CALCUL FORMEL, Fonctions et Suites

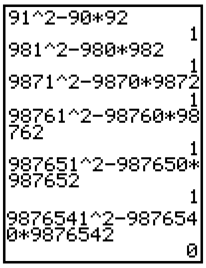
L’objectif, l’essence, qui a encore augmenté, de ce qui suit est d’essayer de comprendre comment ça se passe dans le petite boite pleine de boutons.
Le calcul formel c’est travailler sur les formes, faire du calcul littéral. Ce que l’on opposera au calcul numérique <u>fait d’approximations</u>.
Il est plus que conseillé de lire les écrits super intéressants de Philippe FORTIN, par exemple les cahiers T^3 « Questions - Réponses TI 89/TI 92 », « Introduction au calcul formel » ainsi que le cahier « Fonctions et Suites sur TI 89/TI 92 », rédigé par Bernard EGGER. De nombreux exemples en sont extraits.

Passer d’une calculatrice ‘ normale’ à une calculatrice ‘formel’ est un peu surprenant. D’un coup l’outil semble ‘intelligent’ ! Et ce d’autant plus que du côté calcul numérique [à partir d’entiers, pas de décimaux ‘approchés’ (*)] elle se défend rudement bien.

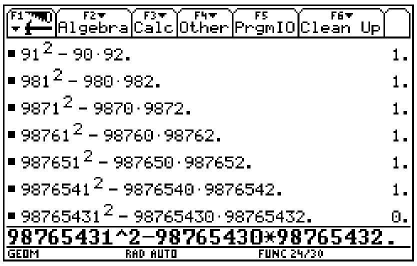
Quelques rapides essais et comparaisons nous montrent les possibilités extraordinaires de cette petite chose qui en plus ‘calcul formel’.

Démarrage gentil : sur toute calculatrice j’effectue quelques tests (certains connus comme les suites stationnaires qui donnent zéro par mauvaises approximations, voir page 3), ce qui nous fait bien rire parfois. Celui qui me plaît parce que j’en ai une utilisation directe en cours, est basé sur un calcul évident d’identité remarquable… a^2-(a-1)(a+1)=1. Ce qui me permet de déterminer le ‘nombre de chiffres utilisables’, ou, ‘quand c’est-y donc que 1 vaut 0’.

TI 83

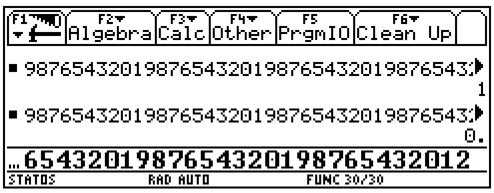


TI 92



Partant de 91^2-90*92, que j’écris au tableau pour les élèves, je demande d’utiliser une calculatrice pour en connaître le résultat. C’est ‘directif et obligatoire’ allez hop (elles trouvent toutes 1, ouf !). Puis je glisse un nouveau chiffre et je réitère jusqu’à ce que les élèves abandonnent (c’est rapide) parce qu’ils disent connaître le résultat. Persuasion passant par là, ils en arrivent ouske je veux (non mais !).

C’est à dire que certaines calculatrices trouvent encore 1 et d’autres 0. Pourquoi ? Eh bien paske (voir écrans) pour la 83 par exemple, quand j’en suis à 9876541^2 qui est en gros $(9.10^7)^2$ qui au même tarif de gros compte 14 chiffres à traiter, demande de comparer deux ‘énormes nombres’ qui ne diffèrent que d’une petite unité ridicule… Autrement dit, (mais oui je peux le dire) c’est comme comparer deux nombres à 10^{-14} près… le petit un vaut rien (1=0 quoi !).
On peut remarquer l’arrêt de la 83 au dessus de 12 chiffres, et celui de la 89 92 au delà de 16 chiffres. MAIS CE N’EST PAS VRAI pour la 89 92. J’ai en effet illustré ici le crochet ci dessus quelque part indiqué [à partir d’entiers, pas de décimaux ‘approchés’ (*)]. J’ai osé tricher d’entrée, les nombres sur la 89 92 sont des décimaux (je force le mode décimal en les terminant par une virgule, i.e. un point in English dans le texte), soyez observateurs que diable.



Reprenons le calcul, je vous laisse regarder l’écran ci-contre, sachant qu’à l’aide de ‘copié-collé’ je me fabrique un petit nombre de trois cent chiffres, donc six cent par carré et produit, ça marche toujours ! A quatre cent non (le résultat étant un zéro <u>pointé</u>). Je vous laisse chercher la limite qui craque. Aucune récompense promise, rien que le plaisir.

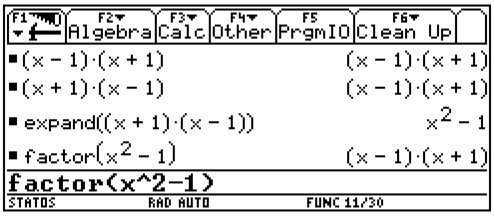
Bon, déjà, un p’tit kéke chose de plus, travailler sur les entiers n’est pas forcément pareil que travailler sur les décimaux (approchés, j’insiste).

Encore une remarque, un ti tour sur $\boxed{\textbf{2nd / VARLINK}}$ me dit :

Nb de chiffres de l’entier :	100	200	300	400	500
Taille en octets :	46	88	129	171	212

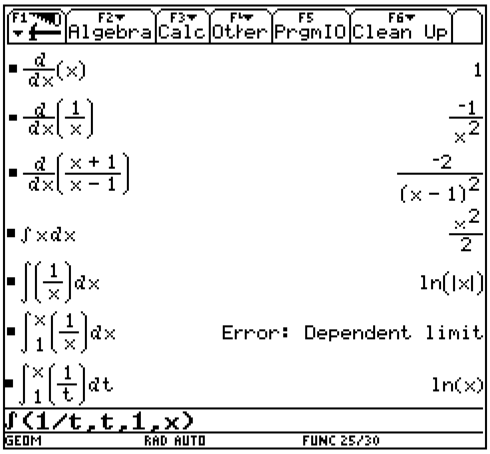
On peut constater une ‘réduction’ dans la représentation, avec une adéquation entre le nombre de chiffres de l’entier et son codage. Pour comparaison, 12 octets stockent les décimaux 200,54 ou $3,14*10^{150}$.

Après ces quelques calculs apéritifs, jouons du formel:



(x+1)(x-1) c'est *(x-1)(x+1)*. Ah bon, je suis déçu. C'est ça le calcul formel réécrire en échangeant les facteurs ? 'Meuh non' dit mon chien (il est polyglotte), 'c'est un problème d'écriture que tu vas devoir bientôt raconter'. Passer par **F2 / EXPAND** et youpee nous formalisons. *(x+1)(x-1)* c'est x^2-1 de même que **FACTOR** (x^2-1) c'est *(x-1)(x+1)*.

Du coup, je prends de l'assurance, je me mets aux petites choses connues :

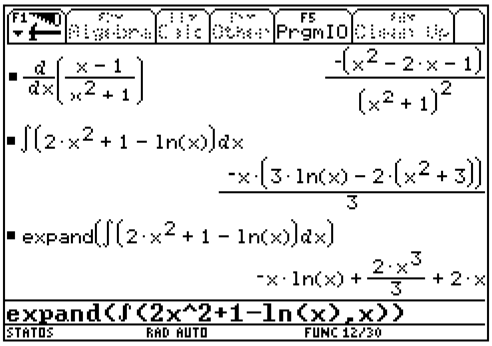


Une petite dérivée par ci, une petite intégrale par là…

Rien que du très commun, mais je tests 'petit', ce que j'utilise et connais bien.

