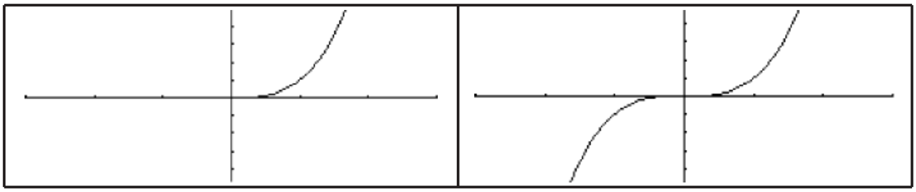
Représentations graphiques et nombres rationnels

Bernard Egger

Quel lien peut-il bien y avoir entre les représentations graphiques sur une calculatrice et les nombres rationnels ? C'est ce que je me propose d'expliquer ici. Mais tout d'abord quelques mots pour préciser comment tout cela a commencé.
Correspondant TICE de mon établissement, peut-être un peu plus intéressé par les calculatrices et les ordinateurs que la moyenne des collègues, je suis quelquefois sollicité pour répondre à des questions rencontrées dans l'utilisation de ces outils.
Durant la précédente année scolaire, un des professeurs de mon lycée est venu me soumettre le problème suivant :
Dans sa classe de terminale S, plusieurs élèves possédaient une TI-89. Lors d'une séance consacrée aux fonctions puissances, il demande à la classe de tracer la courbe représentative de la fonction $x \mapsto x^{\pi}$. Surprise ! Une partie des élèves obtiennent la courbe que l'enseignant attendait (celle de gauche ci-dessous), les autres ont en plus un arc « parasite » pour les x négatifs (celui de droite).



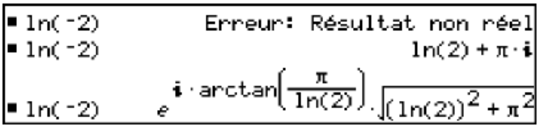
Une même marque de calculatrices, un même modèle retournent deux résultats différents, cela a de quoi perturber l'enseignant le plus placide.
La TI-89 étant une calculatrice flash (c'est-à-dire dont la mise à jour peut se faire par Internet), la première vérification a concerné la version du logiciel mathématique de chaque machine : rien de ce côté, il s'agissait des mêmes versions.

Face à cette situation, j'ai commencé une recherche qui, chemin faisant, m'a permis de découvrir des tas de choses, avec au bout du compte l'impression incommodante d'avoir accouché d'une souris. Néanmoins je ne regrette pas mes efforts qui sur un plan personnel m'ont beaucoup apporté. Cet article est là pour vous faire partager quelques-uns des moments de cette « recherche ».

Fidèle à mes habitudes, refusant par principe d'invoquer le bug ou les contraintes technologiques (et économiques) qui auraient imposé à l'ingénieur des choix peu justifiables, j'ai engagé la recherche de la solution vers des explications mathématiques. Comment aborder le problème ? En se répétant quelques bons principes :

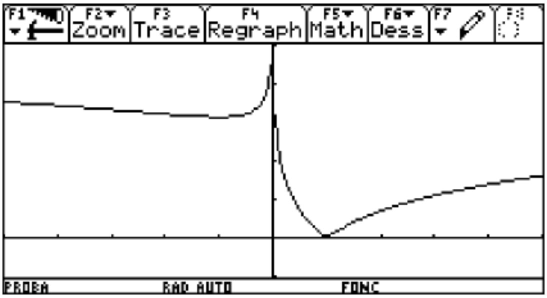
– Dans une calculatrice symbolique (ou un logiciel de calcul formel), un nombre entier, un nombre rationnel, une fraction de π, et les images par une fonction des objets précédents ne sont pas des nombres, mais des symboles qui requièrent un traitement particulier (de nature « grammaticale »).
– Tout nombre à virgule est un nombre décimal ayant 14 chiffres au plus après la virgule (pour la TI-89, la précision pouvant varier suivant la marque et le modèle de la machine).
– Les représentations graphiques n'utilisent que des nombres décimaux. Autrement dit, quand il s'agit de tracer $x \mapsto x^{\pi}$, la calculatrice trace : $x \mapsto x^{3.1415926535898}$.
– Comme tous les outils de calcul formel, la calculatrice travaille avec des nombres complexes. Mais si l'on configure le format complexe avec option : réel, seuls les résultats réels seront affichés.

Pour bien comprendre ce qui se passe pour chacune des trois options du format Complexe, examinons ce que rend la calculatrice pour ln(–2) pour chacune des options possibles.



Si l'on choisit « réel » pour le format complexe, une erreur est retournée, sinon le résultat est affiché sous la forme d'un nombre complexe en écriture algébrique ou exponentielle.
– Avec l'option : réel, la calculatrice ne vérifie pas les calculs intermédiaires. Comme les algébristes italiens, dans leur résolution des équations du troisième degré, elle utilise sans état d'âme des « $\sqrt{-1}$ », c'est-à-dire des complexes, et affiche tout résultat final qui est réel :

– Dans une représentation graphique tout point dont les coordonnées sont réelles est affiché. Ce qui explique cette « étrange » courbe pour la fonction : $x \mapsto |\ln x|$.`



L'écran ci-dessus permet de comprendre ce qui se passe.
Quand $x > 0$, la fonction se ramène bien à ln x.

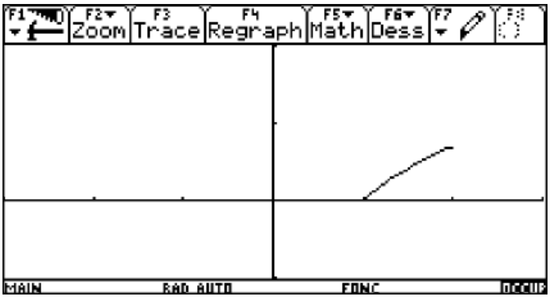
Quand $x < 0$, elle devient : $x \mapsto \sqrt{\ln|x| + \pi^2}$

Il est toutefois possible de préciser à la calculatrice que l'on veut empêcher l'évaluation quand x est négatif à l'aide de l'instruction « sachant que ».
On entre $y1 = \ln(x) \mid x > 0$ et l'on obtient la courbe correcte.

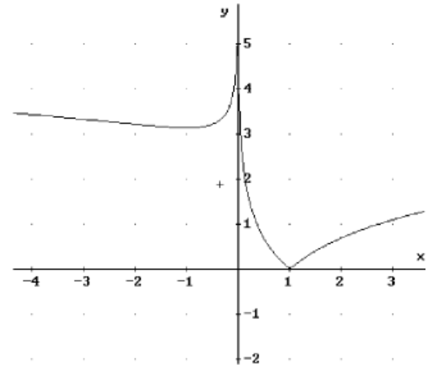
C'est ce principe général (à l'œuvre aussi bien dans les tracés de courbes que dans les résolutions d'équations) : le travail « caché » de la calculatrice dans **C** risque de créer des problèmes, qui a orienté la première partie de ma recherche.

