

Le calcul formel au lycée avec Xcas

1. DE QUOI S'AGIT-IL ?

Xcas est un logiciel de calcul **formel**, c'est à dire qu'il est capable de faire du calcul littéral en plus du calcul numérique. On peut donc résoudre des équations, même celles ayant des coefficients qui ne sont pas numériques, on peut factoriser et développer des expressions, calculer une dérivée ou une primitive d'une fonction. On peut travailler sur des vecteurs, des matrices, des suites... Et faire bien d'autres choses très complexes, mais dans cette introduction à l'utilisation de ce logiciel, on s'en tiendra à certains aspects du programme de lycée général.

Xcas est bien plus qu'un logiciel de calcul formel : il sait faire aussi de la géométrie dynamique (comme GeoGebra), il intègre plusieurs langages de programmation, un tableur un peu spécial... Bref, c'est selon son auteur un « couteau suisse » des maths, très puissant et d'une grande homogénéité entre ses différents modules ; c'est la cause d'une complexité certaine, et qui fait que dans ce document, on n'abordera qu'**une seule des facettes de ce logiciel**. Et là, on verra qu'il est très **abordable** et **simple d'emploi**.

Ce **petit mode d'emploi** est donc **volontairement très réducteur** : quand vous serez familiarisés avec le module de calcul formel, je vous engage à explorer tout le reste !

A quoi sert un logiciel de calcul formel au lycée ?

- Ce n'est absolument pas l'outil miracle qui fait le devoir à la place de l'élève, loin de là, d'autant plus que parfois, il faut arranger le résultat donné par la machine (donc comprendre ce qu'on fait), et que dans tous les cas, dans un devoir, ce n'est pas tant le résultat qui compte que la manière d'y arriver, la démarche, qui doit être explicitée dans la copie ; et ça, Xcas ne vous le fera pas.
- C'est un merveilleux outil de conjecture, d'investigation et de réflexion : quand on flaire une piste dans un raisonnement, si cette piste demande beaucoup de calculs pour être explorée, on peut la valider (ou non) très vite et éviter de perdre du temps inutilement.
- Et aussi, pour les élèves désireux de faire des exercices supplémentaires, c'est un super outil d'autocorrection.

2. PREMIERS CONTACTS

2.1. POUR QUELS ORDINATEURS ?

Tous ! Xcas existe sur les plateformes Windows, Linux et MacOSX. C'est un programme relativement léger qui tourne bien sur toutes les machines, même sur les petites configurations (par exemple sur mon eeepc 701 : opérationnel en 2s après clic).

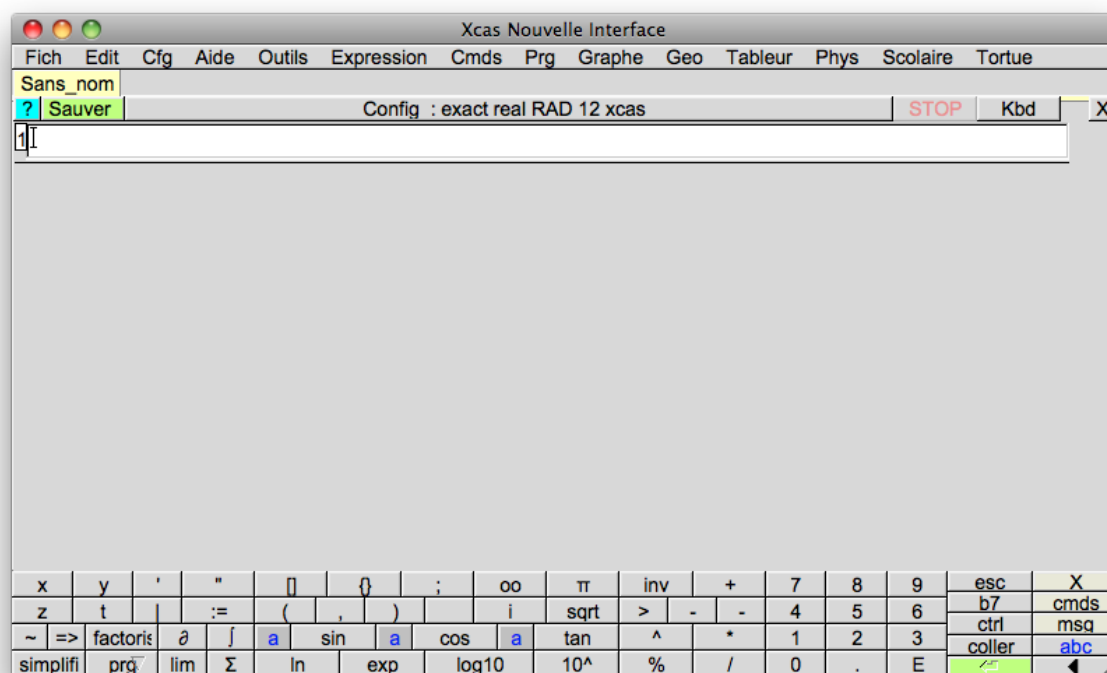
On trouvera le logiciel et une abondante documentation à l'adresse suivante :

http://www-fourier.ujf-grenoble.fr/~parisse/giac_fr.html

2.2. INSTALLATION, CONFIGURATION

Le logiciel s'installe normalement sans problèmes. Quand on le lance pour la première fois, il demande le choix de la configuration (Maple ou Xcas) : on choisit Xcas et c'est tout.