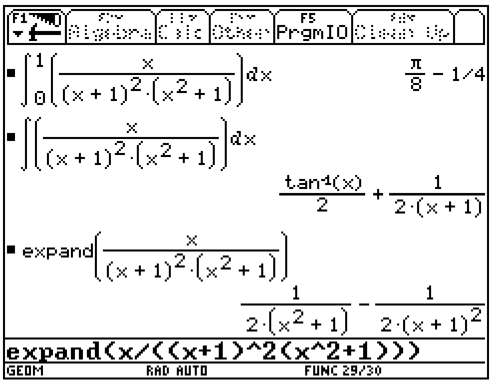
Les derniers résultats me plaisent fort :
$ln|x|$, **Error Depend limit**, et $ln(x)$. C'est plutôt bien tout ça.

Compliquons :



$\left(\frac{x-1}{x^2+1}\right)' = \frac{-(x^2-2x-1)}{(x^2+1)^2}$ ce doit être juste, mais pourquoi l'écrire avec – devant ? (et pourquoi pas !). Nous verrons cela bientôt (des promesses, des promesses…).
Encore un problème d'écriture pour la suivante $\int (2x^2+1-\ln(x))\,dx$ et qui par magie (un petit coup d'expand) devient ce que j'aurais trouvé (à … près).

Toujours plus :



$\int_0^1 \frac{x}{(x+1)^2(x^2+1)}\,dx = \frac{\pi}{8}-\frac{1}{4}$ en moins de 2 secondes !
Alors 'qu'à la main' il faut écrire :

$$\frac{A}{(x+1)^2}+\frac{B}{x+1}+\frac{Cx+D}{x^2+1}$$ et chercher A, B, C, D.

Et dans le genre 'plus plus', ça se trouve ach'ment fastoche grâce à la machine (voir écran) avec B=0.

J'ai une petite impression qui monte, monte,…, si les élèves ont ça ET qu'en plus ils savent s'en servir… ! Mais à quoi Serge alors ?

Comment ça, ils ont même pas besoin de savoir ?

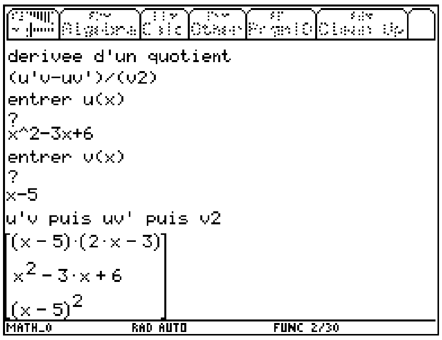
Il faut savoir qu'il existe de nombreux programmes 'qui font tout', genre 'etude', 'fonction', 'fx', 'deriqo', 'intpar', etc. que l'on trouve fastoche sur internet. T'as pas d'adresse ? demande aux élèves !

Voyons cela : en t^{ale} STL, en début d'année, je donne en DST :

Soit $f :]5\,;+\infty[\to \mathbb{R} \quad x \to \dfrac{x^2-3x+6}{x-5}$ quelques questions puis,

4. Montrer que la dérivée f' de f sur I peut s'écrire $f'(x) = \dfrac{x^2-10x+9}{(x-5)^2}$

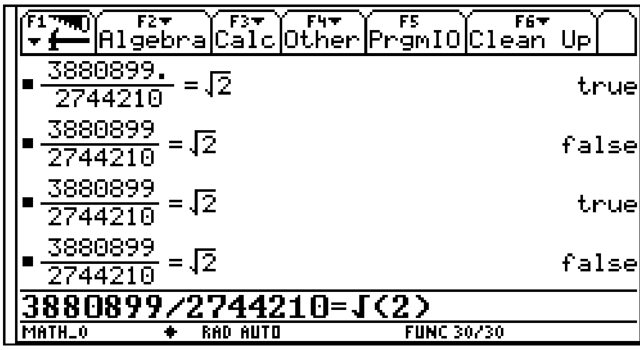
Prenons alors, par exemple le programme deriqo (qui appelle une fonction 'deri'). Je me demande qui a osé faire ça. Bon bien sûr, mon gamin allait passer le bac… m'enfin ! Comme je suis bon et généreux, je me pardonne.

Le programme 'deriqo'	La fonction 'deri'		
Prgm setFold(math_0): Disp "derivee d'un quotient" Disp "(u'v-uv')/(v2)" Disp "entrer u(x)":Input ux Disp "entrer v(x)":Input vx Disp "u'v puis uv' puis v2" Disp deri(ux,vx) EndPrgm	`deri(u,v)` `[[v*¶(u,x)][u*¶(v,x)][v^2]]`		En tapotant ils ont juste à vérifier que *(x-5)(2x-3)-(x²-3x+6) =x²-10x+9* ! Et la machine le leur dit !

Faut-il interdire la calculatrice ? (aux élèves, pas aux profs !)
Chacun ses égouts…, certains profs laissent tous les documents de cours et exercices pendant les devoirs. Ceux qui peuvent s'en servir ne le font que rarement par hasard !

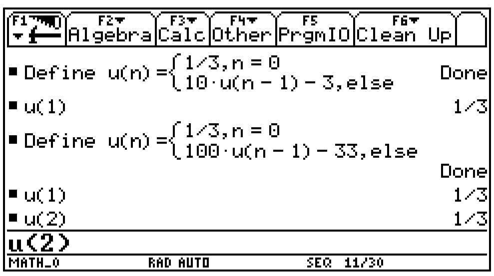
Nous sommes avertis, et donc, c'est à nous qu'il importe de poser des questions où il est plus utile de raisonner et réfléchir que d'être pointu en manipulations et techniques opératoires. (éternel débat : à quoi sert de faire 200 exos sur la dérivation… est-ce que le plus important c'est l'application de la notion ou de posséder la technique opératoire ? bon, les deux je l'accorde, mais sans exagérations !). Déchargés du calcul bête (mais à savoir faire et à pratiquer en solitaire) ils pourraient (!) passer plus de temps sur le 'pourquoi' et 'comment' on le fait.

Intermède burlesque :

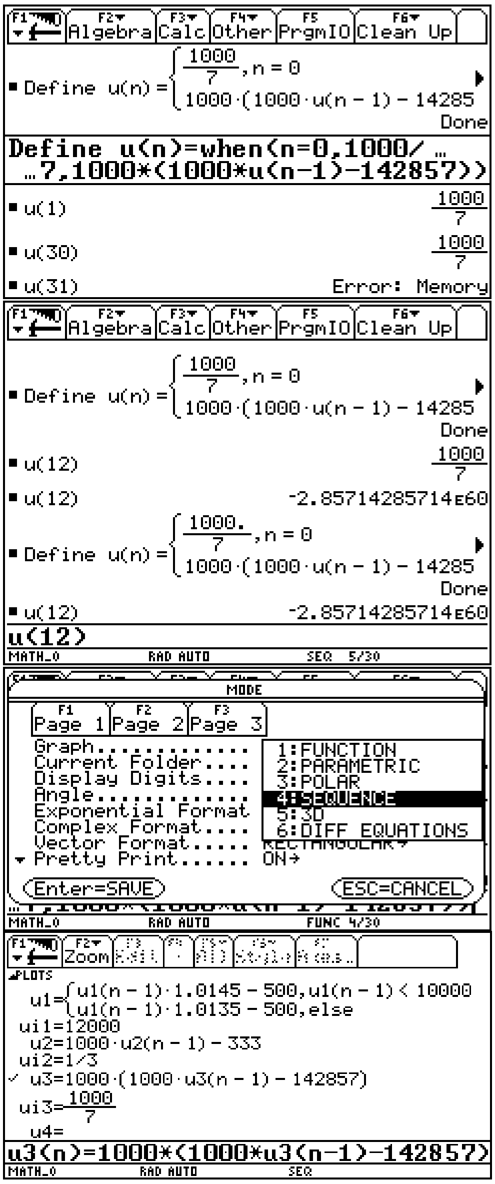
	Grâce à un petit programme de développement en fractions continues, je peux déterminer une fraction rationnelle 'équivalente' à un réel donné, jusqu'à une certaine précision. Ici $\sqrt{2}$. L'écran ci-contre nous montre que la rationalité de $\sqrt{2}$ c'est du pile ou face… ! Ouvrez l'œil… quelle sera la prochaine indication ?