Il est donc préférable d'indiquer à chaque fois sur quel ensemble on travaille : $x > 0$ pour $]0,+\infty[$ ou ($x > 1$ and $x \leq 2$) pour x dans $]1\ ;\ 2]$.
Cela a de quoi satisfaire le professeur de maths que je suis puisqu'il devient nécessaire d'indiquer sur quel ensemble on travaille. Bien entendu je précise ce problème à mes élèves dès le début de l'année.
Par exemple la courbe de $x \mapsto \ln|x|$ sur l'intervalle $]1\ ;\ 2]$ correspondant à :

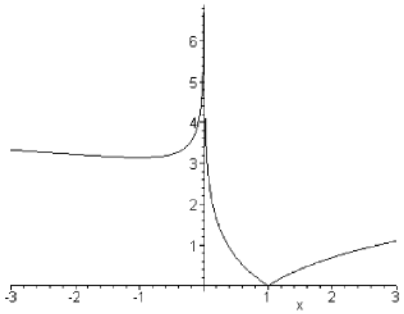
✓y1=ln(|x|)|x > 1 and x ≤ 2



Remarquons que le tracé obtenu pour $x \mapsto \ln|x|$ sur **R** à la page précédente n'est d'ailleurs pas spécifique à la TI-89.



Sur DERIVE 5.0



Sur Maple V R 5

Toutes ces remarques m’ont conduit à conjecturer que le problème pouvait provenir de l’option choisie pour le format complexe.
En choisissant tour à tour chacune des deux options : *réel* ou *rectangulaire*, et en faisant tracer la courbe représentative de $x \mapsto x^\pi$, j’ai bien obtenu les deux graphes différents du tout début de cet article, celui de gauche (donc le plus correct selon notre enseignement) correspondant à l’option *rectangulaire*, l’autre à l’option *réel*.

Cette découverte pour être un premier pas satisfaisant ne m’apportait que bien partiellement la réponse au problème rencontré. Le choix du format complexe influe effectivement sur le tracé, mais pourquoi ? On le comprend bien dans l’exemple de la fonction $x \mapsto \ln|x|$, mais pourquoi également dans la fonction $x \mapsto x^\pi$?
Il restait donc à examiner d’un peu plus près cette fonction, et en particulier pour les valeurs négatives de x.

Comme pour ln(–2), si l’on fait afficher $(-2)^\pi$ suivant les trois options possibles du format complexe, on trouve les résultats ci-dessous :

▪$(-2)^\pi$	Erreur: Résultat non réel
▪$(-2)^\pi$	$2^\pi \cdot \cos(\pi^2) + 2^\pi \cdot \sin(\pi^2) \cdot i$
▪$(-2)^\pi$	$e^{i \cdot \pi^2} \cdot 2^\pi$

Les résultats pour les options *rectangulaire* et *polaire* sont cohérents avec le tracé de la courbe représentative de $x \mapsto x^\pi$ puisque les puissances π-ièmes des nombres négatifs sont des nombres complexes. La transformation est d’ailleurs assez simple à comprendre à l’aide de la forme polaire : $(-2)^\pi = 2^\pi \times (-1)^\pi = 2^\pi \left(e^{i\pi}\right)^\pi = 2^\pi e^{i\pi^2}$.

Pour l’option *réel*, le mystère semble s’épaissir : alors que la calculatrice reconnaît le nombre $(-2)^\pi$ comme un nombre « non réel », elle parvient à calculer les coordonnées du point $(-2\,;(-2)^\pi)$ ce que confirme le calcul en mode approché :

▪$(-2)^\pi$	-8.8249778

C'est donc dans le remplacement de π par 3,1415926535898 que se situait la clef du problème.
Restait à comprendre ce que pouvait être en mode réel la puissance d'un nombre négatif à un exposant décimal…
Et déjà mathématiquement de quoi peut-il s'agir ?
Impossible d'utiliser une définition du type $a^b = \mathrm{e}^{b \ln a}$. Alors comment la calculatrice peut-elle s'en sortir ?
Les premières tentatives sont infructueuses :

▪ $(-2)^{3.786}$	Erreur: Résultat non réel
▪ $(-2)^{2.876549}$	Erreur: Résultat non réel

Il faut donc interroger directement $x^{3.1415926535898}$.

▪ π	3.14159265359
▪ $(-2)^{3.1415926535898}$	$-8\cdot 2^{\frac{244252}{1725033}}$

Pour l'habitué des calculatrices que je suis, cet écran a de quoi surprendre ! J'avais jusqu'alors expliqué aux élèves qu'une calculatrice symbolique posait un problème bien commun en mathématiques : celui de la forme du résultat retourné. Par exemple, dans certaines factorisations la présence de racines carrées incitera la machine à proposer une factorisation plus complète que sans ces racines :

▪ $\text{factor}(x^2-2)$	x^2-2
▪ $\text{factor}(x^2-\sqrt{3})$	$(x+3^{1/4})\cdot(x-3^{1/4})$

Le choix entre l'écriture exacte et l'écriture décimale d'un résultat peut se faire de diverses façons. Il existe une situation particulière pour laquelle la calculatrice détermine elle-même la forme du résultat : c'est quand elle rencontre dans le calcul à effectuer un nombre à virgule.

▪ 2/3 + 3/4	17/12
▪ 2/3 + .75	1.41666666667
▪ résol(2·x = 1 , x)	x = 1/2
▪ résol(2·x = 1. , x)	x = .5

Chacun comprendra ma surprise devant l'écran suivant :

▪ $(-2)^{3.1415926535898}$	$-8\cdot 2^{\frac{244252}{1725033}}$

Le premier étonnement passé, j'ai cherché à comprendre le résultat de la calculatrice. Il semble bien que face à une puissance d'exposant décimal, elle transforme cet exposant en un nombre rationnel, mais quel est exactement ce nombre pour π ?
Passons par les logarithmes pour le savoir :

▪ $\dfrac{\ln\left(-8\cdot 2^{\frac{244252}{1725033}}\cdot -1\right)}{\ln(2)}$	$\dfrac{5419351}{1725033}$

Et il s'agit bien sûr d'une valeur approchée de π :

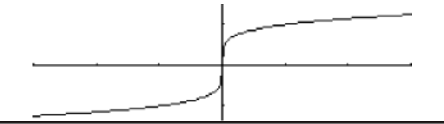
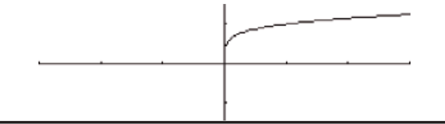
$$\frac{5419351}{1725033}-\pi \qquad 0.$$
$$\frac{5419351}{1725033} \qquad 3.14159265359$$
$$3.1415926535898$$

Que représente cette fraction mystérieuse ? Comment est-elle calculée par la calculatrice ?
Nouveau rebondissement dans l'enquête…

Exposants fractionnaires

L'hypothèse que j'ai faite alors est la suivante : pour calculer la puissance d'un nombre à un exposant décimal, la calculatrice remplace cet exposant décimal par un exposant fractionnaire. C'est donc vers les propriétés des exposants fractionnaires que j'ai dirigé cette nouvelle phase de ma recherche.

Examinons pour commencer la fonction $x \mapsto x^{1/5}$.

	
Voici sa courbe si l'on choisit l'option réel pour format complexe	Et sa courbe si l'on choisit l'option rectangulaire

Comme dans le cas de $x \mapsto x^{\pi}$, le mode « réel » retourne un résultat incorrect selon notre enseignement. Le plus surprenant étant que Derive et Maple sur ce type de courbes retournent des tracés « corrects ».

