TI-92 - PETITS PROGRAMMES

Calcul formel et programmation au lycée - Identité de Bezout - Calcul de pi, de e, de racine de 2.

Sommaire

Site Descartes et les Mathématiques : http://pagesperso-orange.fr/debart/
Document Word : http://www.maths.ac-aix-marseille.fr/debart/doc_dp/ti92prog.doc
Page HTML : http://pagesperso-orange.fr/debart/ti92/ti92programmes.html

Page n° 6, réalisée le 10/12/2000 - mise à jour le 24/8/2005

Ces programmes, pour l'essentiel réalisés au Club Calculatrice, ont surtout un intérêt pédagogique au moment de leur conception. Ils peuvent aussi rendre service à l'occasion d'un cours ou d'un devoir.

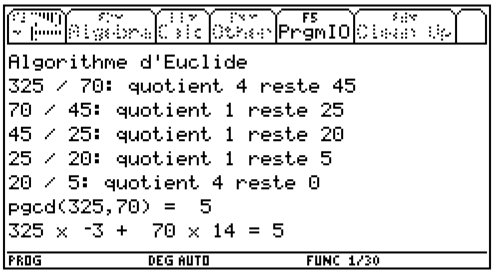
I. PGCD - identité de Bezout

- Les fonctions pgcd(a,b) et ppcm(a,b) {version anglaise gcd(a,b) et lcm(a,b)} de la calculatrice donnent directement le *pgcd* et le *ppcm* de deux nombres *a* et *b*.
- En utilisant le fait que la calculatrice simplifie un quotient d'entiers $\frac{a}{b} = \frac{a'd}{b'd} = \frac{a'}{b'}$.

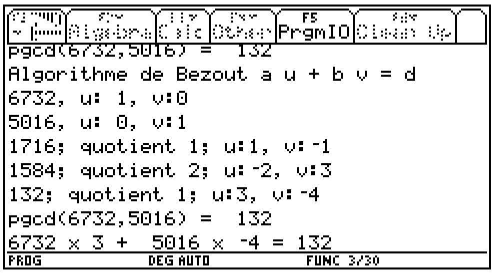
On a donc pgcd(a,b) = $d = \frac{b}{b'}$ et ppcm(a,b) = $\frac{ab}{d} = ab'$.

On peut alors avec la fonction *dénom* (*getnum* en anglais) créer les deux fonctions :

:pgdc(a,b)	:ppmc(a,b)
:Func	:Func
:b/dénom(a/b)	:a*dénom(a/b)
:EndFunc	:EndFunc



- A partir de la fonction pgcd étendu de Jean-Michel Ferrand (Les programmes TI-92, Dunod 1996) voici le programme pgcd(a,b) qui à partir des divisions de l'algorithme d'Euclide permet de trouver le pgcd de deux nombres *a* et *b*, puis donne le résultat de l'identité de Bezout.



Le programme bezout(a,b) reprend le programme pgcd et ajoute un deuxième écran donnant les coefficients *u* et *v* permettant d'exprimer les divers restes.

L'élève devant rédiger l'algorithme devra reprendre les égalités obtenues : (le programme reste à faire !)

$6732 = 6732 \times 1 + 5016 \times 0$ ligne d'initialisation de l'algorithme pour *a*

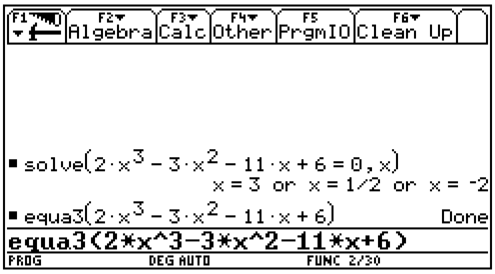
$5016 = 6732 \times 0 + 5016 \times 1$ idem pour *b*

$1716 = 6732 \times 1 + 5106 \times (-1)$ on commence

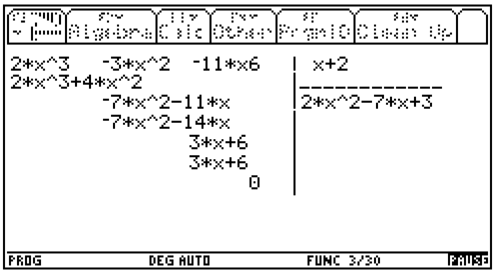
$1584 = 1716 \times (-2) + 5016 = 6732 \times (-2) + 5106 \times 3$ on écrit le reste et on remplace à droite l'ancien reste 1716 par le résultat de la ligne précédente.