Et en Suites ?

Continuons le sourire : fabriquer une suite stationnaire n'a pas grand mérite. Il suffit de comparer un rationnel à lui même, en diminuant le nombre de chiffres significatifs contenus dans la machine grâce à sa période. Etant certain de ne pas être très clair, un petit exemple valant mieux qu'un long discours, illustrons notre propos avec 1/3 dont la période est 3.



Poser $u_1=1/3$ puis $u_n=10u_{n-1}-3$ ou $u_n=100u_{n-1}-33$ ou $u_n=1000u_{n-1}-333$ ou, … vous avez compris.

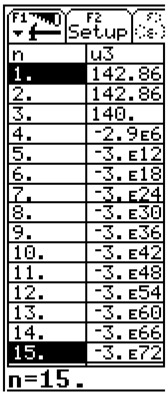


Innovons, prenons 1/ 7=0,<u>142857</u> période de longueur six (il faut multiplier par 10^6 et soustraire 142857).
Il suffit d'écrire u_1=1 / 7 puis $u_n=10^6\ u_{n-1}$-142857 ou ce que vous trouverez.
JE décide de prendre u_1=1000 / 7et u_n=1000 (1000 u_{n-1}-142857)
La suite reste stationnaire (c'est bien ça), mais je ne peux pas aller très loin pour les calculs. 'Tilt' pour le 32$^{\text{ième}}$ terme.

Ce qui veut dire que les calculs sont bien faits. Nous sommes en mode algébrique, tous les nombres utilisés sont des entiers (ceux donnés en définition :1000, 7).
Remarquons que le passage en numérique entraîne le même type d'erreur que sur les autres calculatrices : premier u_{12} correct, second à problème (j'ai appuyé sur la touche…).
Et si directement dans la définition de la suite j'inclus un décimal (vous avez vu le point, donc la virgule avant même que je n'en parle Bravo !) le résultat calculé est aussi mauvais que précédemment. Ah, le numérique !

Justement tient, parlons en du numérique. Du mode algébrique passons au mode 'suites' donc **SEQUENCE** sur la machine.

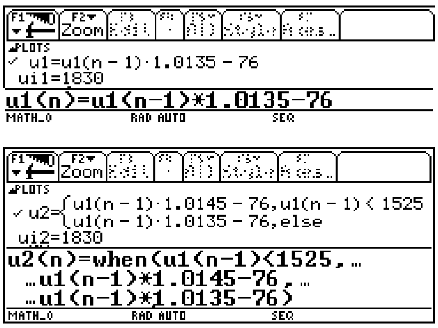
(Ne pas oublier de revenir au mode standard **FUNCTION** après quand ce sera fini).

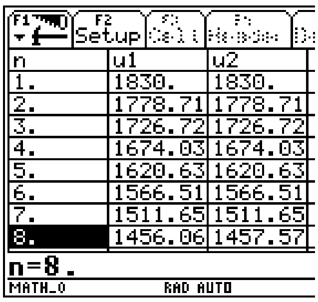


Définir la suite dans l'écran **Y=** qui n'affiche plus les fonctions mais les suites.
Aller dans **TblSeT**, départ 1, pas de 1.
Résultat dans **TABLE**.
C'est du numérique, la preuve, n=15**.** si 15 est pointé, les calculs sont en mode approché, d'où erreurs rapides.

Prenons un petit problème de crédit, description et commentaires, voir : TI-MAG septembre 2001, ou 'TI 89 exemples d'utilisation en classe' cahier T[3] ou '**perso.wanadoo.fr/serge-etienne**' un site sympa.

Pour acheter une jolie mob, un petit emprunt de 1 830 € (≈12 000 F) est contracté en 'crédit permanent' dit aussi 'revolver' parce que…, avec remboursement mensuel de 76 € (≈500 F) (le taux est donné ci-dessous).





Pas de difficulté sur une machine quelconque munie d'un tableur (ou par programmation).
Sauf que condition : 'pour toute somme inférieure à 1 525 € (≈10 000 F) le taux est de 1,45 % <u>par mois</u>, et de 1,35 % sinon'.

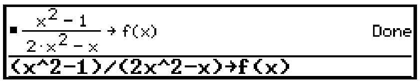
Là la (pas lalère) TI 89 92 ou V200 montre sa supériorité par la possibilité d'utiliser l'instruction '$\boxed{\text{WHEN}}$'. En mode suites ($\boxed{\text{Sequence}}$), déclarer (dans $\boxed{\text{y=}}$) $\boxed{\text{U1=When(U1(n-1)<1 525, U1(n-1)*1.0145-76, U1(n-1)*1.0135-76)}}$ puis $\boxed{\text{U1=1 830}}$.

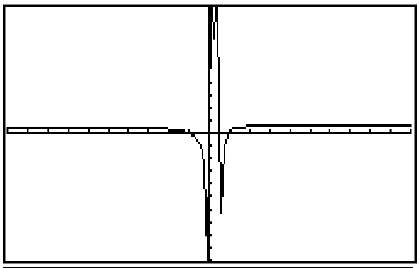
Remarque : le terme général c'est U_1 Pas U_{i1}.

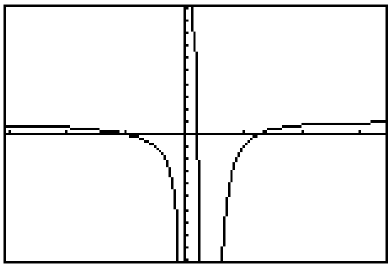
Fin du petit tour d'horizon. Bon dépêchons nous de trouver quelques problèmes sur ces machines, sinon je sens que je risque de jeter la machine par la fenêtre et poser ma tête sur le bureau, ou le contraire… (je ne dis ça que parce que je suis au rez de chaussée..).

D'une part les 'bugs', qui sont corrigés autant que possible lors de la mise à niveau du logiciel (c'est l'un des avantages de l'appellation 'Flash'). Qu'est-ce qu'un bug ? c'est une erreur 'anormale' de la machine. Ce qui sous entend qu'il y a des erreurs 'normales'…
Si vous en connaissez, il suffit de les signaler (site TI ou tout animateur TI passant par là). Ceux que je connaissais ont été corrigés, donc pas d'exemple. il est inutile de dire 'quand on faisait ça…'.

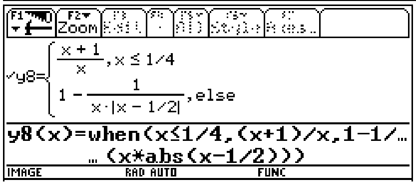
D'autre part les erreurs, qui elles même sont de deux types. Celles de l'utilisateur, et celles que l'on attribuera à la machine 'bien que'. C'est le sujet principal de ce roman.







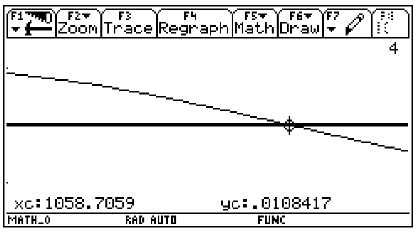
f mise dans $\boxed{\text{Y1}}$ en zoom standard ([-10 ; 10] et [-10 ; 10]). L'élève pense (si si, j'en ai vu) à la courbe de droite alors que… reprendre le tracé sur [-5 ; 5].