Pour comprendre le problème il faut examiner le statut d'un nombre de la forme $x^{1/5}$ pour chacune des options.

$$9^{1/5} \qquad 3^{2/5}$$
$$32^{1/5} \qquad 2$$
$$(-32)^{1/5} \qquad -2$$

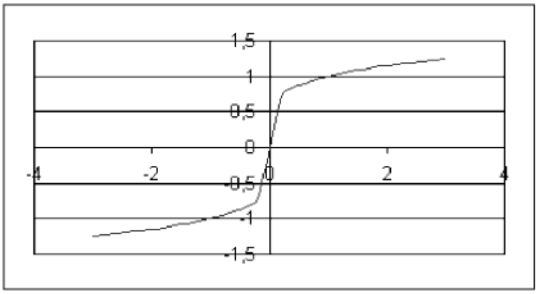
$$9^{1/5} \qquad 3^{2/5}$$
$$32^{1/5} \qquad 2$$
$$(-32)^{1/5} \qquad \frac{\sqrt{5}+1}{2}+\frac{\sqrt{-2\cdot(\sqrt{5}-5)}}{2}\cdot i$$

Est-il besoin de préciser dans quel mode le calcul est fait sur chaque écran ?
En se rappelant les règles que j'ai évoquées au début, le tracé de l'arc « en trop » devient « normal » en option : *réel.*

Pourquoi Derive ou Maple ne se trompent-ils pas ? Parce que par défaut ces logiciels travaillent en option « rectangulaire ».
Par exemple pour DERIVE

$$(-32)^{1/5} = \frac{\sqrt{5}}{2}+\frac{1}{2}+\hat{\imath}\cdot\sqrt{\left(\frac{5}{2}-\frac{\sqrt{5}}{2}\right)}$$

-3	-1,24573094
-2,8	-1,22865968
-2,6	-1,21058328
-2,4	-1,1913579
-2,2	-1,17080491
-2	-1,14869835
-1,8	-1,12474611
-1,6	-1,09856054
-1,4	-1,06961038
-1,2	-1,03713729
-1	-1
-0,8	-0,9563525
-0,6	-0,90288045

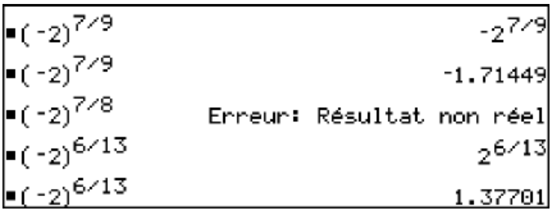


Par contre un tableur comme Excel fera comme la plupart des calculatrices et permettra donc de tracer la courbe représentative de la fonction $x \mapsto x^{1/5}$.
Mais nous reviendrons plus loin sur le tableur qui présente quant à lui certaines particularités.

Les logiciels de géométrie dynamique comme Geoplan ou Cabri refusent de calculer la puissance « non entière » d'un nombre négatif et donc évitent ce problème.
Les tableurs comme les calculatrices intervenant plus nettement dans des secteurs de maths appliquées ont un mode de fonctionnement différent.
La TI-89 complique les choses en faisant coexister les deux modes sur un même outil.

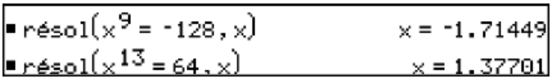
Si en mode réel, la calculatrice peut déterminer $(-2)^{1/5}$, on sait bien qu'elle ne pourra pas trouver une valeur réelle pour $(-2)^{1/2}$ ou $(-2)^{1/4}$.
Il y a donc des rationnels pour lesquels le calcul est possible et d'autres non.

Peut-on donner un critère pour déterminer les rationnels (sous forme irréductible) qui vont rendre le calcul possible et les autres ?
Il y a trois possibilités pour la forme des fractions irréductibles (au signe près) : $\dfrac{2n+1}{2m+1}$, $\dfrac{2n+1}{2m}$, $\dfrac{2n}{2m+1}$.



Un seul cas semble éliminé. On peut comprendre facilement pourquoi.

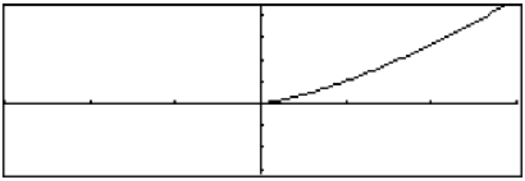
La calculatrice peut écrire $(-2)^{7/9} = (-128)^{1/9}$.
Le nombre qu'elle rend alors est l'unique solution réelle de l'équation $x^9 = -128$.



Même chose évidemment pour $(-2)^{6/13}$ qui devient $64^{1/13}$.

Par contre $(-2)^{7/8}$ devient encore $(-128)^{1/8}$ et l'équation $x^8 = -128$ n'a évidemment pas de solution réelle.
L'approximation rationnelle de π que la calculatrice utilise correspond au premier cas et donc, en mode graphique, les calculs sont effectués pour les nombres négatifs.

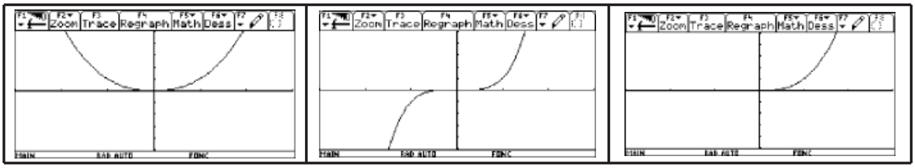
Examinons ce qui se passe pour $x \mapsto x^{\sqrt{2}}$.
En mode réel, on obtient la courbe suivante :



Pas d'arc pour x négatif. L'approximation décimale utilisée pour $\sqrt{2}$ doit certainement s'écrire sous la forme d'une fraction irréductible à dénominateur pair (il est difficile de le savoir pour le moment, les calculs n'étant pas effectués).

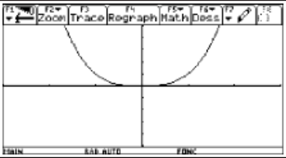
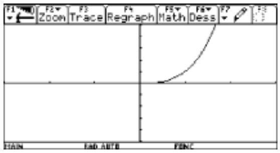
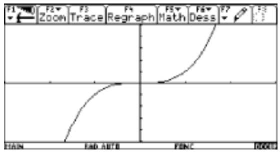
Résumons-nous : **quand on choisit l'option réel pour le format complexe de la TI-89 les puissances de nombres négatifs avec un exposant fractionnaire ne donnent pas d'arc pour les valeurs négatives si le dénominateur de la forme irréductible de cet exposant est pair.**
Quand la calculatrice doit tracer la courbe d'une fonction puissance d'exposant décimal, elle commence par transformer cet exposant en un nombre rationnel irréductible.

Avec ces règles simples on peut prévoir les différents cas possibles.
Voici trois fonctions puissances et trois écrans. Je vous laisse rendre à chacune sa courbe représentative :
a) $x \mapsto x^{523/127}$; b) $x \mapsto x^{532/217}$; c) $x \mapsto x^{71/24}$.



Même si ces conclusions éclairent les « choix » de la calculatrice, il n'empêche que nous restons dans une même incertitude en ce qui concerne les exposants décimaux.

Examinez ces trois exemples pour vous en convaincre :

$x \mapsto x^{3.141592653589}$	$x \mapsto x^{3.141592653587}$	$x \mapsto x^{3.141592653585}$
		

La calculatrice remplace chacun de ces trois exposants par un rationnel dont nous pouvons prévoir la forme, mais dont nous ne savons pour l'instant pas grand chose de plus si ce n'est qu'il ne s'agit pas de $\dfrac{3\,141\,592\,653\,589}{10^{12}}$ ou de $\dfrac{3\,141\,592\,653\,587}{10^{12}}$ ou encore de $\dfrac{3\,141\,592\,653\,585}{10^{12}}$ puisque nous aurions alors trois courbes identiques à la deuxième (ces trois fractions se ramenant évidemment à des fractions irréductibles à dénominateur pair).