On obtient alors la fenêtre de travail :



On peut commencer par escamoter le clavier du bas en cliquant sur le bouton « Kbd » en haut à droite sous les menus : ce clavier ne nous servira pas vraiment (enfin, c'est ma façon de faire, si vous voulez l'utiliser, libre à vous). Ensuite, vous pouvez redimensionner et déplacer la fenêtre sur votre écran : si vous allez dans le menu Config → Sauver préférences, ces réglages seront conservés lors d'une utilisation ultérieure : très pratique pour se retrouver « chez soi ».

Pour la configuration, c'est simple : le logiciel est bien configuré d'origine pour l'usage qu'on en fera, on ne touchera donc à rien.

A part ça, on remarque une interface assez dépouillée : les menus en haut, et le champ de saisie d'une expression, numéroté [1] et en blanc : on y tapera simplement nos commandes.

2.3. PRINCIPE D'UTILISATION

Pour démarrer, il suffit de connaître deux choses de base : l'**affectation** d'une valeur ou d'une expression à une variable ou à une fonction, et **utiliser l'aide** en conjonction avec vos connaissances de base en maths.

L'**affectation** (ou **assignation**) se fera toujours de la même manière, en utilisant « := »

Ensuite, en fonction de l'opération qu'on voudra faire (développer, factoriser, simplifier, dériver, intégrer, calculer une limite...), on commencera à taper le nom de l'opération désirée, et on appuiera sur la touche « → » du clavier (tabulation) pour voir apparaître la fenêtre d'aide.

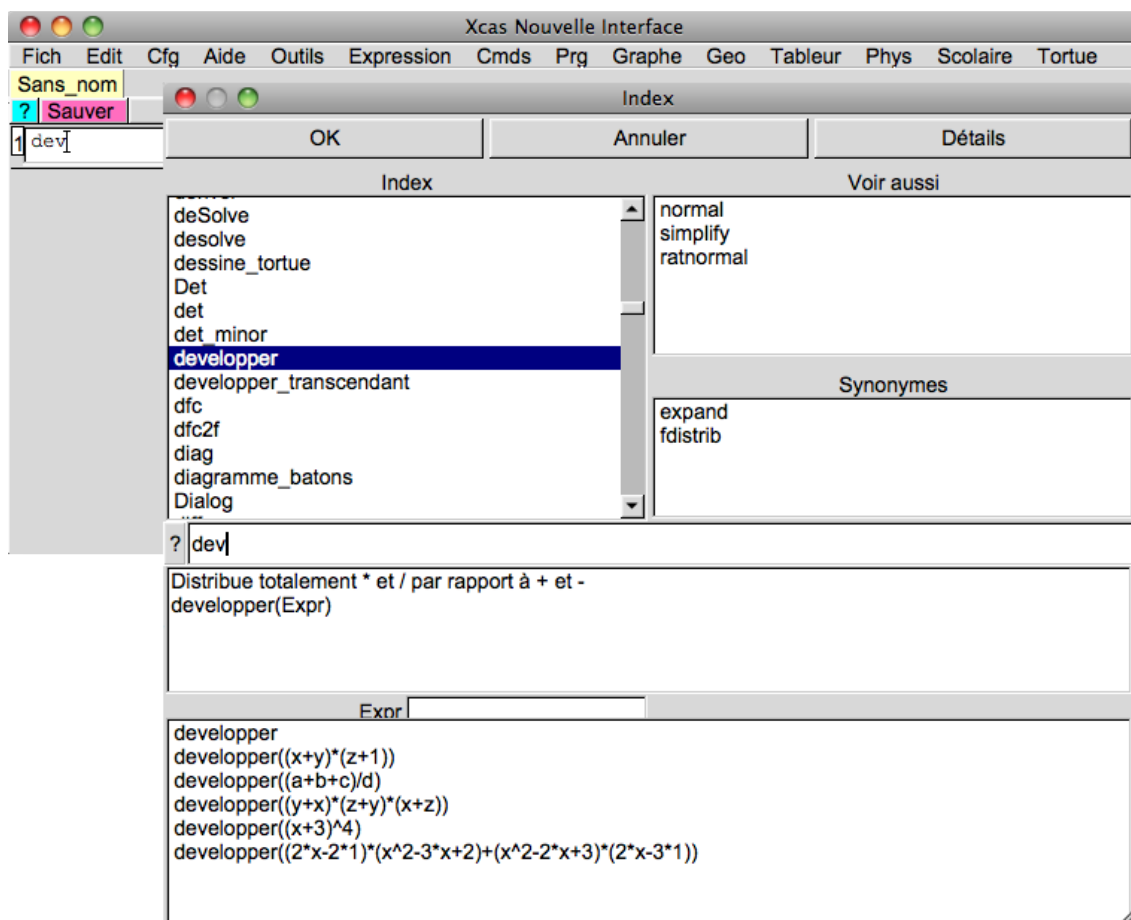
Cette fenêtre est le **dictionnaire** de Xcas (voir illustration page suivante). En haut à gauche, dans la case « index », vous avez la liste de toutes les commandes Xcas, par ordre alphabétique ; sur la ligne du centre, vous pouvez taper les premières lettres d'une commande, et l'index se positionne en conséquence ; la case en dessous vous donne une explication de la fonctionnalité de la commande, et celle encore en dessous donne des exemples d'application. A droite, vous avez les synonymes : Xcas est très tolérant sur la syntaxe, il parle Français ou Anglais (on peut utiliser indifféremment « solve » ou « résoudre »). Au dessus, vous avez une case qui indique des commandes en lien avec celle que vous étudiez.