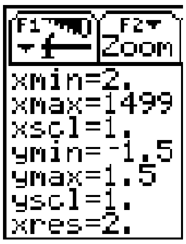


Pour obtenir la courbe de droite, sur l'intervalle [-5 ; 5] j'ai utilisé la fonction ci-contre. C'en est une qui donne l'aspect de ce que je voulais.

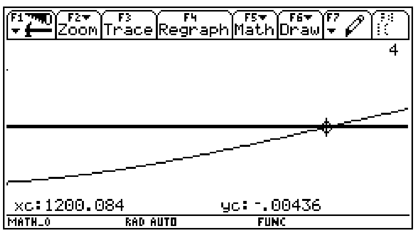
Un peu éventé ?… si vous en avez d'autres, merci par avance : **serge-etienne@wanadoo.fr**

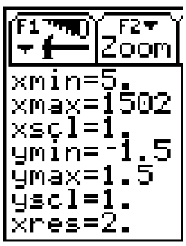
À partir d'une idée d'Alain Ladureau :





Pour une fonction connue (et sympa car définie et tout et tout)…, représentation graphique sur l'intervalle [2 ; 1499].
Mes impressions : fonction strictement décroissante + théorème bijection d'où solution unique pour $x \approx 1060$.

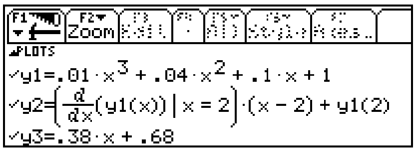




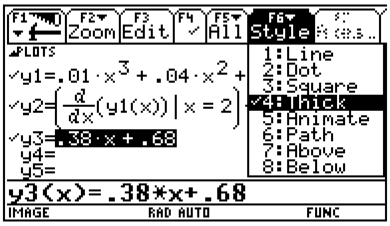
BON,… toujours avec la même fonction,
Sur l'intervalle de même longueur [5 ; 1502], je ressens comme un malaise… elle serait strictement croissante, et la solution unique serait alors ≈1200 !

La fonction n'est autre que sin(x). Le pas obtenu produit un bel effet.

Le calcul formel c'est bien du calcul… (exemple suivant) :

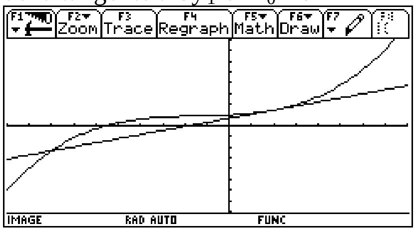


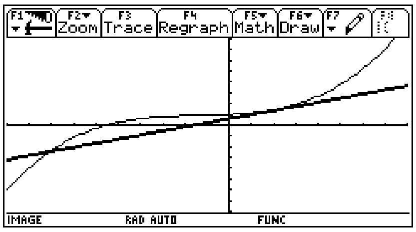
Je déclare ces petites fonctions.
Pour $y_2(x)$ je reconnais la formulation de la tangente à Cy_1 en $x_0=2$.



Et en y_3 c'est encore la tangente à Cy_1 mais à partir des valeurs calculées pour $y_1(x_0)$ et son nombre dérivé en $x_0=2$.

Comme il va se passer des choses, je demande que Cy_3 soit tracée en gras.





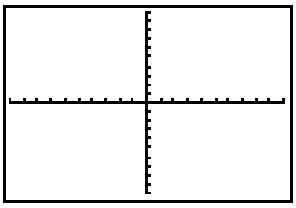
Bon, sur le papier vous n'avez rien vu…
Le temps de tracé de y_2 est d'environ 25'' alors que celui de Cy_3 n'est que de 5'' tout au plus. Pourquoi ?
Parce que dans le premier cas notre gentille calculatrice calcule pour chacun des points du tracé, la dérivée de $y_1(x)$, puis son image. C'est long.

Encore un, j'utilise une 83 pour changer, mais c'est pareil sur les autres :

Mais où donc est la courbe ?

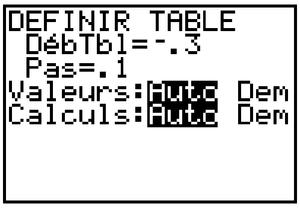
Soit $f(x)=\dfrac{0{,}1}{(1-10x)}$. Question : tracer la courbe représentative de *f* sur I=[-10 ; 10].

Entrer la fonction dans Y_1 par exemple, choisir Zoom Standard.

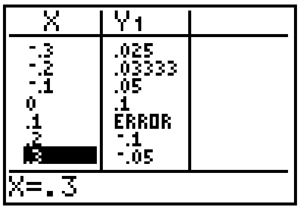


Résultat intéressant et "presque" surprenant, surtout si ce sont des élèves qui se sont précipités sur la calculatrice. Que faire ?

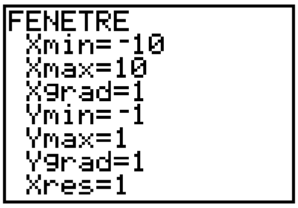
Une observation du tableau de valeurs permet de régler la fenêtre d'affichage (y dans [-1 ; 1] et suppression des axes).



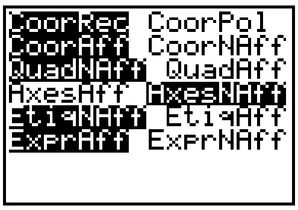
On sait où chercher…



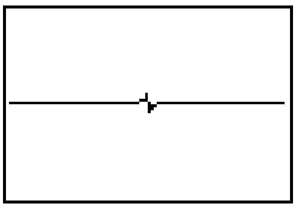
indication des valeurs prises par la fonction.



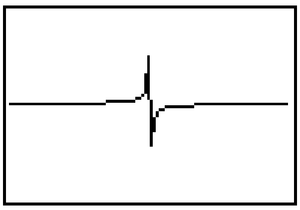
Caractéristiques de la fenêtre.



Ne pas afficher les axes.



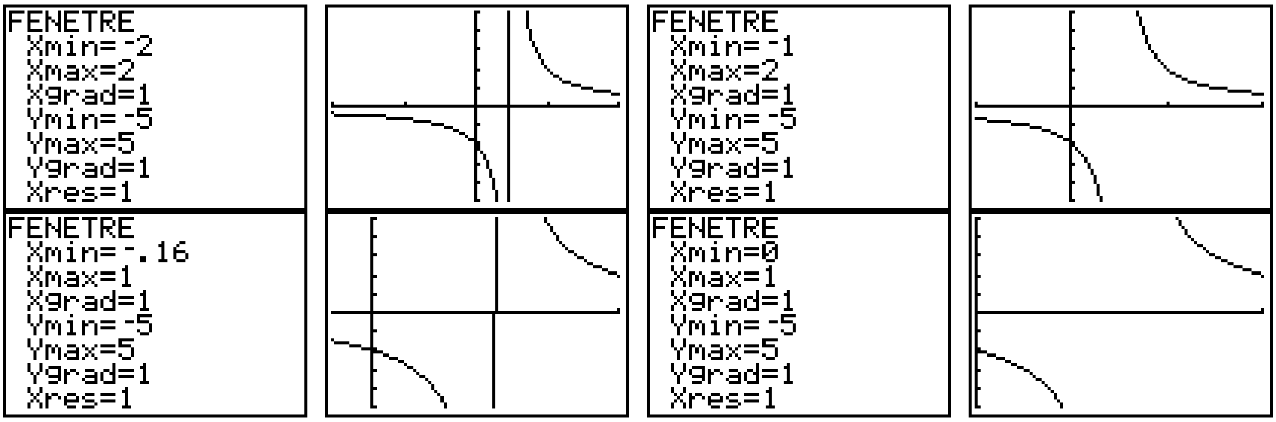
Le tracé permet d'apercevoir un début de courbe. Poursuivre la méthode.