Retour à π.

Mais, heureusement, il nous reste π : nous avons vu que la calculatrice a remplacé le nombre décimal 3,141 592 653 598 par $\dfrac{5\,419\,351}{1\,725\,033}$.

Plaçons-nous dans un problème plus large. Comment en mathématiques trouver des approximations rationnelles d'un nombre réel ? Les méthodes sont nombreuses suivant le type de nombres que l'on cherche à approcher, mais il en est une très classique et tout à fait générale : celle des réduites… Rappelons-la brièvement ici en reprenant notre exemple de 3,141 592 653 598.

On a : $3{,}141\,592\,653\,598 = 3 + 0{,}141\,592\,653\,598 = 3 + \dfrac{1}{\dfrac{1}{0{,}141\,592\,653\,598}}$.

Ce qui donne :

$$3{,}141\,592\,653\,598 \approx 3 + \frac{1}{7{,}062\,513\,305\,930\,7}.$$

On a donc $3{,}141\,592\,653\,598 \approx 3 + \dfrac{1}{7}$ et l'on retrouve $3{,}141\,592\,653\,598 \approx \dfrac{22}{7}$.

On peut alors continuer : $3{,}141\,592\,653\,598 \approx 3 + \dfrac{1}{7 + \dfrac{1}{\dfrac{1}{0{,}062\,513\,305\,930\,7}}}$.

Donc $3{,}141\,592\,653\,598 \approx 3 + \cfrac{1}{7 + \cfrac{1}{15{,}995\,944\,067\,74}}$.

Ce qui donne : $3{,}141\,592\,653\,598 \approx 3 + \cfrac{1}{7 + \cfrac{1}{15}}$ ou encore : $3{,}141\,592\,653\,598 \approx \dfrac{333}{106}$.

Et ainsi de suite…

On peut facilement créer un petit programme pour calculer les réduites successives :

```
■ frc(3.1415926535898, 1)          22/7
■ frc(3.1415926535898, 2)          333/106
■ frc(3.1415926535898, 3)          355/113
■ frc(3.1415926535898, 4)          103993/33102
```

Et là, miracle !

```
■ frc(3.1415926535898, 10)         4272943/1360120
■ frc(3.1415926535898, 11)         5419351/1725033
```

Il s'agit donc d'une approximation rationnelle obtenue à partir d'une réduite (ici la onzième).
Mais quelle est donc cette réduite ?
Pour répondre à cette nouvelle question, j'ai sollicité les « compétences » de la calculatrice.

La transformation d'un nombre décimal en nombre rationnel est une opération prévue sur la TI-89 (cela est également vrai dans d'autres modèles de calculatrices, mais avec des limitations…).
Sur la TI-89, elle correspond à l'instruction **exact**, dont le fonctionnement est peu explicité par Texas Instruments.
Examinons son écriture pour quelques valeurs :

```
■ exact(3.1415926535898)           15707963267949/5000000000000
■ exact(3.1415926535898, 10^-1)    22/7
■ exact(3.1415926535898, 10^-2)    22/7
■ exact(3.1415926535898, 10^-3)    22/7
```

Et si l'on continue :

$\text{exact}(3.1415926535898, 10^{-4})$	$\frac{333}{106}$
$\text{exact}(3.1415926535898, 10^{-5})$	$\frac{355}{113}$
$\text{exact}(3.1415926535898, 10^{-10})$	$\frac{208341}{66317}$
$\text{exact}(3.1415926535898, 10^{-13})$	$\frac{5419351}{1725033}$

Il s'agit bien de fractions continues correspondant aux réduites (toutes les réduites ne sont pas utilisées…)

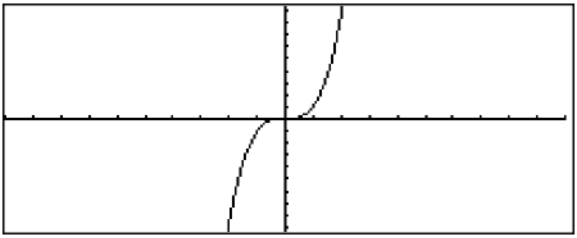
Pour comprendre son fonctionnement, on peut examiner l'écran ci-dessous :

$\text{exact}(3.1415926535898, 10^{-1})$	22/7
3.1415926535898 − 22/7	−.001264
$\text{exact}(3.1415926535898, 10^{-3})$	22/7
$\text{exact}(3.1415926535898, 10^{-4})$	$\frac{333}{106}$
$3.1415926535898 - \frac{333}{106}$	.000083

Comment fonctionne donc cette fonction **exact** ?
Elle fournit des réduites, mais il n'est pas toujours simple de comprendre ses résultats. Peut-on envisager comme hypothèse que dans tous les cas la calculatrice utilise la réduite obtenue par **exact**(nombre,10^{-13}) comme c'est le cas dans notre exemple ?
Une rapide vérification va nous convaincre du contraire.

3.141592653585 → a	3.14159
$\text{exact}(a, 10^{-13})$	$\frac{2813846}{895675}$

Dénominateur impair, numérateur pair : la courbe représentation dans l'option réel doit être celle d'une fonction paire.



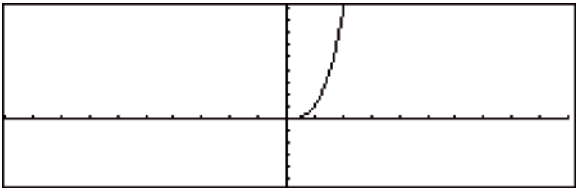
Il n'en est rien.
Ce que nous confirme d'ailleurs l'écran ci-dessous :

$(-2)^{a}$	$-8\cdot 2^{\frac{544860}{3848081}}$
$\dfrac{\ln\left(8\cdot 2^{\frac{544860}{3848081}}\right)}{\ln(2)}$	$\frac{12089103}{3848081}$

Pourtant **exact**(nombre,10^{-13}) semble bien convenir pour les exemples suivants :

```
■ 3.141592653587 → b                3.14159
                                    1980127
■ exact(b, 10^-13)                  -------
                                     630294
```

Et la courbe correspondante (normalement sans arc « négatif » puisque le dénominateur est pair) :



Ou encore :

```
■ 3.141592653589 → c                3.14159
                                    7399478
■ exact(c, 10^-13)                  -------
                                    2355327
    ln((-2)^c)                      7399478
■  -----------                      -------
     ln(2)                          2355327
```

L'écran suivant peut nous permettre de comprendre pourquoi la fonction **exact** ne joue pas toujours le rôle convenable :

```
■ Définir f(x)=x - exact(x, 10^-13)      Fait
■ f(a)                                 -3.E-13
■ f(b)                                 -1.E-13
■ f(c)                                 -1.E-13
```

On peut faire comme hypothèse que la réduite utilisée est celle pour laquelle la précision est inférieure ou égale à 10^{-13}.
Pour « vérifier » cette assertion, on peut faire un programme qui détermine cette réduite. On l'appellera **fcb**.
En voici le listing :

```
:fcb(nb,precis)
:Func
:Local a,chiffre,i,d,n,s
:nb→a
:{exact(partEnt(a))}→chiffre
:0→i
:Loop
:  partDéc(a)→d
:  1/d→a
:  exact(partEnt(a))→n
:  augmente(chiffre,{n})→chiffre
:  ajout(chiffre)→s
:  If abs(s-nb)≤precis
:  Exit
:EndLoop
:Return s
:EndFunc
```

La fonction **fcb** utilise la fonction **ajout**.