Vous noterez qu'**aucune commande** ne comporte de **caractère accentué**, même celles exprimées en Français, et l'aide ne les prend pas en compte : on tapera donc « deriver » et non pas « dériver ».

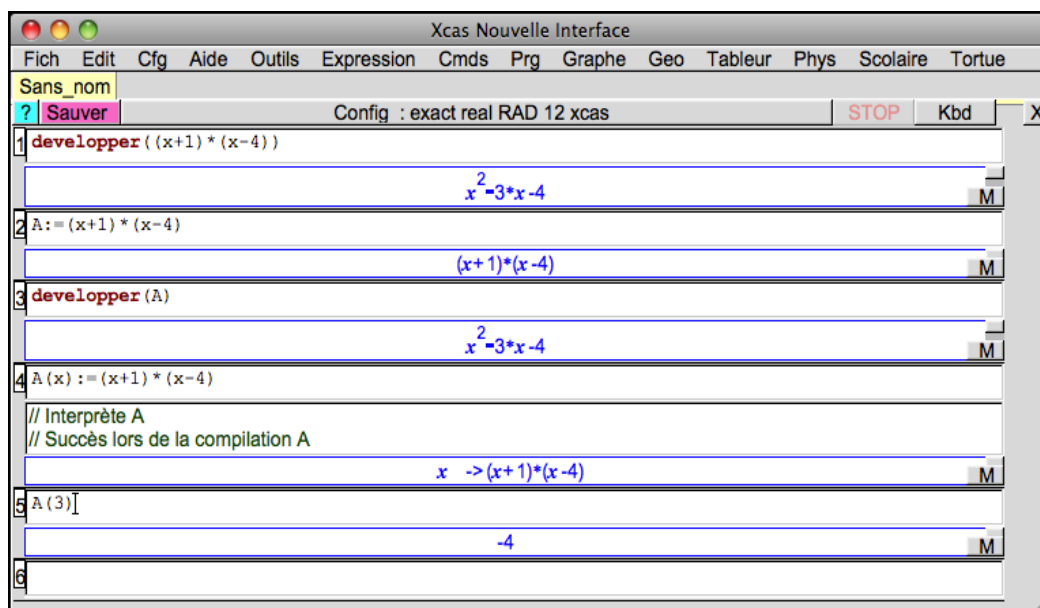
Prenez bien le temps de lire les paragraphes ci-dessus, et d'explorer cette fenêtre d'aide. On comprendra qu'il suffit de taper les premières lettres présumées d'une commande pour obtenir facilement via l'aide la syntaxe exacte de l'action qu'on désire exécuter.

Maintenant, place à l'exemple : votre petit frère (ou sœur) est en 4^e, vous devez vérifier si ses exercices de développement sont bons, et vous avez la flemme de les faire à la main...

Voyons ce que ça donne avec Xcas :



On commence à taper « dev », on appuie sur « → » et on obtient l'écran ci-dessus. Pour ne pas se fatiguer à taper la commande en entier, on la sélectionne dans la fenêtre « index », et on va simplement cliquer sur le bouton « OK » en haut à gauche. La fenêtre d'aide disparaît et la commande a été complétée dans la ligne [1] ; il suffit de rentrer notre expression et de valider la commande en appuyant sur la touche « entrée » du clavier.



Ci-dessus, on peut juger du résultat, qui apparaît sous la commande, et voir aussi diverses manières d'aborder notre expression :

- on peut directement développer notre expression comme en [1]
- on peut d'abord donner l'expression et ensuite la développer ([2] et [3]) : cette manière de faire nous permettra d'effectuer d'autres opérations sur la variable A qui contient notre expression
- on peut aussi exprimer A sous forme d'une fonction A(x) comme en [4], ça sera utile si par exemple, on demande d'évaluer A pour $x = 3$ ([5]), ce qui ne serait pas faisable aussi simplement avec la définition [2] de A.

Il est donc important avant de commencer un calcul, de voir l'exercice dans sa globalité afin de choisir le paramétrage optimum correspondant à notre problème. Rien de nouveau par rapport à une résolution « à la main » en fait : il faut être rigoureux et procéder avec ordre et méthode. Qui a dit que les logiciels de maths étaient des outils pour les ânes et qu'ils s'occupaient de tout à la place de l'élève ?

On aura noté que quand on valide une commande avec la touche « entrée », une autre ligne de commande blanche apparaît en dessous, prête pour l'ordre suivant.

Si on se trompe dans la dernière commande (on obtient un message d'erreur, parfois en Anglais...), il suffit de la modifier et de la revalider avec la touche « entrée ».

Si une commande ne sert à rien, on peut l'effacer : il faut cliquer sur son n° à gauche, il devient noir, et on appuie sur la touche d'effacement « ← ».

Si on a oublié une commande au beau milieu de la feuille, il faut cliquer dans la ligne de commande située en dessous de celle qu'on veut rajouter, et on sélectionne le menu « outils → nouvelle entrée ». Une ligne de commande blanche apparaît au dessus.