Tracé obtenu pour y compris entre - 0,1 et 0,1. Attention à l'interprétation sur la fausse asymptote (voir plus loin).

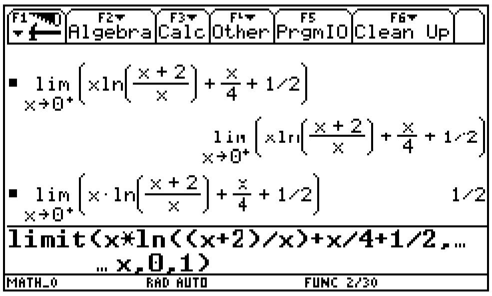
Un p'tit dernier (provisoirement… car j'attends les vôtres !)

Pour $f(x)=\dfrac{1}{\left(x-\frac{1}{2}\right)}$, tracé sur différents intervalles, autour de la valeur interdite. (Heu… si elle est interdite alors pourquoi que ?). Les commentaires sont après toutes ces belles images : l'intervalle est suivit de la représentation graphique.



Première image (courbe) je vois une asymptote, la suivante non, puis oui mais avec une cassure, puis non. Pourquoi cela ? parce que ce n'est pas le tracé de l'asymptote… c'est parce que le calcul de points de tracé fait que d'un bout de courbe à l'autre (vers les branches infinies) les points sont dans une zone « traçable » donc reliés. D'ailleurs, en style 'non relié' y a pas de tracé de 'fausse asymptote'.

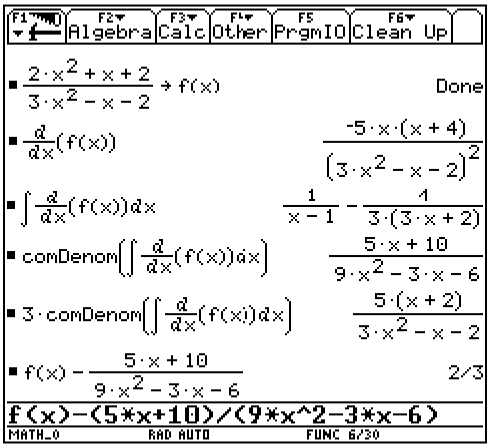
Continuons sur les erreurs fréquentes et remarquables :



Notre écriture triviale des fonctions nous fait parfois oublier quelques signes qu'il faut s'efforcer de mettre.

C'est une erreur fréquente de l'utilisateur.

Egarements de machine ?



Et si je cherche la primitive de la fonction que j'ai dérivée ?

Ben ça donne de drôles de choses (numérateur premier degré), mais pas ce que j'avais au départ (même forme mais numérateur deuxième degré), pourtant je me bats.

Oh, c'est vrai qu'à une constante près… de quoi perturber les élèves (mais pas les profs !).

Essayez avec $f(x)=\dfrac{2x^2+2x+1}{x^2+x-2}$ définie sur $\mathrm{I}=\left]1\,;\infty\right[$

Ce préambule nous montre que la machine fait de beaux calculs, dont le résultat parfois nous étonne. Du moins dans sa forme qui n'est pas toujours conforme à nos habitudes.

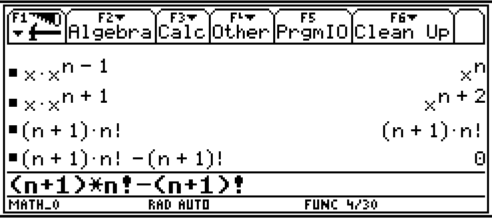
Par exemple, $\displaystyle\int\frac{x^2-10x+9}{(x-5)^2}dx=(\text{machine})\ \frac{16}{x-5}+x=(\text{nos habitudes})\frac{x^2-5x+16}{x-5}$, nous ne l'aurions pas écrit comme ça.

Un grand principe : une sorte de 'simplification' (guillemets volontaires).

En effet, nous avons : $\dfrac{x^2-5x+16}{x-5}=\dfrac{x(x-5)+16}{x-5}=x+\dfrac{16}{x-5}$, qui m'interpelle… « mais c'est bon sang bien sûr » sur un air d'accordéon… non, vous voyez pas ? OUI, HORNER. Je ne sais pas si c'est toujours comme ça, mais je soupçonne ici très fort le programmeur du logiciel d'avoir cherché à réécrire les fonctions sous la forme Horner qui a l'avantage d'un minimum de calculs, et donc sans doute aussi, d'une plus grande rapidité d'opération. Cette méthode fut d'ailleurs développée pour son utilisation en informatique ou calculatrice.

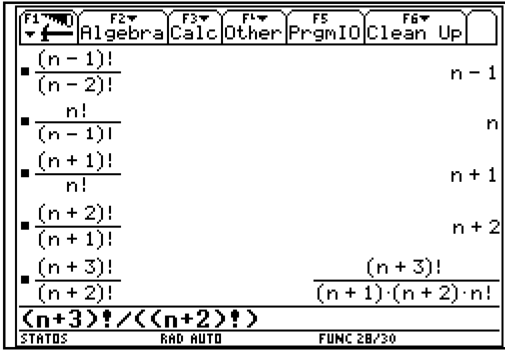
Le logiciel de calcul formel traduit les expressions comme ça lui a été demandé par son (ses) programmeur(s).

Un exemple P. Fortin :



La calculatrice 'semble' mieux connaître certaines règles avec les puissances qu'avec les factorielles. Pourtant faire la différence des deux expression donnant zéro, montre qu'elle sait que c'est bien la même chose. Mais elle ne sait pas qu'elle doit l'écrire autrement (comme notre habitude nous le suggère).

Je r'prendrais bien un p'tit coup de factorielle,



Je commence là où ça marche… pour arriver là où ça s'arrête. On peut être surpris de ce comportement. Dire ou crier 'bug'.

Pourtant… soyons raisonnables, lorsque <u>nous</u> travaillons (écriture) <u>nous</u> choisissons la forme qui nous paraît convenable en fonction de la situation. Avec certains logiciels de calcul formel (Dérive, Mapple,…) nous pouvons introduire certains choix. Sur 89 92 (Dérive réduit) <u>nous subissons</u> (le jour où la miniaturisation aura centuplé les possibilités des calculatrices…).

Notre représentation "cervicale" de la factorielle, ou de toute autre expression, nous permet immédiatement (récompense du travail et de l'expérience), de traduire sous une forme qui semble plus judicieuse et appropriée au problème posé, l'expression proposée.
C'est, pour l'instant, notre grande supériorité vis à vis des machines.
Il faut bien prendre conscience que notre calculatrice fonctionne non pas sur des reconnaissances globales, mais à partir d'un programme testant un certains nombres d'algorithmes.
Ce qui veut dire, qu'à un moment donné, il a été décidé par les concepteurs et les programmeurs du logiciels, de ce qu'est une forme 'simplifiée' d'un certain type d'expression.
On ajoute ensuite quelques "et si c'est de cette forme alors..." correspondant à des cas classiques que l'on déclare en "exceptions" (c'est à dire "forcer le traitement de cette situation autrement") qu'il est trop surprenant pour l'utilisateur de ne pas voir traités (faut bien que ça serve à quelque chose les râleurs qui trouvent des "erreurs").
La traduction de notre *n!* n'est qu'algorithmique (sous forme d'une fonction récursive), donc non écrite sous forme de facteurs. Cette traduction est complétée jusqu'à l'ordre *n-2* et *n+2* par des exceptions du genre *(n-2)!=n!/(n.(n-1)* et *(n+2)!=(n+1)(n+2)n!* qui permettent quelques simplifications évidentes et nécessaires (pour l'aspect sérieux du logiciel).
Au delà.. (revoir écran), il n'est pas possible de tout considérer comme "exception". Il faudrait un nouvel algorithme qui demanderait de nombreux tests supplémentaires et donc beaucoup plus de temps.
Pour l'heure, ils ont choisi.