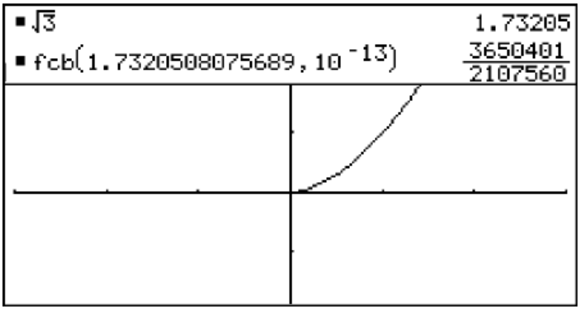
```
:ajout(liste)
:Func
:Local i,n,d
:dim(liste)→d
:liste[d]→n
:For i,d-1,1,-1
:  liste[i]+1/n→n
:EndFor
:Return n
:EndFunc
```

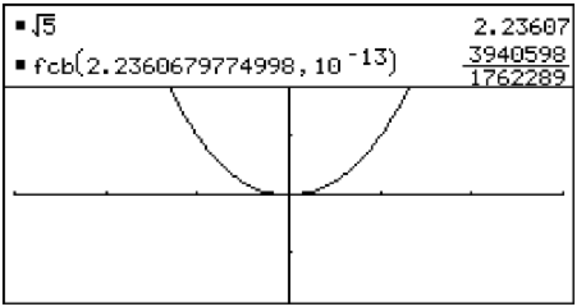
En voilà deux exemples pratiques :

■ fcb(a, 10^{-13})	$\frac{12089103}{3848081}$
■ fcb(b, 10^{-13})	$\frac{1980127}{630294}$
■ fcb(c, 10^{-13})	$\frac{7399478}{2355327}$

Hypothèse sans doute confirmée ! Il est difficile de le prouver, mais on peut assez légitimement l'admettre.

On peut maintenant prévoir les problèmes :

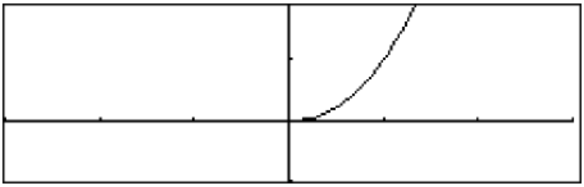




On peut donc comprendre en particulier pourquoi $x \mapsto x^{\pi}$ a l'allure d'une fonction impaire et $x \mapsto x^{\sqrt{5}}$ celle d'une fonction paire.

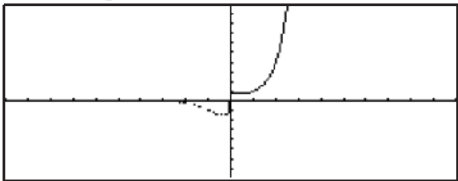
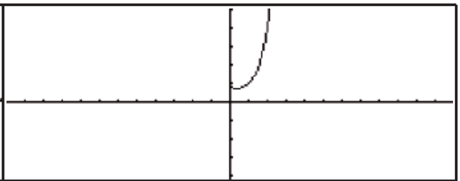
Mais bien entendu, comme pour $|\ln x|$, on peut toujours obtenir la courbe « correcte » dans chaque cas :

✓y1=$x^{\sqrt{5}} \mid x \geq 0$ donne :

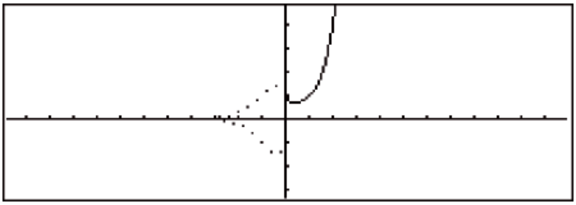


Plus spectaculaire : le cas x^x

Cette année, décidément propice à une réflexion sur les fonctions puissances, m'a amené une autre question encore plus énigmatique.
Un de mes élèves de terminale S est venu me voir avec le problème suivant : travaillant avec un camarade, il s'était proposé d'examiner la fonction $x \mapsto x^x$. Avant de commencer l'étude, comme la majorité des élèves de terminales, ils ont fait tracer la courbe représentative sur leurs calculatrices (des TI-89).
Et là, surprise :

	
Tracé obtenu pour un « zoom » standard	Tracé obtenu pour un « zoom » décimal

Le « zoom » dit standard concerne une fenêtre −10…10 sur chacun des axes, le « zoom » décimal est calculé par rapport au nombre de pixels : il permet qu'un déplacement de 1 pixel sur l'axe des abscisses corresponde à 0,1.
Les élèves ont immédiatement repéré que la différence provenait des « zooms ».
Le premier des deux se replaça donc en « zoom » décimal. Nouvelle surprise :



Que venaient faire ces petits points ?
Ils sont donc venus me signaler le problème.

À ce niveau de l'article on devrait pouvoir comprendre ce qui se passe.

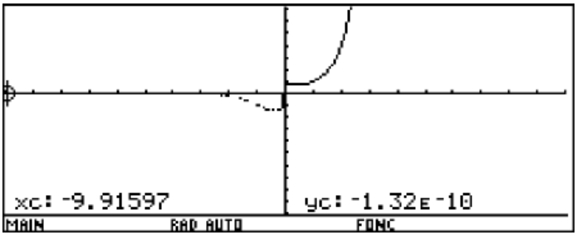
Commençons par le début : pourquoi cette différence suivant le type de repère ?
Il suffit de se rappeler comment une calculatrice trace une courbe.
Elle calcule les images par la fonction de certaines valeurs ; ces valeurs sont déterminées en divisant la longueur de l'intervalle par le nombre de pixels qu'il y a dans la largeur de l'écran. Par exemple sur la TI-92, il y a 238 pixels dans la largeur. Avec un repère dont les abscisses vont de –10 à 10, les deux pixels extrêmes

correspondent à –10 et 10. À partir de –10, on passe d'un pixel au suivant par un déplacement de

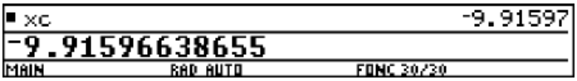
$$\frac{20}{238}=\frac{10}{119}\approx 0.084\,033\,613\,443\,78\,.$$

L'abscisse du premier point après –10 est donc environ :

$$-10+0.084\,033\,613\,443\,78\approx -9.915\,966\,386\,556.$$



Ce que confirme l'écran graphique ci-dessus et l'écran de calcul suivant :

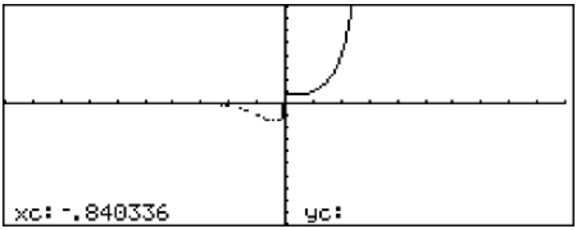


On pourra remarquer que la calculatrice tronque les deux derniers chiffres (elle n'utilise que 12 chiffres au lieu des 14 habituels).