Si on modifie une commande située au milieu de la feuille de calcul (la session Xcas), ou si on en rajoute une, il faut recalculer toute la feuille à l'aide du menu « edit → exécuter session » ou de la combinaison de touches « CTRL + F9 » (« Fn + CTRL+F9 » sur les portables Mac).

Pour sauvegarder son travail, il faut cliquer sur le bouton « sauver », situé en haut à gauche sous les menus.

A propos des menus, on notera qu'ils n'ont (presque) servi à rien ! On peut bien sûr les utiliser, mais ils sont touffus, redondants (on aura la même commande dans plusieurs menus différents), et il n'y a pas d'aide pour expliciter les commandes. Pour débiter, mieux vaut les oublier. On pourra éventuellement utiliser le menu « aide », qui donne accès à des documents très intéressants, mais qui peuvent être complexes pour des lycéens, et aussi le menu « scolaire », qui regroupe les commandes essentielles qu'on verra ici.

Voilà le principe général de fonctionnement : finalement, c'est simple, non ?

3. EXEMPLES

3.1. CALCULS

2	$2 - (3 - 5) * (8 - 12)$		6	49^{23}	
	-6			749048330965186233494494102694564493649	
3	$1/3 + 1/4$		7	pi	
	$\frac{7}{12}$			π	
4	$264/418$		8	evalf(pi)	
	$\frac{12}{19}$			3.14159265359	
5	sqrt(48)		9	evalf(pi, 38)	
	$(\sqrt{3}) * 4$			3.1415926535897932384626433832795028842	

Presque pareil que ma Casio 92 collègue ! On peut donc se servir de Xcas comme d'une calculatrice. Il faut bien évidemment **respecter les priorités d'opérations** et écrire le calcul correctement, comme sur une calculatrice d'ailleurs. Xcas simplifie les fractions, et les racines.

Par contre, Xcas est un peu plus puissant qu'une calculatrice du commerce, même très perfectionnée. On peut le voir en [6] (nota : le « ^ » signifie « puissance » ; ici, on calcule 49 puissance 23) : il y a quelques chiffres dans le résultat (un peu plus que 10 !). De même, Xcas utilise « pi » comme valeur exacte, il n'utilise pas de valeur approchée ; mais on peut en obtenir une avec la commande « evalf » (en [8] et [9], et dans cette dernière, on spécifie le nombre de chiffres significatifs désirés, ici 38).

Il y a bien d'autres subtilités qu'on n'abordera pas ici : les courageux iront voir la doc.

3.2. TRINÔME DU SECOND DEGRÉ (2^{NDE} ET 1^{ES} ET ES)

On a ici un problème classique de 1^e (voire de 2^{nde}) : on pourra mettre le trinôme sous forme canonique, le factoriser, rechercher ses racines, et on pourra aussi étudier son signe grâce à une inéquation.

La syntaxe des commandes est très simple et intuitive : comme on a vu au §2.3, il suffit de commencer à taper une commande présumée, d'appeler l'aide avec la touche de tabulation « → », et de choisir la commande adéquate dans l'index.

On aura donc les commandes « **forme_canonique** », « **factor** » ou son synonyme « **factoriser** », et « **solve** » ou son synonyme « **resoudre** » pour faire tout ce travail.

```

1 p(x) := 3x^2 - 2x - 4
// Interprète p
// Succès lors de la compilation p
x -> 3*x^2 - 2*x - 4

2 forme_canonique(p(x))
3*(x - 1/3)^2 - 13/3

3 factor(p(x))
(x + (sqrt(13)-1)/3)*(3*x - (sqrt(13)-1))

4 solve(p(x)=0, x)
[-(sqrt(13)+1)/3, (sqrt(13)+1)/3]

5 solve(p(x)>0, x)
[-(sqrt(13)+1)/3 > x, x > (sqrt(13)+1)/3]

6 f(x) := (2x-3)(5x+1)
Attention : utilisation de multiplication implicite pour (2*x-3)(5*x+1)
// Interprète f
// Succès lors de la compilation f
x -> (2*x-3)*(5*x+1)

7 developper(f(x))
10*x^2 - 13*x - 3

```

On notera en [3] que la forme factorisée n'est pas tout à fait la même que celle donnée dans les manuels de 1^e : il aurait fallu factoriser par 3 le 2^e facteur pour retrouver la 2^e racine. L'analyse du résultat demande un peu de réflexion.

On note pour les commandes « **solve** », qu'on rajoute un « , x » après l'égalité ou l'inégalité à résoudre : ça signifie qu'on résout une (in)égalité d'inconnue x.

La solution de l'inéquation [5] nous permettra de vérifier le tableau de signes du trinôme. La solution est fractionnée en deux intervalles, la virgule a le sens de la réunion de ces intervalles. Il y a des cas où la réponse présente en plus des « and » (le « et » anglais) : il faut le voir comme l'intersection d'intervalles.

Si on rentre un trinôme sous forme factorisée, on peut bien entendu le développer : la commande « **developper** » se fera une joie d'exécuter ce travail pour nous ([7]).