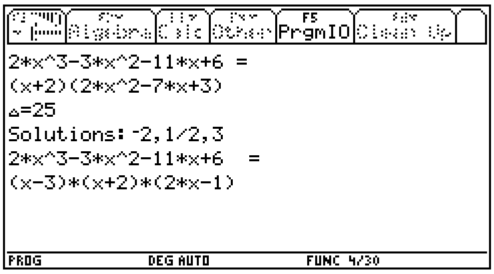
$132 = 1584 \times (-1) + 1716 = 6732 \times 3 + 5106 \times (-4)$ on remplace les deux restes par leurs valeurs trouvées dans les deux lignes précédentes. On trouve enfin l'identité.

II. Équation du troisième degré



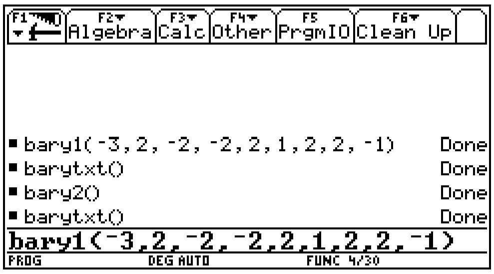
Le programme equa3(*p(x)*) permet d'expliquer comment trouver les solutions d'une équation du troisième degré *p(x)*=0 ayant une solution évidente égale entière dont la valeur absolue n'est pas trop grande (la variable *maxi*=10 dans le programme joint permet de trouver les solutions entre -9 et 9). *Equa3* utilise le sous-programme divEucli(*p(x)*,*q(x)*) qui pose la division euclidienne d'un polynôme du troisième degré par un binôme du premier degré.





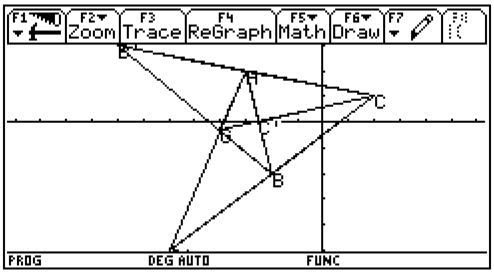
III. Barycentre

Ces programmes calculent et tracent le barycentre G de trois points (A,α), (B,β) et (C,γ). Ils utilisent les trois barycentres partiels : A' barycentre de (B,β) et (C,γ) ;
B' barycentre de (A,α) et (C,γ); enfin C' barycentre de (A,α) et (B,β)



Cet ensemble se compose de trois modules *bary1, bary2, barytxt* et des sous-programmes *calbary*, *affbary*, *wind*, *initwind, nompoint et orthonorm*. Les coordonnées des points sont stockées dans la matrice *b* et les coefficients dans la liste *c*.

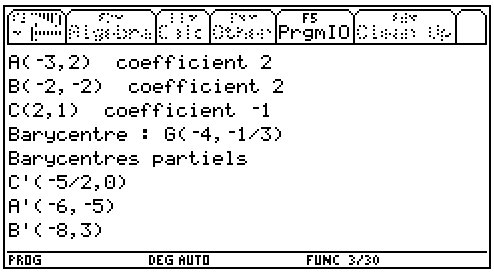
bary1(xa,ya,xb,yb,xa,yc,α,β,γ) utilise les six coordonnées des points A, B et C et les trois coefficients α,β et γ.



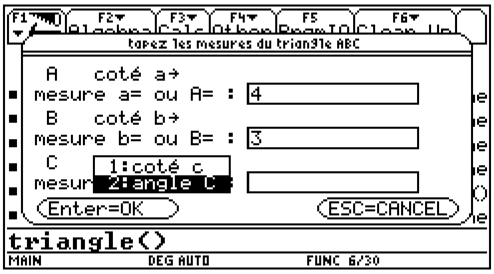
barytxt() permet ensuite de connaître les coordonnées du barycentre G et des barycentres partiels A', B' et C'.

Le programme bary2() permet de saisir les points directement dans l'écran graphique en se déplaçant avec les flèches de directions.