Dès lors pour chaque valeur calculée, par la fonction $x \mapsto x^x$, en format complexe *option réel*, le problème de l'évaluation possible de cette quantité en tant que nombre réel se pose : tout dépend de la réduite de précision inférieure à 10^{-13} de l'abscisse.

Nous devrions pouvoir vérifier cette affirmation. Pour cela reprenons l'écran avec le repère standard et à l'aide de Trace, examinons un point dont nous avons l'abscisse mais pas l'ordonnée (c'est-à-dire pour lequel la quantité x^x n'est pas un réel pour la calculatrice.

Le décimal $-x$ doit correspondre à une réduite dont le dénominateur est pair.



On peut récupérer *xc* dans l'écran de calcul.

Tout semble bien se passer.

Quelle est la réduite en question qui remplace *xc* ?

```
■ fcb(-xc,10^-13)                33138918379/39435312871
```

Déception totale ! Le programme **fcb** ne donne pas un résultat du type attendu.
Que se passe-t-il du côté de la fonction **exact** ?

```
■ exact(-xc,10^-13)              33334755679/39668359258
```

Les deux fractions sont trop proches pour être « honnêtes ». Que peut-il s'être passé ? Si l'on fait l'hypothèse que la fonction **exact** retourne bien une réduite juste, alors c'est la fonction **fcb** qui risque d'être incorrecte.
Le problème est celui de la précision. Les nombres disponibles ont au mieux 14 chiffres après la virgule et dans le courant du calcul bien souvent ils en ont moins. Les erreurs d'arrondi sont la plupart du temps peu gênantes, mais dans certaines situations, comme celle-ci (nous verrons pourquoi un peu plus loin), elles deviennent source d'erreurs. Pour répondre à ce problème il faut améliorer le programme **fcb** pour obtenir une précision supérieure à 10^{-14} et constante tout au long des calculs.

Pour ce faire on peut construire une fonction inverse qui utiliserait la division euclidienne des entiers (qui est en précision « infinie » dans un logiciel de calcul formel), les nombres décimaux étant écrits sous forme de chaîne de caractères.

```
:inverse(nbc,pr)
:Func
:Local pe,pf,pl,d,nb,df,a,inv,dinv,i,j
:dim(nbc)→d
:posTexte(nbc,".")→pl
:If pl=0 Then
:   d+1→pl
:   0→pf
:   0→df
:Else
:   mid(nbc,pl+1,d)→pf
:   dim(pf)→df
:   expr(pf)→pf
:EndIf
:expr(mid(nbc,1,pl-1))→pe
:pe*10^pr+exact(arrondi(pf*10^(pr-df),0)
 )→nb
:10^(2*pr)→a
:(a-reste(a,nb))/nb→inv
:chaîne(inv)→inv
:dim(inv)→dinv
:dinv→i
:Loop
:   If i>pr
 :   Exit
 :   If i=pr Then
 :      "0."&inv→nb
 :      Return nb
 :   EndIf
 :   "0"&inv→inv
 :   i+1→i
 :EndLoop
 :gauche(inv,dinv-pr)&"."&droite(inv,pr)→
 nb
 :Return nb
 :EndFunc
```

Utilisation du programme :

```
■ inverse("2.6574937",10)
                          "0.3762944009"
■ inverse("2.6574937",19)
                   "0.3762944009989562722"
■ inverse("0.165274937",5)      "6.05070"
■ inverse("3",9)             "0.333333333"
```

Nous sommes aussi obligés de définir de nouvelles fonctions partie entière et partie décimale pour ces nouveaux types de « décimaux longs ».

```
:part_ent(n)
:Func
:Local pl
:posTexte(n,".")→pl
:If pl=0 Then
:   Return n
:Else
:   Return expr(gauche(n,pl-1))
:EndIf
:EndFunc
```

```
:part_dec(n)
:Func
:Local pl,dm
:dim(n)→dm
:posTexte(n,".")→pl
:If pl=0 Then
:   Return "0"
:Else
:   Return "0."&droite(n,dm-pl)
:EndIf
:EndFunc
```

On remarquera que le premier programme rend un nombre et le deuxième une chaîne de caractères.

Passons enfin à la nouvelle version du programme **fcb**.
Pour en accélérer l'exécution, ce qui est d'autant plus nécessaire que le programme **inverse** ralentit les calculs, nous utiliserons une propriété de construction des réduites successives : si $r_n = \dfrac{u_n}{v_n}$ est la n-ième réduite, si $r_{n-1} = \dfrac{u_{n-1}}{v_{n-1}}$ est la $(n-1)$-ième réduite et si d est le dénominateur calculé pour la réduite de rang $n+1$, on a :

$$r_{n+1} = \frac{u_n \times d + u_{n-1}}{v_n \times d + v_{n-1}}.$$

On peut alors donner le programme suivant :

```
fcb1(nb,precis,tol)
Func
Local  i,d,n,nombre,a,b,u,v,s
exact(entPrec(nb))→a
1→b
1→u
0→v
a→s
chaîne(nb)→nombre
Loop
 part_dec(nombre)→d
 inverse(d,precis)→nombre
 part_ent(nombre)→n
 (a*n+u)/(b*n+v)→s
 a→u:b→v:numér(s)→a:dénom(s)→b
 If  abs(s-nb)≤tol
 Exit
EndLoop
Return  s
EndFunc
```

```
■ .840336134454 → q                .840336134454
■ fcb1(.840336134454, 20, 10^-13)
                                    32254734879
                                    38383134506
■ fcb1(.840336134454, 15, 10^-13)
                                    32254734979
                                    38383134625
```

On remarque que bien entendu « la réduite calculée » n'est pas la même suivant la précision du calcul choisie. Elle reste stable pour une tolérance de 10^{-13} à partir de 19 chiffres significatifs (avec un nouveau changement à partir de 60 chiffres significatifs).

Quelle est donc la vraie réduite ? Nous sommes dans une situation très instable (quasiment « chaotique » au sens habituel que l'on donne à ce mot).
La valeur sur laquelle nous travaillons correspond au centième pixel à partir de -10 et correspond donc en réalité au rationnel : $\dfrac{100}{119}$.

Mais nous avons vu que la calculatrice ne garde que 12 chiffres. Cette différence est cause de tout.

```
■ .84033613445378 - q                  -2.2E-13
■ fcb1(.84033613445378, 15, 10^-13)      100
                                         119
```

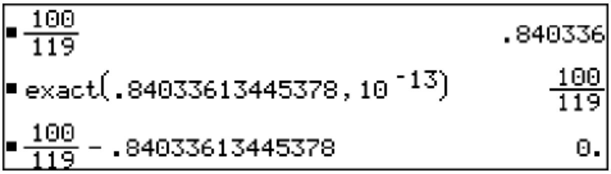
L'écart étant très faible, il occasionne à la sixième réduite un grand dénominateur, très sensible au nombre de chiffres significatifs choisis.
Dans la plupart des situations, il y a une forte stabilité des dénominateurs successifs au moins jusqu'à une tolérance pas trop petite.
Que fait effectivement la calculatrice dans cette situation ? La réduite calculée est-elle « aléatoire » ?
Si l'on demande ce que rend la fonction **exact**, on trouve :

```
■ exact(q, 10^-13)                  33334755679
                                    39668359258
```

La précision de cette « réduite » est suffisante pour qu'elle soit un bon candidat :