On note en [6] le message d'erreur lors de la définition de la fonction f : on n'a pas mis de symbole multiplier « * » entre les deux expressions entre parenthèses : Xcas le signale, mais finalement, il accepte la syntaxe. Celle-ci est donc très souple comparée à celle de logiciels concurrents, où **tous** les symboles « * » doivent apparaître, par ex : $2x^2-3(2x-5)$. Avec Xcas, on écrit $2x^2-3(2x-5)$.

Il ne faudra pas oublier ceci si d'aventure on utilise WxMaxima ou un autre logiciel de calcul formel. Xcas s'est assoupli au fil des versions, et c'est tant mieux !

3.3.ÉTUDE DE FONCTION (1^{ES} ET ES)

L'étude « sur papier » d'une fonction se fait pas à pas :

- on détermine l'ensemble de définition de f (s'il n'est pas donné ou imposé)
- on calcule la dérivée f'
- on recherche les valeurs qui annulent la dérivée : $f'(x)=0$
- on détermine le signe de la dérivée : $f'(x)>0$
- on dresse le tableau de variations

Avec Xcas, on fera exactement la même chose. Si on veut juste vérifier le tableau de variations, on verra une procédure « accélérée ».

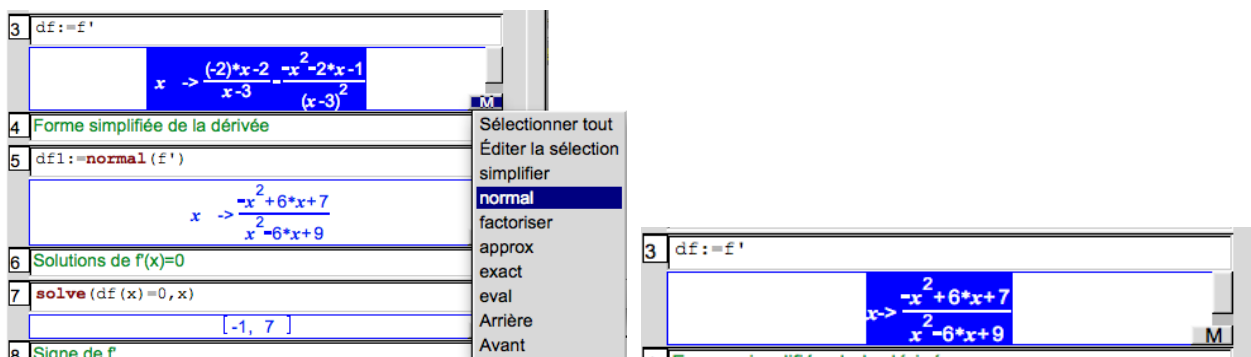
1	<code>f(x) := (-x^2-2x-1) / (x-3)</code>
	<code>// Interprète f</code> <code>// Succès lors de la compilation f</code>
	$x \rightarrow \frac{-x^2-2x-1}{x-3}$
2	Calcul de la fonction dérivée
3	<code>df:=f'</code>
	$x \rightarrow \frac{-2x-2}{x-3} - \frac{-x^2-2x-1}{(x-3)^2}$
4	Forme simplifiée de la dérivée
5	<code>df1:=normal(f')</code>
	$x \rightarrow \frac{-x^2+6x+7}{x^2-6x+9}$
6	Solutions de $f'(x)=0$
7	<code>solve(df(x)=0,x)</code>
	$[-1, 7]$
8	Signe de f'
9	<code>solve(df(x)>0,x)</code>
	$[(x>(-1))\text{and}(3>x), (x>3)\text{and}(7>x)]$

En [1], on définit notre fonction ; en [2], on met un petit commentaire (menu « outils → nouveau commentaire »), et en [3], on définit la dérivée de la manière la plus évidente qui soit. Mais le résultat n'est pas franchement sympa...

Dans ce cas là, on peut l'améliorer en ajoutant la commande « **normal** » ou encore « **simplifier** ».

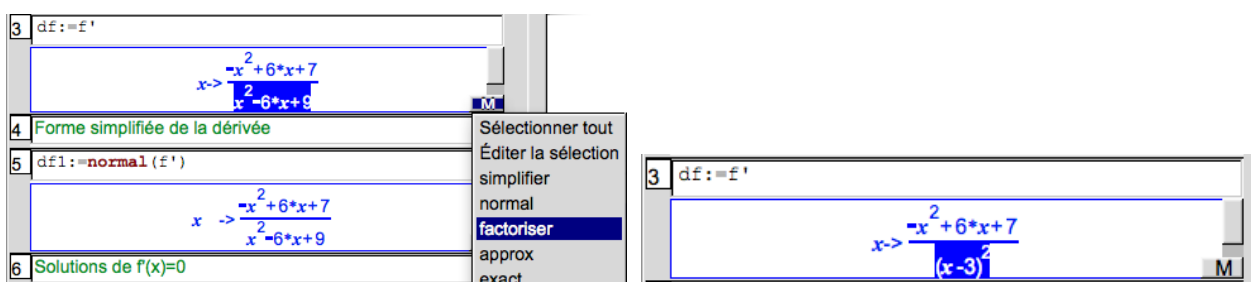
Ces deux commandes ne sont pas des synonymes ; on testera l'une ou l'autre selon les cas de figure, de manière à obtenir un résultat convenable. Nota : ici, j'ai créé une autre fonction dérivée $df1$ à des fins de démonstration. Il aurait suffi de corriger la commande [3] après coup en rajoutant « normal ».