Il nous appartient de maîtriser suffisamment les mathématiques pour en utiliser les outils ! et de prendre un peu de recul devant des résultats qui nous étonnent. Déjà, pour nous, nous ne sommes pas capables de donner une définition de LA bonne forme d'écriture d'une fonction.

Exemple (P. Fortin) :

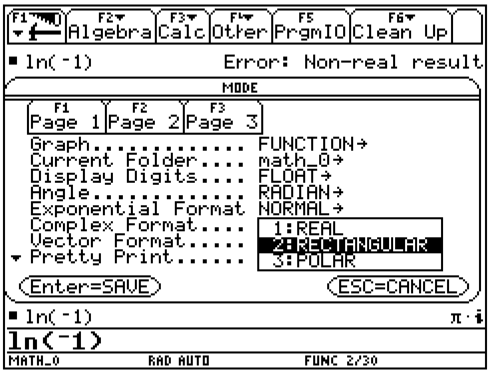
$p(x) = 243x^5 + 405x^4 + 270x^3 + 90x^2 + 15x + 1$ est une bonne forme pour calculer *p(0)*,
$p(x) = 1 + (15 + (90 + (270 + (405 + 234x)x)x)x)x$ est mieux pour d'autres calculs rapides,
$p(x) = (3x+1)^5$ pour factoriser sa dérivée, ou résoudre *p(x)*=0.

De même que vaut-il mieux écrire ln(4) ou 2 ln(2) ? etc.

A retenir : la calculatrice choisi UNE forme (dite simplifiée) traduite de l'expression entrée.

Et comment travaille-t-elle ?

Important (tout l'est…) Elle travaille dans les complexes.

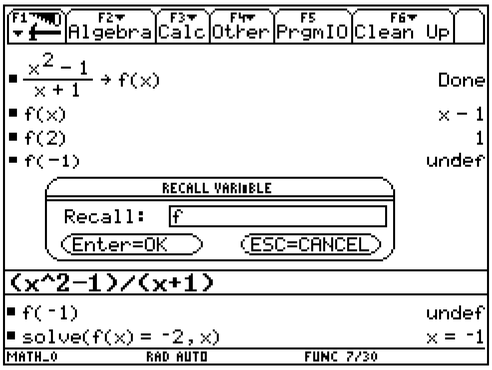


Etonnant de voir qu'il pourrait y avoir un résultat à ln(-1).

Passer en **MODE RECTANGULAR** (complexes), alors ln(-1) a bien un résultat. Revenir en mode réel.

MEME si nous sommes en mode réel, elle travaille dans les complexes puis elle revient aux réels, si le résultat le permet, sinon message d'erreur. (entre le début et la fin … peu importe !)

Elle 'simplifie', elle réécrit les objets (fonctions) dans 'sa' forme.



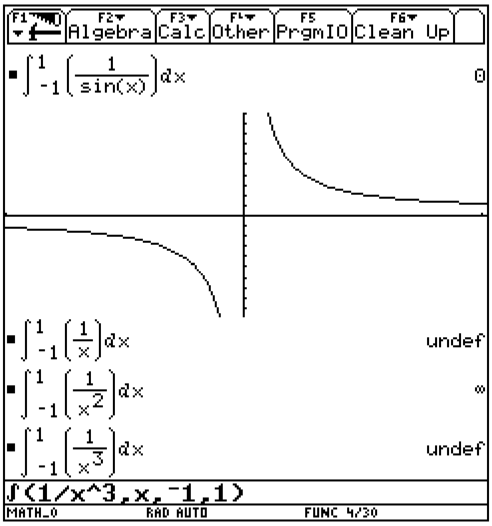
L'exemple ci-contre montre qu'il y a bien récriture de *f*. MAIS qu'à tout moment, les deux formes cohabitent en effet, *f(-1)* n'est pas évalué à partir de la forme 'simplifiée'.

Une évaluation semble avoir lieu chaque fois que l'on travaille sur des nombres.

Preuve que *f* est bien conservée sous sa forme originelle, en rappelant la fonction par **2nd STO** Remarque : entrer *f*, pas *f(x)* !

Nous avions une 'bonne' réponse pour *f(-1).*
Patatras… que nous dit-elle là ?

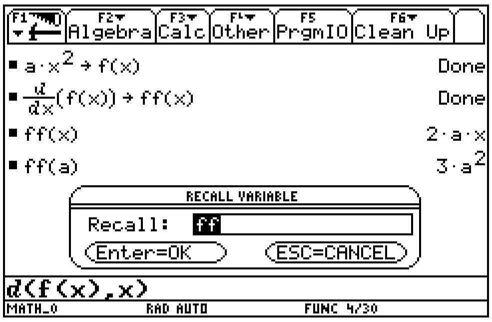
Et elle 'saute' des étapes…, voyons quelques intégrales 'pas propres'



Que fait elle ? l'algorithme se dit que c'est pas dur, symétrie, limites aux bornes tranquilles, donc fastoche.
Mais il oublie le point singulier. Que pourtant le tracé de la courbe nous rappelle.
(N'oubliez pas d'être en radians pour tout travail sur les fonctions trigonométriques).

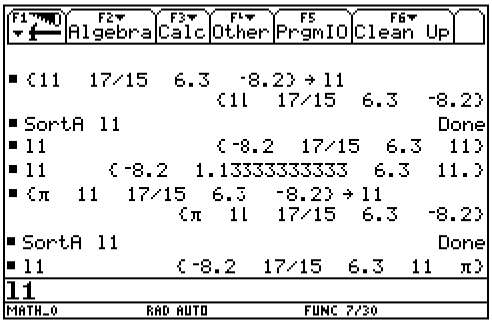
Et dans les autres cas ? il faut supposer qu'un 'traitement de faveur' a été demandé. Dans l'algorithme doivent figurer quelques 'exceptions'.

Revenons à un type d'erreur qui nous étonne 'Et pourtant pourtant…' !



Fonction de fonction n'est pas toujours fonction…
(Ne pas oublier le * entre a et x).
Par habitude, j'aime bien appeler *ff(x)* la dérivé de ma fonction *f*.
ff(x) ça va. Mais qu'est-ce donc ce *ff(a)* qui vaut 3a² ? (vous auriez préféré 2a²).
Comme (vérif par **2nd STO**) *ff(x)* c'est la dérivée pour la variable x, *ff(a)* c'est la dérivée pour la variable a. Or $\frac{d(a^3)}{da} = 3a^2$. ELLE a raison ! mais ça contrarie nos pensées.

Traitement des listes (tri de 'nombres') : (doc P.F.)

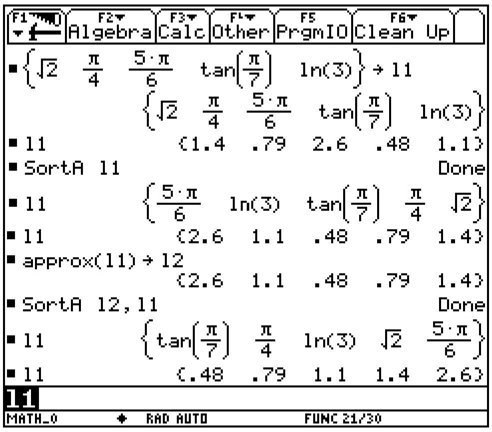


J'ai une liste de nombres. Je la trie par ordre croissant.
Je vérifie 'le bon tri' en mode approché (¥).

Alors je range π dans la liste, je trie et stupéfaction.

Pourquoi donc, je la Woua, dirait mon chien, la bonne place de π. Alors qu'est-ce qu'elle fait ?