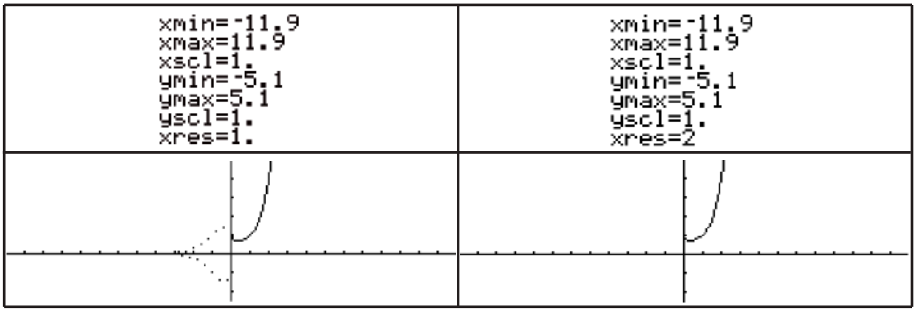
```
■ 33334755679
  ----------- - q                      -1.E-14
  39668359258
```

Si cette procédure est celle choisie par la calculatrice, l'apparition des points devient terriblement aléatoire, car soumise au traitement « chaotique » de l'arrondi…
Nous n'insisterons pas plus sur ce point, mais d'autres tests semblent confirmer que c'est bien ce qui se passe. Il est vrai qu'il s'agit d'un point marginal puisqu'en cours on ne choisira que fort rarement comme exposant une fonction de la variable.
Comme nous l'avons vu c'est parce que la calculatrice ne prend pas tous les chiffres significatifs qu'il y a un problème.



Dès lors il est possible que le calcul qu'elle effectue soit exact, c'est-à-dire avec une fraction continue « convenable », mais que la distorsion vienne de l'instruction Trace.

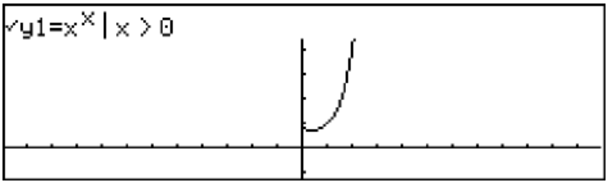
Il nous reste encore une énigme à lever : pourquoi avec la même fenêtre les deux élèves ont-ils obtenu des graphiques différents ?
Examinons cela de plus près. Voici les deux écrans définissant les paramètres d'affichage sur chacune des calculatrices et les courbes obtenues.



La différence est faible : seul le paramètre xres diffère. Que signifie ce paramètre ?
« xres = n » signifie : la calculatrice calcule et trace un point tous les n pixels.
Rappelons que le repère décimal est calculé pour correspondre au nombre de pixels de l'écran : sur la TI-92 dont les écrans ci-dessus sont tirés, il y a 238 pixels sur la largeur et 102 sur la hauteur.
Le premier pixel calculé et affiché est celui correspondant à $-11{,}9$. En résolution 1, le suivant sera : $-11{,}9+\dfrac{(11{,}9-(-11{,}9))}{238}=-11{,}9+0{,}1=-11{,}8$ et ainsi de suite de 0,1 en 0,1 jusqu'à 11,9.
Il est clair que, quelle que soit la précision du calcul, chacun de ces décimaux a une réduite de la forme $\dfrac{k}{10}$ qui se simplifie en $\dfrac{k'}{5}$ pour k pair.
Tous les points correspondant à ces valeurs de k seront tracés. Si l'on n'en voit que peu c'est parce qu'à partir d'un certain moment pour x négatif , x^x est très proche de 0. La fonction Trace peut permettre de visualiser ce phénomène :

En mode de résolution 2, les points calculés ont pour abscisses successives –11,9 bien sûr, puis –11,7 puis –11,5, etc.
Et l'on comprend pourquoi il n'y a plus de points « aberrants ».

Rappelons pour terminer que la TI-89 permet d'obtenir un arc non problématique si on lui indique l'ensemble de définition :



Un petit tour du côté de Derive

Chacun sait sans doute que Derive est un logiciel de calcul formel, qui fonctionne sous Windows depuis la version 4. Il en est à la version 5 dont l'interface est en français. C'est celle-ci que nous utiliserons dans la suite.
Quoiqu'un peu différente, la TI-89 est l'héritière de ce logiciel.
Une différence importante entre Derive et la calculatrice c'est que le logiciel permet de choisir le nombre de chiffres après la virgule. Il n'y a plus la limitation à 14 chiffres :

```
NotationDigits := 100

π

3.1415926535897932384626433832795028841971693993751058209749445923078164062862089986280348~
  2534211706 7
```

Les calculs en mode Exact et Approché diffèrent assez nettement de la calculatrice.

```
Precision := Exact

1.22
                                   61
                                   --
                                   50

Precision := Approximate

61
--
50
                                 1.22
```

Examinons l'écran suivant :

```
Precision := Approximate
NotationDigits := 20
```

$$\frac{100}{119}$$

```
                                         0.84033613445378151260
```

Il n’y a rien de surprenant : la précision d’affichage est de 20 chiffres après la virgule et nous sommes en mode approché.

```
Precision := Approximate
NotationDigits := 20
0.8403361344538
                                         0.84033613445378151260
0.840336134
                                         0.84033613445378151260
0.84033613
                                         0.84033612999994699936
0.840336
                                                   0.840336
```

Cette fois-ci nous avons entré des nombres décimaux et le logiciel ne « respecte pas » nos entrées. Elles sont transformées mais pas n’importe comment.
Le logiciel permet de configurer la précision de calcul qui devient alors la précision d’affichage.
Ce que l’on obtient est alors très instructif :

```
0.8403361344538
PrecisionDigits := 10
                                                0.8403361344
PrecisionDigits := 15
                                             0.840336134453781
PrecisionDigits := 19
                                          0.8403361344537815126
PrecisionDigits := 20
                                         0.84033613445380000000
PrecisionDigits := 21
                                         0.840336134453800000000
```

Ce qui semble clair, c’est que suivant la précision choisie et suivant la valeur entrée, le logiciel procède de la façon suivante : il remplace le décimal par un rationnel (tiens ! voilà à nouveau les réduites) puis calcule à nouveau la valeur approchée de ce rationnel et l’affiche. Dès que la précision de calcul est supérieure ou égale à la précision d’affichage, le nombre est retourné « normalement ».