L'interface de Xcas nous réserve aussi d'autres bonnes surprises, notamment un menu local situé à droite de toute ligne de résultat. Voyons ça de plus près :



Quand la machine nous donne un résultat dont l'aspect n'est pas trop avenant, ou que la forme ne nous convient pas, on peut rectifier le tir en sélectionnant le résultat (il devient bleu), et en cliquant dans le « M » situé à droite du résultat : un menu local apparaît. Sur la figure ci-dessus, on peut choisir « normal », et on obtient une forme plus exploitable.

Mieux, on peut appliquer le traitement uniquement à une partie de l'expression : ci-dessous, on factorise uniquement le dénominateur.

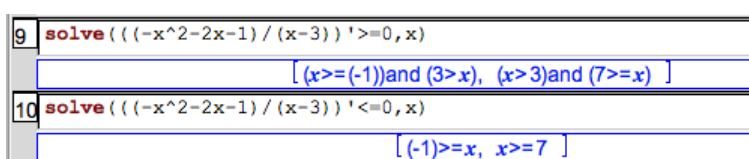


Mais attention : si vous recalculiez toute la session (touches « CTRL + F9 »), ces modifications seront écrasées, il vous faudra les refaire.

Poursuivons l'analyse de nos résultats : la commande [7] nous donne les valeurs de x qui annulent la dérivée ; et la commande [8] nous donne le signe de la dérivée, qu'il faut interpréter ainsi : $x > -1$: $x \in]-1 ; +\infty[$, $3 > x$: $x \in]-\infty ; 3[$, $x > 3$: $x \in]3 ; +\infty[$, et $7 > x$: $x \in]-\infty ; 7[$.

On a vu que les virgules correspondent à une réunion et les « and » à des intersections, on aura donc au final : $x \in]-1 ; 3[\cup]3 ; 7[$. Cet ensemble sera celui sur lequel notre fonction est croissante, car la dérivée y est positive.

L'élève qui a fait tout son calcul et qui veut juste vérifier son tableau de variations peut simplement taper la commande suivante [9] :



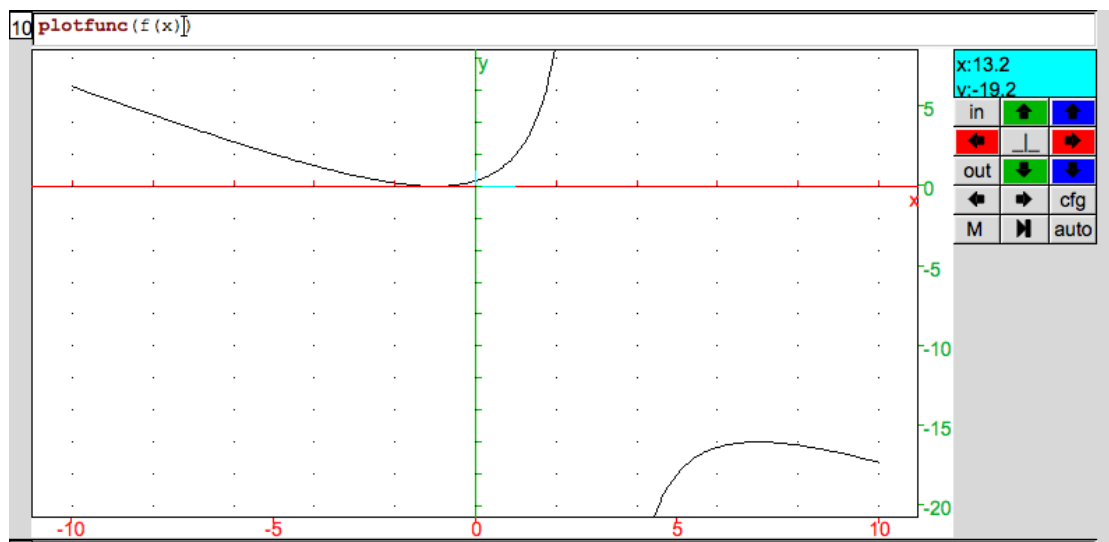
Il y a beaucoup de parenthèses : vérifiez bien l'expression saisie. On s'est contenté ici d'écrire l'expression de $f'(x)$ entre parenthèses, et d'ajouter « ' » à la fin : ceci correspond donc à $f'(x)$.

En [10], on a répété la même commande, mais avec $\leq$ au lieu de $\geq$. On note la différence dans le résultat : en [10], 3 qui est valeur interdite n'apparaît pas dans le résultat, car cette valeur est en

dehors de l'intervalle réponse. Les deux réponses sont évidemment bonnes, mais la réponse à la question [9] est plus complète pour notre problème que celle de la question [10].

Encore une fois : il faut bien réfléchir au résultat renvoyé par la machine, prendre le temps de l'interpréter et de le confronter à d'autres informations (un tracé graphique par exemple). Et comme on l'a vu en [9] et en [10], poser deux fois la même question à la machine sous deux formes différentes peut être très instructif !

Pour parfaire le tableau, on peut aussi évidemment tracer le graphe de la fonction dans le plan avec la commande « **plotfunc** ». Les graphiques n'étant pas spécialement intuitifs à manipuler, on en restera là dans ce document :



Si vous êtes un peu curieux, l'aide vous en dira plus...

