<code>rank</code>	rang
<code>det</code>	déterminant
<code>ker</code>	base du noyau
<code>image</code>	base de l'image
<code>idn</code>	matrice identité
<code>ranm</code>	matrice à coefficients aléatoires

Systèmes linéaires	
<code>linsolve</code>	résolution d'un système
<code>rref</code>	réduction de Gauss-Jordan
<code>rank</code>	rang
<code>det</code>	déterminant du système

Réduction des matrices	
<code>jordan</code>	diagonalisation ou réduction de Jordan
<code>pcar</code>	coefficients du polynôme caractéristique
<code>pmin</code>	coefficients du polynôme minimal
<code>eigenvals</code>	valeurs propres
<code>eigenvects</code>	vecteurs propres

Fiche Xcas analyse

Dérivées	
<code>diff(E)</code> ou <code>E'</code>	expression de la dérivée de l'expression E par rapport à x
<code>diff(E,t)</code> ou <code>(E,t)'</code>	expression de la dérivée de l'expression E par rapport à t
<code>diff(f)</code> ou <code>f'</code>	fonction dérivée de la fonction f
<code>diff(E,x\$n,y\$m)</code>	expression de la dérivée partielle $\frac{\partial E}{\partial x^n \partial y^m}$ de l'expression E
<code>grad</code>	gradient
<code>divergence</code>	divergence
<code>curl</code>	rotationnel
<code>laplacian</code>	laplacien
<code>hessian</code>	matrice hessienne

Limites et développements limités	
<code>limite(E,x,a)</code>	limite en a d'une expression E
<code>limite(E,x,a,1)</code>	limite à droite en a de E
<code>limite(E,x,a,-1)</code>	limite à gauche en a de E
<code>taylor(E,a)</code>	développement limité de E en $x = a$ ordre 5
<code>series(E,x=a,n)</code>	développement limité de E en $x = a$ ordre n

Intégrales	
<code>int(E,x)</code>	primitive d'une expression E
<code>int(f)</code>	fonction primitive d'une fonction f
<code>int(E,x,a,b)</code>	intégrale exacte de E
<code>romberg(E,x,a,b)</code>	intégrale approchée de E

Équations	
<code>solve(eq,x)</code>	solution exacte dans \mathbb{R} d'une équation polynomiale
<code>solve([eq1,eq2],[x,y])</code>	solution exacte dans \mathbb{R} d'un système polynomial
<code>csolve(eq,x)</code>	solution exacte dans \mathbb{C} d'une équation polynomiale
<code>csolve([eq1,eq2],[x,y])</code>	solution exacte dans \mathbb{C} d'un système polynomial
<code>fsolve(eq,x=x0)</code>	solution approchée d'une équation ($x_0=x_{\text{estimé}}$)
<code>fsolve([eq],[var],[val])</code>	solution approchée d'un système ($val=x_{\text{estimé}}$)
<code>newton</code>	méthode de Newton
<code>linsolve</code>	système linéaire
<code>proot</code>	racines approchées d'un polynôme

Équations différentielles	
<code>desolve</code>	résolution exacte
<code>odesolve</code>	résolution approchée
<code>plotode</code>	tracé d'une solution
<code>plotfield</code>	tracé du champ de vecteurs
<code>interactive_plotode</code>	tracé de solutions définies à la souris

Tracés de courbes	
plot	graphe d'une expression d'une variable
tangente	tangente à une courbe
pente	pente d'une droite
plotfunc	graphe d'une expression d'1 ou 2 variable(s)
...,couleur=...	choisir la couleur d'un tracé
areaplot	affiche l'aire sous une courbe
plotparam	courbe paramétrique
plotpolar	courbe en polaires
plotimplicit(f(x,y),x,y)	courbe implicite de $f(x,y) = 0$

Un exemple Étude de la fonction f définie par : $f(x) = \frac{\ln(|2-x|)}{\ln(|x|)}$.

On va montrer que f est définie sur $\mathbb{R} - \{-1, 0, 1, 2\}$ et peut être prolongée sur $\mathbb{R} - \{-1, 2\}$. Puis on fera le graphe de f , on tracera les tangentes en $x = -1/2$, $x = 0$ et $x = 1$, on donnera une valeur approchée à 10^{-3} près de l'aire comprise entre $x = 3$, $x = 5$, $y = 0$ et la courbe avec une subdivision en 4 trapèzes.

On tape : `f(x):=ln(abs(x-2))/ln(abs(x))`

`limite(f(x),x,1)` renvoie -1 et `limite((f(x)+1)/(x-1),x,1)` renvoie -1, donc la pente de la tangente au point (1,-1) vaut -1.