Dur à traiter que de comparer des fonctions (et oui, π n'est pas un nombre). Ils n'ont pas mis d'algorithme spécifique. Nous pouvons tout de même trier.



Je donne du 'volume' à ce nouvel exemple.

Entrée des données,

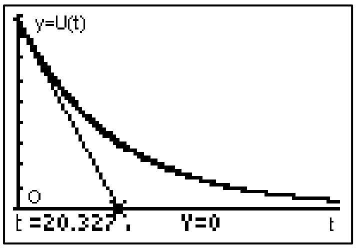
Tri

Valeurs approchées(¥) C'EST PAS TRIE.
Utilisation d'une liste auxiliaire DE VALEURS APPROCHEES, puis tri A PARTIR DE CETTE liste auxiliaire.

Résultats numériques, c'est trié.

Un peu de calcul formel, du vrai de vrai :

Le problème : mise en évidence d'une loi physique, la décharge d'un condensateur chargé, de capacité C=2200 µF au travers d'une résistance R=9200 Ω.
Pour la méthode opératoire et le texte, voir TI-MAG ou sur mon site le texte 'Fonction Puissance'.
A l'aide de capteurs il est mesuré la tension au borne du condensateur. Les nombres obtenus forment une progression géométrique, d'où courbe et équation $y = 8{,}436 \times 0{,}952^x = 8{,}436 \times e^{x \ln(0{,}952)}$.
Il reste à relier R et C aux résultats obtenus Le prof de physique pose τ=RC=2200*10^{-6}*9200=20,24.



Et… ce même physicien me chuchote qu'il suffit de mesurer la distance entre l'abscisse d'un point A quelconque (O par exemple parce que c'est plus précis) et l'abscisse de l'intersection de la tangente à la courbe en ce point A avec la droite des abscisses, et que c'est plus précis en x=0. Petite machine me dit : « nombre dérivé en *x*=0 de *y* c'est - 0,415 » merci machine.

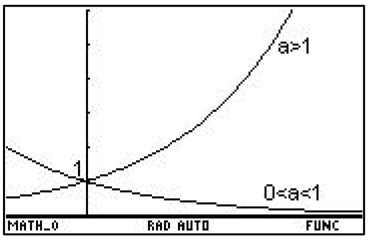
Equation de la tangente : y_{11}=- 0,430*x*+8,741

Tangente qui coupe l'axe des *x* en 20,33 (oui, 20,3277 !). Bon résultat.

Mais, nous n'allons quand même pas rester sur une impression de recette de cuisine… (la distance serait constante, mais plus précise à l'origine des abscisses…)

Au lieu d'étudier a.b^x je vais prendre a^x, pour éviter de traîner un coefficient qui n'est pas utile.

Il faut a>0, hé oui c'est un peu des maths, et si dans ℝ je n'ai pas de cas de conscience pour écrire $(-1)^2$, que penser de $(-1)^{1/2}$?

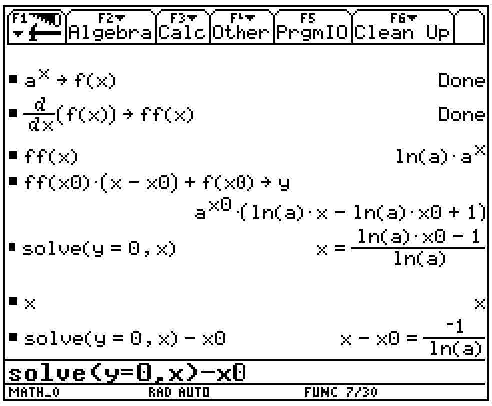


Nous obtenons deux courbes, selon que a soit un réel supérieur à un, ou qu'il soit compris entre zéro et un.

Elles ont bien l'aspect déjà remarqué.

Je connais la formulation habituelle de la tangente en un point d'une courbe définie par une fonction *f* :

$$y = f'(x_0)\,(x-x_0) + f(x_0).$$



je définis *f*, puis *f'* qui sera stockée dans ff(x) car *f'(x)* n'est pas accepté par la machine.

Calcul de *f'*. Il faut rappeler que a^x s'écrit aussi $e^{x\ln(a)}$.
Définition de l'équation de la tangente en x_0.

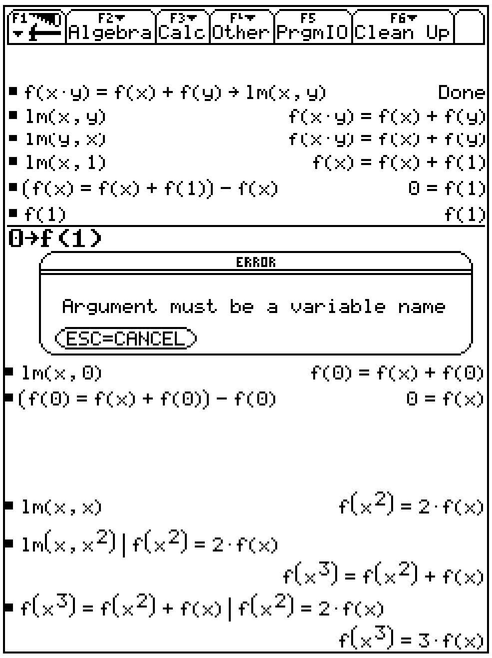
Je demande l'abscisse du point d'intersection de la tangente avec l'axe des abscisses.

Je fais remarquer que x ne contient rien…

Je demande la différence entre l'abscisse du point d'intersection et celle du point de tangence. C'est une constante, indépendante de x, ça ne dépend que de a. CQFD !

Tout ce qui précède est bien fonctionnel, rien de numérique. Je peux donc dire qu'il est possible d'utiliser la courbe et sa tangente en n'importe quel point, pas seulement à l'origine des abscisses. Il est vrai que pour mesurer, il vaut mieux avoir de la place, et que sur une pente presque horizontale l'erreur est risquée.

D'après doc de Bernard EGGER : La fonction ln peut s'aborder de diverses façons. Historiquement les calculs aux taux usuriers prenaient beaucoup de temps. L'idée était de passer de multiplications longues et laborieuses à des additions plus fastoches. En gros *f(x*y)=f(x)+f(y)* faisait rêver.
Que peut on faire dire à la machine en partant de cette seule définition, que l'on complétera si nécessaire ?



Je définis une réalité (la fonction *lm(x,y)*) à partir de mes rêves.

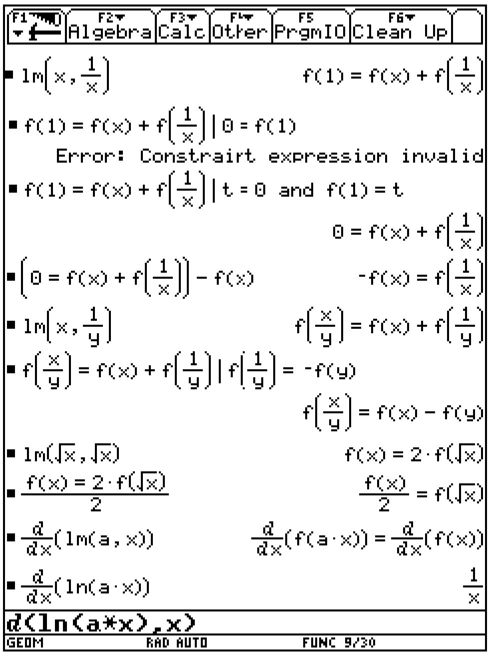
Je vérifie, elle existe, et respecte même la commutativité de l'addition.

Que peut-être l'image de 1 ? (comme si je ne le savais pas !). Il semble que ce soit 0. J'opère sur des égalités, je peux enlever *f(x)* membre à membre.
***f(1)* c'est bien zéro,**
quoique…
et bien sûr, je ne peux l'affecter de sa valeur. Ce sera une limite à mes opérations, il me faudra chaque fois rappeler ce résultat à la machine qui s'empresse de l'oublier (pas d'affectation).