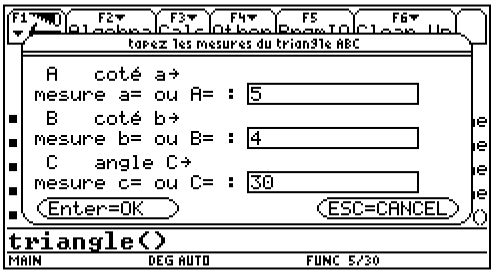
Application : trouver le centre de gravité d'un triangle choisir $\alpha = \beta = \gamma = 1$.

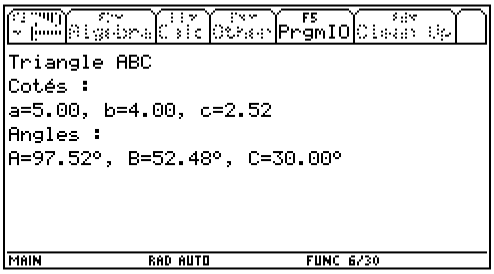


IV. Triangle



Ce programme permet de calculer les côtés et les angles d'un triangle.
Choisir les trois longueurs des côtés, ou deux côtés et un angle ou enfin un côté et deux angles.
Nous obtenons, comme résultats, les trois longueurs des côtés et les trois angles en degré.





Les longueurs *a, b, c* et les angles α, β, γ sont conservés comme variables globales.
Il est possible d'effacer ces variables par programme en supprimant dans le listing l'@ de
:@DelVar a ,b, c, α, β, γ

pour valider la ligne
:DelVar a ,b, c, α, β, γ
(SupVar en version française)

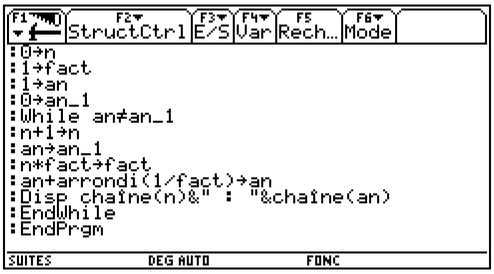
V. Calcul de 600 décimales de *e*

Le calcul de *e* par la série $a_n = \sum_{k=0}^{k=n} \frac{1}{k!}$ converge rapidement vers *e*.

On sait que cette suite a_n est croissante et majorée par *e*.

La suite $b_n = a_n + \frac{1}{n.n!}$ est décroissante et minorée par *e*,

d'où a_n est une valeur approchée par défaut de *e* à $\frac{1}{n.n!}$ près.



En 14 calculs le programme ci-contre donne les 11 premières décimales :

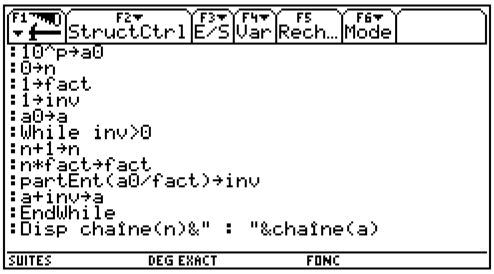
$$e \approx \sum_{k=0}^{k=14} \frac{1}{k!} \approx 2{,}718\ 28\ 18\ 28\ 46$$

En effet, 14 est le premier entier tel *n* que $\frac{1}{n.n!} < 10^{-11}$.

Le programme calcule tant que a_n est différent de a_{n-1} (au sens de la machine).
Pour accélérer les calculs, la factorielle est obtenue par récurrence.

La TI-92 n'utilisant "que" 12 chiffres pour les calculs sur les réels pour trouver plus de décimales, une astuce est d'utiliser des nombres entiers ce qui peut nous permettre de calculer jusqu'à 600 décimales.

En effet sur la TI-92 les calculs sur les entiers sont exacts jusqu'à 10^{614}.



Pour trouver p décimales de e il suffit de multiplier par $10^{p.}$ et de calculer les entiers

$$10^p\, e \approx \sum_{k=0}^{k=n+2} \frac{10^p}{k!} \text{ avec } n.n! > 10^p.$$

Le calcul continue jusqu'à ce que la partie entière de $\frac{10^p}{k!}$ soit nulle.