`limite(f(x),x,0)` renvoie 0, `limite(f(x)/x,x,0,1)` renvoie $-\infty$ et `limite(f(x)/x,x,0,-1)` renvoie $+\infty$. Donc Oy est tangent en (0,0).

`limite(f(x),x,-1)` renvoie ∞ , donc $x = -1$ est asymptote.

`limite(f(x),x,2)` renvoie $-\infty$, donc $x = 2$ est asymptote.

`limite(f(x),x,inf)`, `limite(f(x),x,-inf)` renvoie (1,1), $y = 1$ est donc asymptote.

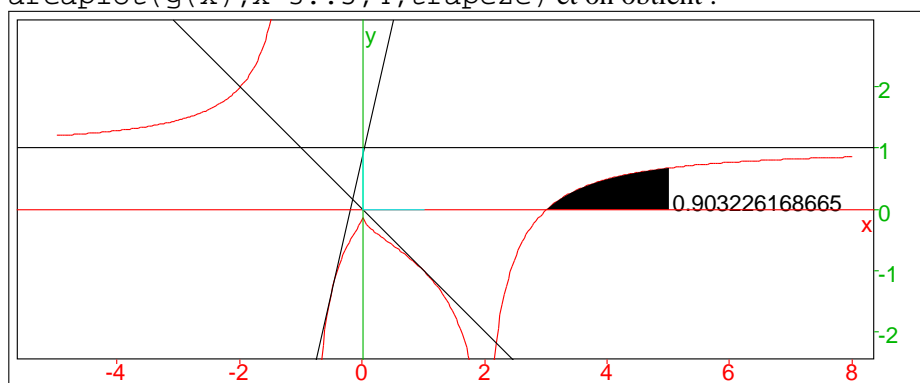
Pour prolonger f par continuité en $x = 0$ et $x = 1$, on tape :

`g:=when(x==0,0,when(x==1,-1,f(x)))`

On tape : `G:=plotfunc(g(x),x=-5..8,couleur=rouge),`

`droite(y=1),tangente(G,-1/2),droite(1-i,pente=-1),`

`areaplot(g(x),x=3..5,4,trapèze)` et on obtient :



Si on approche l'aire avec 4 trapèzes, on peut aussi taper : `Digits := 3 ; 0.5*(f(3)/2+f(3.5)+f(4)+f(5))` et cela renvoie 0.887.

Avec `areaplot(g(x),x=3..5)` l'aire est calculée par la méthode de Romberg et affichée avec 3 décimales. Pour en savoir plus, tapez `romberg(g(x),x,3,5)`, cela renvoie 0.903226168665 avec `Digits := 12 ;`. La méthode de Romberg est une accélération de la convergence de la méthode des trapèzes.

Fiche Xcas la géométrie

Fonctions de géométrie 2-d	
point	point donné par ses coordonnées ou son affixe
affichage=...	dernier argument de point pour l'afficher selon...
legend	met du texte à partir d'un point donné
segment	segment donné par 2 points
droite(A,B)	droite passant par A, B
droite(a*x+b*y+c=0)	droite d'équation $ax + by + c = 0$
triangle(A,B,C)	triangle de sommets A, B, C
bissectrice(A,B,C)	bissectrice issue de A du triangle ABC
angle(A,B,C)	mesure (en radians ou degrés) de \widehat{BAC}
mediane(A,B,C)	médiane issue de A du triangle ABC
hauteur(A,B,C)	hauteur issue de A du triangle ABC
mediatrice(A,B)	médiatrice de AB
carre(A,B)	carré direct de côté AB
cercle(A,r)	cercle de centre A , rayon r
cercle(A,B)	cercle de diamètre AB
rayon(c)	longueur du rayon du cercle c
centre(c)	le centre du cercle c
distance(A,B)	distance de A à B (point ou courbe)
inter(G1,G2)	liste des points de $G1 \cap G2$
inter_unique(G1,G2)	un des points de $G1 \cap G2$
assume	rajout d'un paramètre symbolique ou d'hypothèses
element	rajout d'un paramètre numérique
polygone	polygone fermé
polygone_ouvert	polygone ouvert
coordonees	coordonnées d'un point
equation	équation cartésienne
parameq	équation paramétrique
symetrie(A,M)	image de M par la symétrie point (ou droite) A
translation(B-A,M)	image de M par la translation \overrightarrow{AB}
rotation(A,t,M)	image de M par la rotation de centre A et d'angle t
homothetie(A,k,M)	image de M par l'homothétie de centre A et rapport k
similitude(A,k,t,M)	image de M par la similitude de centre A , rapport k et d'angle t

Pour construire un nouvel objet géométrique dans une figure :

- on sélectionne un à la souris puis on clique les objets définissant le nouvel objet. Si ☐ est décoché, les points cliqués sont à coordonnées exactes et si ☒ est coché, les points cliqués sont à coordonnées décimales.
- ou on remplit les lignes de commande, situées à gauche de la figure, en tapant la commande ou en s'aidant du menu Geo. L'exécution d'une ligne de commande par Entrée entraîne l'exécution des lignes suivantes.