3.4. ÉTUDE DE FONCTIONS : LIMITES (TS) ET INTÉGRATION (TS ET TES)

3.4.1. Limites

En terminale S, on étudie les limites de fonctions. La réponse apportée par Xcas est très simple, avec la commande « **limit** », il y a juste quelques astuces de syntaxe à connaître :

```

11 limit(f(x),x,+infinity)
12 limit(f(x),x,-infinity)
13 limit(f(x),x,3,+1)
14 limit(f(x),x,3,-1)

```

On retrouve le même type de syntaxe qu'avec la commande « **solve** » : il faut dire à Xcas quelle est la variable utilisée, d'où le « $f(x)$, x ». Quand on veut la limite en $\pm\infty$, on utilise les mots clé « **+infinity** » ou « **-infinity** ». En [13] et [14], on cherche les limites près de la valeur interdite 3 : il faut rajouter un argument, qui vaut +1 si on cherche la limite par valeur supérieure ($x \rightarrow 3$ et $x > 3$), et -1 si on cherche la limite par valeur inférieure ($x \rightarrow 3$ et $x < 3$).

3.4.2. Intégration

Là encore, on va voir que c'est un peu plus rapide avec Xcas qu'à la main. On va prendre une fonction du type $u'u^n$, avec ici $n = -2$. Il suffit de taper la commande « **int** » avec les bons arguments : en [18], c'est juste $h(x)$, donc le logiciel nous renvoie une primitive (sans constante) qui correspond au « catalogue » vu au lycée. Si on veut LA primitive qui s'annule en une valeur donnée, on se rappellera de la formule suivante :

$$\int_a^x f(x) dx = F(x) \text{ avec } F(a) = 0$$

Ceci est mis en application en [19], où on a rajouté deux arguments qui sont simplement les bornes d'intégration : ici, on cherche la primitive qui s'annule en 0.

L'intégrale prise entre 2 et 3 de la même fonction s'obtient juste en donnant les bornes d'intégration, comme en [20].

En [21], on calcule une primitive d'une fonction du type $u' \exp(u)$.

17	<code>h(x):=1/(x+1)^2</code>	
	<code>// Interprète h</code>	
	<code>// Succès lors de la compilation h</code>	
		$x \rightarrow \frac{1}{(x+1)^2}$
18	<code>int(h(x),x)</code>	
		$-\frac{1}{x+1}$
19	<code>int(h(x),x,0,x)</code>	
		$1 - \frac{1}{x+1}$
20	<code>int(h(x),x,2,3)</code>	
		$\frac{1}{12}$
21	<code>int(2x*e^(x^2),x)</code>	
		$\exp(x^2)$

4. RÉSOLUTION DE PROBLÈMES

4.1. RÉSOLUTION D'ÉQUATIONS ET DE SYSTÈMES

Nous avons déjà vu la commande « **solve** » pour résoudre l'équation $f'(x) = 0$. On va voir d'autres types d'équations et de systèmes d'équations.

1	<code>e1:=x-2y=3</code>	
		$(x-2*y)=3$
2	<code>e2:=-3x+4y=-1</code>	
		$(-3*x+4*y)=(-1)$
3	<code>solve([e1,e2],[x,y])</code>	
		$[-5, -4]$
4	<code>solve(ln(x+2)=3,x)</code>	
		$[\exp(3)-2]$
5	<code>e3:=ln(exp(x+2)-3)=4</code>	
		$\ln(\exp(x+2)-3)=4$
6	<code>solve(e3,x)</code>	
		$[\ln(\exp(4)+3)-2]$

Par exemple, en [1] et [2], on définit deux équations avec deux inconnues x et y . On note la syntaxe (standard Xcas) un peu particulière : on affecte une égalité à une variable. Il ne faut surtout pas se tromper quand on tape « = » et « := ». Cet exercice montre bien la différence entre une égalité et une affectation d'une valeur à une variable.

La commande « **solve** » [3] est la même que celle déjà utilisée, sauf qu'on met les noms des équations entre crochets et les inconnues aussi ; on aurait pu aussi mettre directement les équations dans la commande « **solve** », mais ça nuit un peu à la lisibilité. On peut bien entendu résoudre des systèmes avec beaucoup plus d'équations.

Dans tous les cas, même si le problème vous semble infaisable, donnez le à Xcas, s'il n'est pas content, il vous le dira, et sinon, il vous donnera une solution.

En [4] et [5], on a des équations faisant intervenir des logarithmes et des exponentielles : la syntaxe est strictement la même.

Il arrivera que Xcas soit mis en échec : l'équation [7] n'a pas de solution analytique dans IR, ni pour Xcas, ni pour le meilleur mathématicien du monde : Xcas nous le dit. On peut alors obtenir une approximation numérique avec la commande « **fsolve** »

7	<code>e4:=ln(x+2)+x=3</code>
	<code>(ln(x+2)+x)=3</code>
8	<code>solve(e4,x)</code>
	"Unable to isolate x in ln(x+2)+x-3"
9	<code>fsolve(e4,x)</code>
	1.69344135896