On obtient ainsi le nombre entier de 614 chiffres :

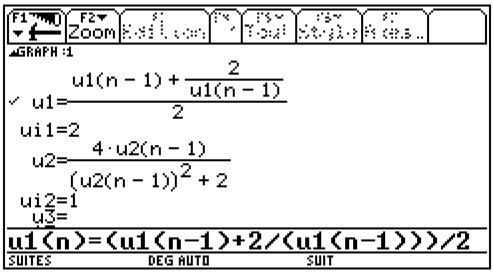
2 71828 18284 59045 23536 02874 71352 66249 77572 47093 69995 95749 66967
62772 40766 30353 54759 45713 82178 52516 64274 27466 39193 20030 59921
81741 35966 29043 57290 03342 95260 59563 07381 32328 62794 34907 63233
82988 07531 95251 01901 15738 34187 93070 21540 89149 93488 41675 09244
76146 06680 82264 80016 84774 11853 74234 54424 37107 53907 77449 92069
55170 27618 38606 26133 13845 83000 75204 49338 26560 29760 67371 13200
70932 87091 27443 74704 72306 96977 20931 01416 92836 81902 55151 08657
46377 21112 52389 78442 50569 53696 77078 54499 69967 94686 44549 05987
93163 68892 30098 79312 77361 78215 42499 92295 76351 48220 82698 95193
66803 31825 28869 39849 64651 05820 93923 98294 88793 32036 25094 43117
29353 40688 510

Le calcul nécessite 299 additions d'entiers dont 296 sont des arrondis de décimaux.

Les 3 dernières décimales sont donc comprises entre 510-214=214 et 510+296=806.

Sachant que $a_{299} < e < b_{299}$ les dernières décimales de e sont donc comprises entre 214 et 807. On a donc calculé e avec 610 décimales : $e \approx 2{,}71828....40688$.

VI. Calcul de 300 décimales de $\sqrt{2}$. Algorithme de Babylone

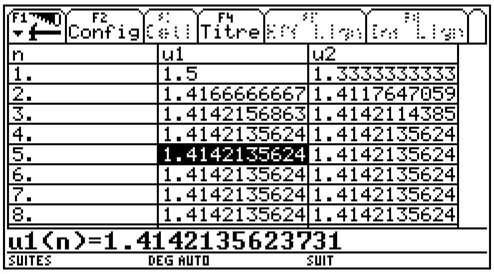


Il y a 4000 ans les Sumériens connaissaient déjà un processus itératif qui permettait de calculer $\sqrt{c}$ à partir de c.

On choisit a_0 (souvent 1) et on calcule la suite (a_n) par la formule de récurrence :

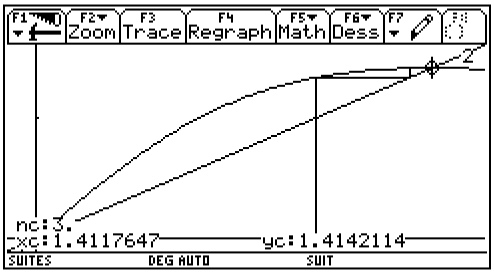
$$a_{n+1} = \frac{1}{2}\left(a_n + \frac{c}{a_n}\right).$$

Cette suite à partir de a_1 est décroissante et minorée par $\sqrt{c}$.



La fonction itérative de Newton correspondant à cette suite est $f(x) = \frac{1}{2}\left(x + \frac{c}{x}\right)$ qui admet comme point fixe $\sqrt{c}$.

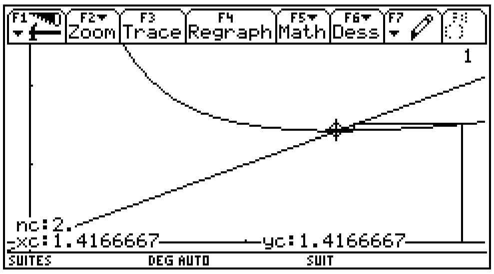
On remarque que la suite $b_n = \frac{c}{a_n}$ est une suite croissante à partir de b_1 et majorée par $\sqrt{c}$.



Le calcul par récurrence de b_n donne :

$$b_{n+1} = \frac{2cb_n}{b_n^2 + c}.$$

La fonction itérative correspondant à cette suite est $g(x) = \frac{2cx}{x^2 + c}$ qui admet comme point fixe $\sqrt{c}$.