En cochant , l'écran graphique est plus large et les lignes de commandes sont en dessous.

Par exemple, pour tracer un triangle ABC , la médiatrice de AB et le cercle circonscrit à ABC , on ouvre un niveau de géométrie 2d (Alt+g) et avec la souris :

- On choisit comme mode : Mode►Polygones►triangle
On clique pour désigner le premier point, puis on déplace la souris. Le segment AS où S est le point pointé par la souris, se dessine en pointillé. On clique pour désigner le 2-ième point, puis on déplace la souris un triangle en pointillé se dessine. On clique pour désigner le 3-ième point. On obtient un triangle et des lignes de commandes apparaissent ($A:=point(\dots), \dots$).
- On choisit comme mode : Mode►Lignes►mediatrice
On clique sur le point A , puis on déplace la souris : la médiatrice de AS , où S est le point pointé par la souris, se dessine en pointillé. On clique sur B . On obtient la médiatrice de AB et dans la ligne de commandes on a :
 $E:=mediatrice(A,B, 'affichage'=0)$
- On choisit comme mode : Mode►Cercles►circonscrit
On clique successivement sur les points A, B : le cercle circonscrit à ABS , où S est le point pointé par la souris, se dessine en pointillé. On clique sur C . On obtient le cercle circonscrit à ABC et dans la ligne de commandes on a :
 $F:=circonscrit(A,B,C, 'affichage'=0)$
- On choisit Mode►Pointeur pour pouvoir déplacer le point A, B ou C .

On peut aussi taper directement les commandes dans les lignes de commandes :

```
A:=point(-1,2);
B:=point(1,0);
C:=point(-3,-2);
D:=triangle(A,B,C);
E:=mediatrice(A,B);
F:=circonscrit(A,B,C);
```

Objets graphiques 3-d	
plotfunc	surface par équation
plotparam	surface ou courbe paramétrique
point	point donné par la liste de ses 3 coordonnées
droite	droite donnée par 2 équations ou 2 points
inter	intersection
plan	plan donné par 1 équation ou 3 points
sphere	sphère donnée par centre et rayon
cone	cône donné par sommet, axe, demi-angle d'ouverture
cylindre	cylindre donné par axe, rayon, [hauteur]
polyedre	polyèdre
tetraedre	tétraèdre régulier direct ou pyramide
tetraedre_centre	tétraèdre régulier direct
cube	cube
cube_centre	cube
parallelepiped	parallélépipède
octaedre	octaèdre
dodecaedre	dodécaèdre
icosaedre	icosaèdre

Fiche Xcas la programmation

1. Pour écrire une fonction ou un programme

Il faut :

- choisir une syntaxe, on décrit ici la syntaxe Xcas,
 - soit avec le menu `Cfg►Mode(syntax)►xcas`,
 - soit en ouvrant la fenêtre de configuration du CAS en appuyant sur le bouton `Config : ...` et choisir Xcas dans `Prog style`,
- ouvrir un niveau éditeur de programme soit en tapant `Alt+p`, soit avec le menu `Prg►Nouveau programme`. Il contient déjà le `;` qui doit terminer le programme.
- taper la fonction en terminant chaque instruction par `;`. Le nom de cette fonction et de ses arguments ne doit pas déjà être un mot-clef ou nom de commande Xcas, ceux-ci apparaissent en bleu et brun dans l'éditeur de programmes. On peut commencer le nom des fonctions par une Majuscule pour diminuer le risque.
- cliquer sur OK ou appuyer sur F9, pour compiler le programme.
- pour exécuter le programme, il suffit de se placer dans une ligne de commandes vide, taper le nom du programme suivi entre parenthèses par les valeurs des paramètres séparées par des virgules.

2. Le menu Ajouter d'un niveau éditeur de programme

Ce menu vous permet d'avoir la syntaxe d'une fonction, d'un test et des boucles.