Et l'image de zéro ? ben si elle existe, alors ma fonction est identiquement nulle.
DONC je supprime 0 de mon intervalle de définition.
C'est bien, sur des calculs totalement théoriques, nous avons la mise en évidence des particularités de notre fonction. Et même mieux, nous en voyons des propriétés algébriques :

Résultat direct.

Mais là nous retrouvons le manque d'inclusion des résultats obtenus à l'écriture des résultats suivants. Supériorité (provisoire ?) de l'homme sur la machine.



Je continue,

Bien, sauf que *f(1)* n'est pas affecté de la valeur 'déterminée' précédemment.

Et pb. Parce que la machine pose *f(1)*=0 et donc 'sort' du formel pour calculer, mais ne peut 'évaluer' *f(x)+f(1/x)*.

Mais en passant par une variable muette auxiliaire,

ça marche. D'accord, c'est du 'bricolage fonctionnel'.

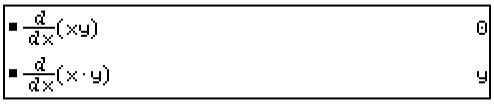
De même pour x*1/y

Et les racines.

Pour la dérivation ce que je demande ressemble bien à ce qui suit, c'est a dire ln, même si je n'ai pas explicitement la fonction dérivée.

Il y a du contenu.

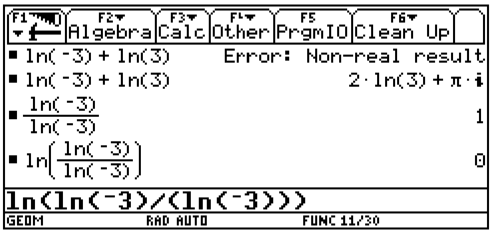
Profitons de l'occasion pour rappeler une erreur classique (toujours utilisateur).



Ne pas oublier le signe de la multiplication !

Pour instructif que cela puisse être, nous ne pourrons pas passer aux calculs numériques. Reprenons le document de Bernard,

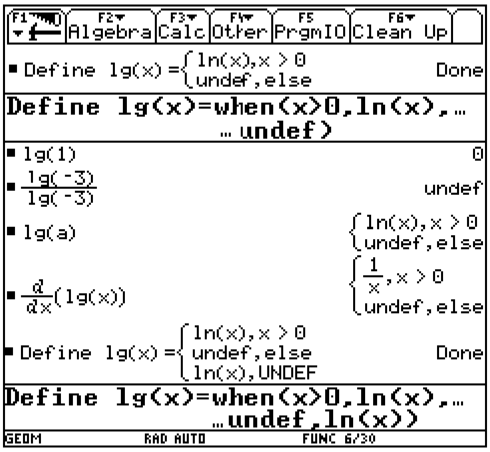
L'idée : ce serait bien si la réponse de la machine se faisait en fonction de notre cours, du domaine de définition que nous avons déclaré, et du niveau des élèves ou étudiants. Le problème principal, c'est que le(s) programmeur(s) fait travailler ses algorithmes dans ℂ, et ne retourne que le résultat, si possible, dans ℝ. Revoyons le :



Je suis en mode Réel.
Je passe en mode complexe, et je retourne en mode réel pour ce qui suit.

Et voilà qui illustre ce qui précède… calculs dans ℂ, et sans complexes, retour dans ℝ pour un résultat qui alors ne pose pas de problème.
Amusant non ? déroutant pour quelques uns, source d'erreurs pour les élèves.

Faisons comme Bernard, définissons notre propre fonction ln, baptisée lg pour l'occasion.



Définition de la fonction, y compris la déclaration dans le bandeau des entrées. Je suis en mode réel, œuf Corse.

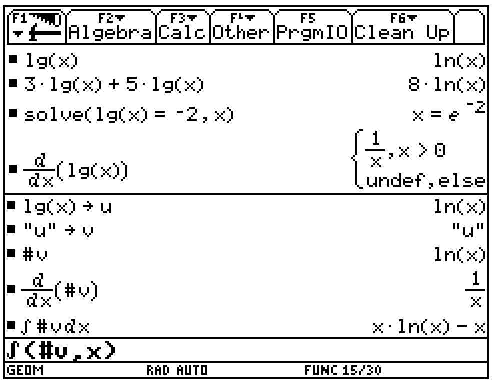
Petit essai,

Ah bien ça,

Mais là, nouveau(X) problème(S).

Alors ? c'est un peu notre faute, quand elle ne peut pas évaluer (non numérique, pas possible de comparer la variable, le paramètre, avec zéro), elle répond vis à vis de la déclaration faite.

Après quelques cours de yoga…, reprenons donc la définition.
Si x>0 Alors ln(x), sinon (pour x<=0) undef, et en cas d'indétermination (peut pas dire signe de x) ln(x).



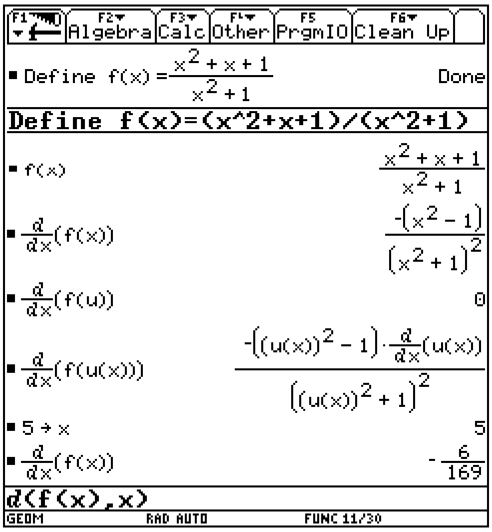
Du coup, ça marche.

Hé bé non, y a encore un blème. Qui vient du programme (algo). TOUT ne peut pas être évalué, donc faut faire avec.
Tout de même, des outils 'forçant' l'évaluation' existent. Par exemple #, l'indirection.
Notre objet étant placé dans une variable auxiliaire u, que l'on transforme en 'suite de caractères' 'u' stockés dans une deuxième variable v. (pourquoi faire simple…).
Alors, # permet de lire le contenu, en forçant l'évaluation de ce contenu. D'où le résultat attendu. La méthode pouvant être incluse dans un programme. Nous pourrions utiliser cette définition en classe.

Pour se reposer des efforts précédents, petit retour sur les paramètres, variables et fonctions :

Remarque : l'habitude est une force de la nature, mais on ne la connais pas souvent bonne.

Pour nous, x est une variable, a, u,.. un paramètre. Qu'en pense la machine ?



f(x) est une fonction, de même que a(x) ou u(x) !

u, qu'il contienne ou non une valeur est considéré comme un paramètre (dérivation par rapport à x…).

Joli, u(x) est bien pris comme il se doit.

Et là aussi ça me plait. Pourquoi ? parce que nous avons le résultat de *f'(5)* alors qu'elle aurait pu nous répondre *(f(5))'*.

La calculatrice semble nous comprendre, elle est programmée selon nos habitudes mathématiques. La difficulté de compréhension se fera sur un travail 'général', avec des fonctions non définies, où ce que nous croyons (de ce qu'elle va faire) n'est pas une certitude, bien que la calculatrice connaisse plutôt bien ses définitions.

Je rappelle que de nombreux exemples sont issus des écrits de Philippe FORTIN et Bernard EGGER.

Rem : les écrans affichés ne correspondent pas forcément à ceux obtenus par votre calculatrice, ils sont parfois plus large, plus haut, avec plus d'indications… c'est normal, j'ai une machine dont l'écran peut s'étirer en largeur ou hauteur, dans des proportions raisonnables. (hé hé !).