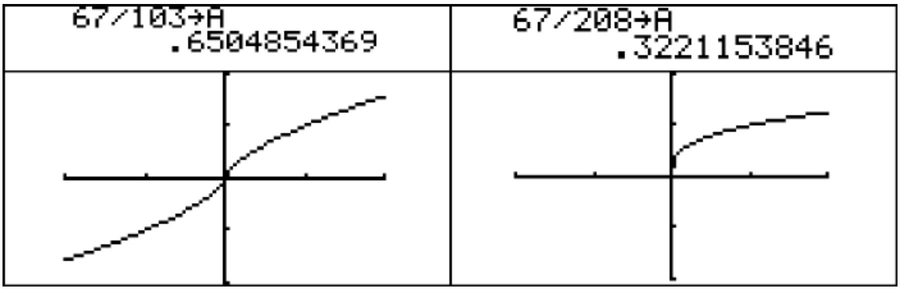
Ce mode de fonctionnement pose parfois de curieux problèmes :
Comme sur la calculatrice et selon le même type de fonctionnement, il existe un format réel pour les complexes, qui force l’affichage d’une solution réelle pour $a^{\frac{1}{n}}$, quand a est un réel (un décimal, bien sûr) négatif et n un entier naturel non nul.
En mode approché, π est remplacé par un décimal, c'est-à-dire par une réduite …
Suivant la précision de calcul choisie, cette réduite change et alors…

```
Precision := Approximate

Branch := Real

PrecisionDigits := 18

     π
(-2)
                                          8.82497782707628762

PrecisionDigits := 19

     π
(-2)
                       -7.966178303885685738 - 3.797398698989756366·î

PrecisionDigits := 20

     π
(-2)
                                        -8.8249778270762876238
```

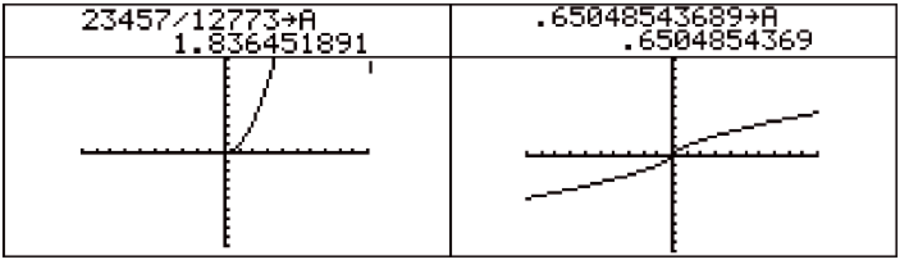
Pour retrouver les chemins utilisés par Derive dans ses calculs, il faudrait écrire un programme un peu plus performant que ceux proposés plus haut. Maintenant que le principe est posé, je laisse à ceux que cela intéresse le soin de s’y atteler.

Et sur d’autres calculatrices

Il ne s’agit pas de faire ici une comparaison de marques. C’est pourquoi nous continuerons avec Texas Instruments et une calculatrice graphique : la TI-83.



Comme l'on peut s'en douter les deux courbes représentent la fonction $x \mapsto x^A$ pour deux valeurs de A.
Ce résultat semble conforme au traitement des puissances rationnelles.
Pourtant ce n'est pas tout à fait aussi simple que cela comme le montrent les exemples suivants :

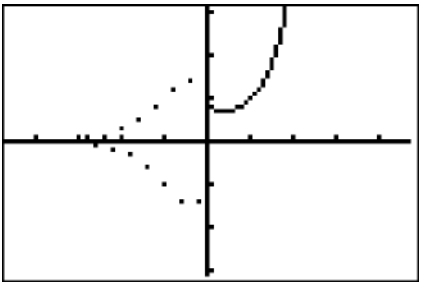


Il existe sur la TI-83 une fonction appelée Frac qui retourne un nombre décimal sous forme fractionnaire pourvu que son dénominateur soit inférieur à 10000.

Examinons les deux nombres précédents avec cette fonction Frac.

.65048543689→A .6504854369 A▸Frac .6504854369	23457/12773→A 1.836451891 A▸Frac 1.836451891

Fort de notre expérience sur la TI-89 ou sur Derive, on peut conjecturer que la calculatrice (qui travaille en mode approché dans tous les cas) remplace donc un rationnel par un décimal, mais pour tracer la fonction puissance, elle revient à un rationnel à partir d'une réduite qui, à partir des erreurs éventuelles d'approximations, revient ou pas au rationnel initial.
La courbe de $x \mapsto x^\pi$ dans le repère « décimal » semble confirmer cette hypothèse (si l'on prend une résolution 1).



Mais sans possibilité d'accéder à des résultats exacts, il est difficile d'aller plus loin.

Les réduites sur Excel

x	x^(1/3)	x^(2/6)	(x^2)^(1/6)	(x^(1/6))^2
-2	-1,259921	-1,259921	1,25992105	#NOMBRE!

On voit bien sur le tableau ci-dessus qu'Excel n'écrit pas pour x négatif : $\left(x^{\frac{1}{m}}\right)^n = x^{\frac{n}{m}}$, ni d'ailleurs que $\left(x^n\right)^{\frac{1}{m}} = x^{\frac{n}{m}}$, ce que fait la TI-89 ou Derive, alors qu'Excel calcule bien la racine cubique d'un nombre négatif.

x	x^(3/5)	x^(1/5)	(x^3)^(1/5)	(x^(1/5))^3	2^(3/5)
-2	#NOMBRE!	-1,1486984	-1,5157166	-1,51571657	1,51571657

Mais de façon générale, Excel ne calcule pas la puissance d'un nombre négatif si l'exposant est fractionnaire et que la fraction sous forme irréductible n'est pas de la forme $\frac{1}{n}$. Par contre, il n'y a aucun problème pour un nombre positif.

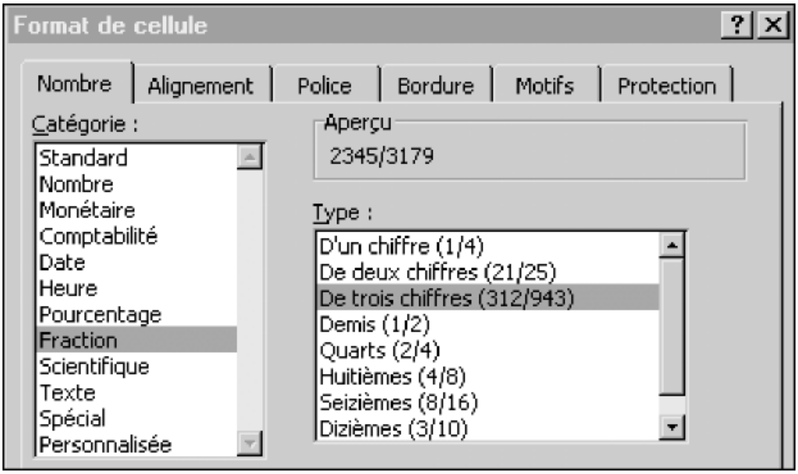
Ce traitement des fonctions puissances élimine tous les problèmes que nous avons rencontrés sur la calculatrice. Pourtant les réduites sont aussi dans Excel.

2345/3179
0,73765335
239/324

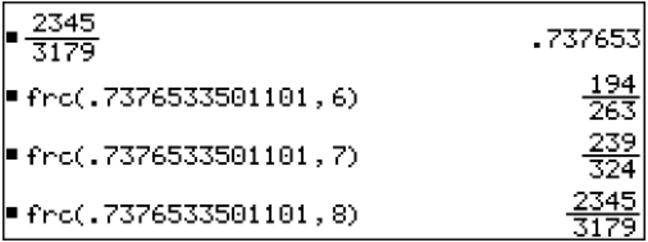
Dans le tableau ci-contre en ligne 2, nous avons le résultat de la division de la ligne 1.

La ligne 3 est ce même résultat mais en demandant un format d'affichage fractionnaire.

Excel permet en effet d'obtenir un tel format pour n'importe quel nombre décimal. Il suffit de choisir le menu Format Cellule.



La fraction $\frac{2\,345}{3\,179}$ étant irréductible, devinez ce qu'est $\frac{239}{324}$?



C’est bien sûr une réduite ; c’est même « la meilleure » ayant trois chiffres au dénominateur.

Voilà un long périple à partir de questions qui au départ me paraissaient bien anodines. Mais il reste encore pas mal de choses à explorer !