Euclide et l'identité de Bézout :

Syntaxe d'une fonction :

```
f(x,y):={
  local z,a,...,val;
  instruction1;
  instruction2;
  val:=...;
  .....
  instructionk;
  retourne val;
};;
```

```
Bezout(a,b):={
  local la,lb,lr,q;
  la:=[1,0,a];
  lb:=[0,1,b];
  tantque b!=0 faire
    q:=iquo(la[2],b)
    lr:=la+(-q)*lb;
    la:=lb;
    lb:=lr;
    b:=lb[2];
  ftantque
  retourne la;
};;
```

3. Compilation La réponse à une compilation réussie sera Done si on met `;` à la fin ou sera la traduction du programme si on met `;` à la fin.

Pour `Bezout(a,b)`, on clique sur OK (ou touche F9) et on obtient `// Parsing Bezout // Success compiling Bezout puis Done`. Puis, on tape : `Bezout(78,56)` et on obtient `[-5,7,2]` (car $-5*78+7*56=2=\text{pgcd}(78,56)$).

4. Pas à pas Vous pouvez exécuter un programme commandes par commandes ou le mettre au point grâce au débogueur. Pour cela on tape :

```
debug(Bezout(78,56))
```

Une fenêtre s'ouvre et on appuie sur `sst` pour une exécution au pas à pas.

Instructions en français	
affectation	<code>a:=2;(a prend la valeur 2) et purge (a);(a redevient formelle)</code>
entrée expression	<code>saisir("a=",a);</code>
entrée chaîne	<code>saisir_chaine("a=",a);</code>
sortie	<code>afficher("a=",a);</code>
valeur retournée	<code>retourne a;</code>
arrêt	<code>break;</code>
alternative	<code>si <condition> alors <inst> fsi;</code> <code>si <condition> alors <inst1> sinon <inst2> fsi;</code>
boucle pour	<code>pour j de a jusque b faire <inst> fpour;</code> <code>pour j de a jusque b pas p faire <inst> fpour;</code>
boucle répéter	<code>repeter <inst> jusqu'a <condition>;</code>
boucle tantque	<code>tantque <condition> faire <inst> ftantque;</code>
boucle faire	<code>faire <inst1> si(<condition>)break;<inst2>ffaire;</code>

Instructions comme en C++	
affectation	<code>a:=2;(a prend la valeur 2) et purge (a);(a redevient formelle)</code>
entrée expression	<code>input("a=",a);</code>
entrée chaîne	<code>textinput("a=",a);</code>
sortie	<code>print("a=",a);</code>
valeur retournée	<code>return(a);</code>
arrêt	<code>break;</code>
alternative	<code>if (<condition>) {<inst>;}</code> <code>if (<condition>) {<inst1>} else {<inst2>;}</code>
boucle pour	<code>for (j:= a;j<=b;j++) {<inst>;}</code> <code>for (j:= a;j<=b;j:=j+p) {<inst>;}</code>
boucle répéter	<code>repeat <inst> until <condition>;</code>
boucle tantque	<code>while (<condition>) {<inst>;}</code>
boucle faire	<code>do <inst1> if (<condition>) break;<inst2> od;</code>

Signification des signes de ponctuation	
.	sépare la partie entière de la partie décimale
,	sépare les éléments d'une liste ou d'une séquence
;	termine chaque instruction d'un programme
: ;	termine les instructions lorsqu'on ne veut pas l'affichage de la réponse

Opérateurs			
+	addition	-	soustraction
*	mutiplication	/	division
^	puissance	a mod p	a modulo p
==	teste l'égalité	!=	teste la différence
<	teste la stricte infériorité	<=	teste l'infériorité ou l'égalité
>	teste la stricte supériorité	>=	teste la supériorité ou l'égalité
, ou	opérateur booléen infixé	&& , et	opérateur booléen infixé
non	inverse logique de l'argument	!(. .)	inverse logique de l'argument
vrai	est le booléen vrai ou true ou 1	faux	est le booléen faux ou false ou 0
=	permet de définir une équation	n!	n ! est la factorielle de n

Fiche Xcas la tortue

Déplacements	
efface	pour effacer
avance	pour avancer
recule	pour reculer
saute	pour avancer ou reculer sans tracer
pas_de_cote	pour faire des pas de coté à gauche (ou à droite) sans tracer
tourne_gauche	pour tourner à gauche
tourne_droite	pour tourner à droite

Les couleurs et la tortue	
crayon	pour connaître ou changer la couleur du crayon
cache_tortue	cache la tortue
montre_tortue	montre la tortue
dessine_tortue(n)	dessine la tortue en une forme pleine (n=0) ou non (n=1)