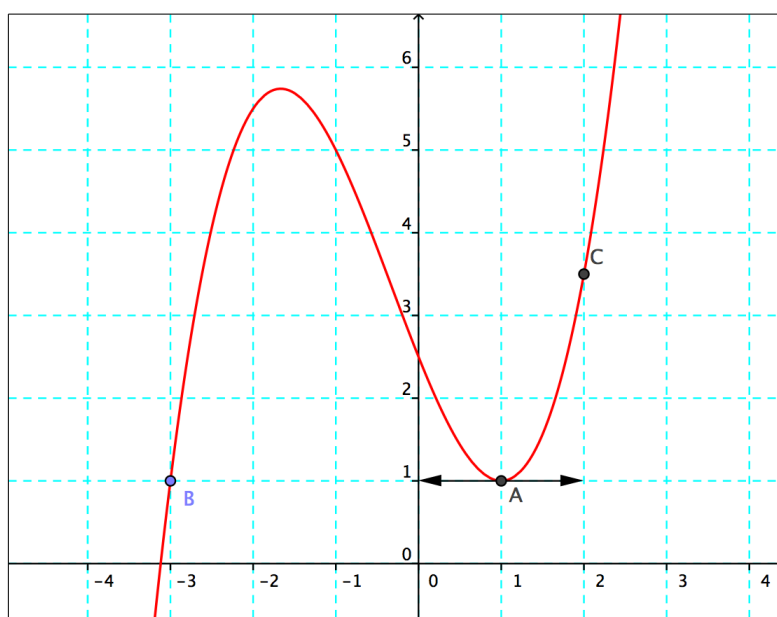
4.2.DÉTERMINER UNE FONCTION AVEC PARAMÈTRES

Un exercice fréquent consiste à donner l'énoncé d'une fonction dépendant de plusieurs paramètres, de donner des infos graphiques permettant de calculer des images et des nombres dérivés et d'en tirer des équations permettant de calculer nos paramètres.

Ex : soit la fonction f définie de la manière suivante :

$$f(x) = ax^3 + bx^2 + cx + d$$

D'après le graphique suivant, déterminez les images de -3, 1 et 2 par f , ainsi que le nombre dérivé en 1. Déduisez-en les coefficients a , b , c et d .



Voyons ce que ça donne avec Xcas. (on note qu'il faut savoir lire le graphique...)

1	<code>f(x):= a*x^3 + b* x^2+c*x+d</code>
	<code>// Interprète f</code>
	<code>// Attention: a,b,c,d, déclarée(s) comme variable(s) globale(s) lors de la compilation f</code>
	$x \rightarrow a*x^3 + b*x^2 + c*x + d$
2	<code>df:=f'</code>
	$x \rightarrow a*3*x^2 + b*2*x + c$
3	<code>e1:=f(-3)=1</code>
	$((-a)*27 + b*9 - c*3 + d) = 1$
4	<code>e2:=f(1)=1</code>
	$(a + b + c + d) = 1$
5	<code>e3:=df(1)=0</code>
	$(a*3 + b*2 + c) = 0$
6	<code>e4:=f(2)=7/2</code>
	$(a*8 + b*4 + c*2 + d) = \frac{7}{2}$
7	<code>solve([e1,e2,e3,e4],[a,b,c,d])</code>
	$\left[\frac{1}{2}, \frac{1}{2}, -\frac{5}{2}, \frac{5}{2} \right]$

En [1], on définit notre fonction : Xcas n'est pas gêné par le fait que les coefficients sont des valeurs littérales, et pas numériques. En [2], on définit la dérivée comme au § 3.3 : Xcas retrouve ses petits car la variable déclarée dans la définition de la fonction est x : pas de soucis.

Les lignes [3] à [6] définissent les équations : on rentre simplement les données lues sur le graphique ; et la ligne [7] donne la solution.

On peut aussi vouloir prolonger notre problème ; par exemple, quand notre fonction est définie, on peut vouloir déterminer ses variations, calculer les coordonnées des extrema locaux, etc.

On va procéder ainsi :

7	<code>[[a,b,c,d]]:=solve([e1,e2,e3,e4],[a,b,c,d])</code>
	$\left[\frac{1}{2}, \frac{1}{2}, -\frac{5}{2}, \frac{5}{2} \right]$
8	<code>f(x)</code>
	$\frac{5}{2} + \frac{x^3}{2} + \frac{x^2}{2} - \frac{5*x}{2}$
9	<code>df(x)</code>
	$-\frac{5}{2} + \frac{3*x^2}{2} + x$
10	<code>[x1,x2]:=solve(df(x)=0,x)</code>
	$\left[-\frac{5}{3}, 1 \right]$
11	<code>f(x1)</code>
	$\frac{155}{27}$
12	<code>f(x2)</code>
	1

On réaffecte les solutions de l'équation aux variable a , b , c et d en [7]. Notez la syntaxe particulière, avec double `[[ ]]`. Ceci s'explique par le fait que Xcas donne une liste de solutions, et cette liste est exprimée comme ceci : $[s_1, s_2, \dots, s_n]$ dans le cas où il y a n solutions. Ici, il n'y a qu'une solution, qui est composée d'un **quadruplet** ; or, Xcas définit ce quadruplet comme une liste. Au final, on a une liste dans une liste, ce qui donne `[[a, b, c, d]] := ...`

On voit en [8] et [9] que les fonctions f et df ont intégré les valeurs des paramètres, et sont donc opérationnelles : on peut calculer les valeurs des extrema ainsi que leurs images. Nota : en [10], l'équation a deux solutions x_1 et x_2 , ce n'est pas un doublet, on écrit donc `[x1, x2] := ...`