Les formes	
rond	dessine un cercle ou un arc de cercle
triangle_plein	dessine un triangle plein
rectangle_plein	dessine un carré, rectangle, losange ou parallélogramme plein
disque	dessine un disque ou un secteur angulaire tangent à la tortue
disque_centre	dessine un disque ou un secteur angulaire de centre la tortue
polygone_rempli	remplit le polygone dessiné juste avant

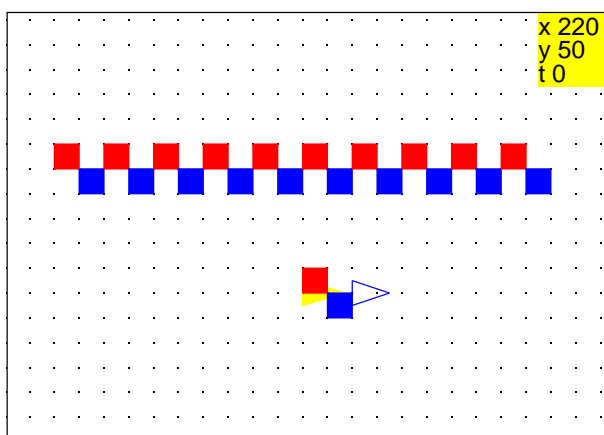
Les légendes	
ecris	pour écrire là où se trouve la tortue ou en un point
signe	pour signer, en bas à gauche, son dessin

La programmation et la tortue	
si <c> alors <inst> fsi	fait <inst> si la condition <c> est vraie
si <c> alors <inst1> sinon <inst2> fsi	fait <inst1> si la condition <c> est vraie ou fait <inst2> si <c> est fausse
repete n,<i1>,<i2>	repète n fois les instructions <i1>,<i2>
pour j de j1 jusqu'à j2 faire <inst> fpour	fait <inst> en itérant la variable j avec un pas=1
pour j de j1 jusqu'à j2 pas p faire <inst> fpour	fait <inst> en itérant la variable j avec un pas = p
tantque <c> faire <inst> ftantque	fait <inst> tant que la condition <c> est vraie
retourne <obj>	définit la valeur de la fonction par <obj>
lis(a)	met dans a, une expression lue au clavier
lis_phrase(a)	met dans a, une phrase lue au clavier
sauve("toto",a,b)	sauve dans le fichier toto les fonctions a,b
ramene("toto")	ramène les fonctions du fichier toto

Positionnement	
position	pour avoir la position de la tortue ou pour changer sa position
cap	pour avoir le cap de la tortue ou pour changer son cap
vers	pour diriger la tortue selon un point

Il ne doit y avoir qu'un seul écran de dessin tortue par session.

Pour piloter la tortue, on met des commandes à gauche du dessin : soit on les tape en toutes lettres, soit on les sélectionne dans le menu *Tortue*, soit on clique sur leurs abréviations situées sur la barre des boutons (le bouton *cr* affiche en plus la palette des couleurs du crayon), soit on réutilise une commande déjà tapée. À droite de l'écran se trouve l'enregistreur qui enregistre toutes les commandes utilisées : en cas d'erreur, il est facile de le modifier, puis d'exécuter toutes les commandes depuis le début en appuyant sur F7.



Cette frise est la répétition d'un même motif, isolé ci-dessus avec en jaune la position de départ de la tortue. On réalise d'abord le dessin du motif : on ouvre un niveau de dessin tortue (*Alt+d*) et on met dans les lignes de commandes :

```
crayon 1;
rectangle_plein ;
saute ;
tourne_droite ;
crayon 4;
rectangle_plein ;
tourne_gauche ;
saute ;
```

Pour cela on appuie successivement sur les boutons *cr*, *rp*, *sa*, *td*, Ces commandes se mettent automatiquement à droite dans l'enregistreur (une erreur se corrige directement dans l'enregistreur et on réexécute celui-ci avec F7).

On ouvre un éditeur de programme (*Alt+p*) et d'un coup de souris on recopie les commandes de l'enregistreur dans l'éditeur : on remplace alors *efface;* par *motif():=*{, puis on rajoute *}* avant *;;* et on appuie sur F9.

On tape ensuite à gauche, dans les lignes de commandes : *repete 10,motif()* Vous pouvez zoomer le dessin, en avant ou en arrière, avec la molette de la souris et déplacer la fenêtre visible en la translatant avec la souris.

Cet exemple vous montre comment on réalise un dessin complexe en le décomposant et aussi comment on utilise l'enregistreur pour écrire facilement une procédure à partir d'un dessin exécuté en pas à pas.