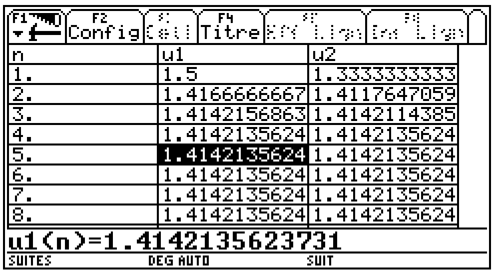


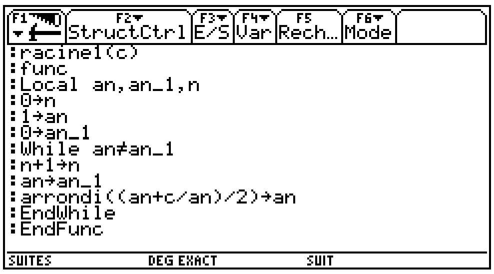
Or $b_1 < b_2 < .... < b_n < \sqrt{c} < a_n < ... < a_1$

avec $b_{n+1} - a_{n+1} < \frac{(b_n - a_n)^2}{4}$.

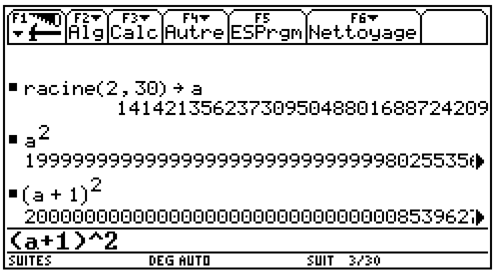
La convergence est donc très rapide.

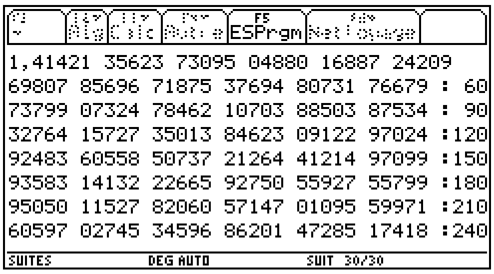
Les calculs se font facilement dans l'écran table ou avec les fonctions *racine* en simple ou multiprécision.



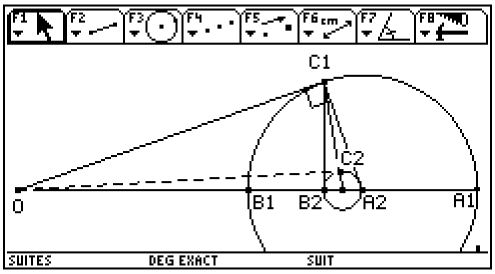


Toutes les décimales sont exactes et on obtient les résultats suivants pour $\sqrt{2}$:





Calcul de la rapidité de convergence



La représentation géométrique permet d'expliquer la convergence vers $\sqrt{2}$ a été faite pour les triangles $A_1B_1C_1$ et $A_2B_2C_2$ avec a_1=2, d'où a_2=1,5 et b_1 =1. La différence de taille de ces triangles donne une idée de la convergence.
Soient A_i et B_i les points d'abscisses a_i et b_i sachant que a_ib_i = c. Le cercle de diamètre A_iB_i est centré en A_{i+1} d'abscisses b_{i+1}. Une des tangentes à ce cercle, issue de O, coupe le cercle en C_i. Dans le triangle rectangle $OA_{i+1}C_i$ le pied de la Hauteur H issue de C_i est tel que :

$$OH.OA_{i+1} = OC_i^2.$$

or $OC_i^2 = OA_i\,OB_i = a_ib_i = c$; puissance d'un point par rapport à un cercle, se calcule aussi avec le théorème de Pythagore.
On a donc OH . $a_{i+1} = c$; OH est donc égal à b_{i+1}. Le pied de la hauteur est donc le point B_{i+1}.
Les triangles rectangles $OA_{i+1}C_i$ et $A_{i+1}B_{i+1}C_i$ sont semblables donc :
$\frac{A_{i+1}B_{i+1}}{A_{i+1}C_i} = \frac{A_{i+1}C_i}{OA_{i+1}}$ d'où $A_{i+1}B_{i+1} = \frac{A_{i+1}C_i^2}{OA_{i+1}} = (\frac{A_iB_i}{2})^2\frac{1}{OA_{i+1}}$ en remplaçant le rayon $A_{i+1}C_i$ par le diamètre A_iB_i. Or $OA_{i+1} < \sqrt{c}$ on en déduit $A_{i+1}B_{i+1} < \frac{A_iB_i^2}{4}\frac{1}{\sqrt{c}} < \frac{A_iB_i^2}{4\sqrt{c}}$

Pour c > 1 on a $a_{i+1} - b_{i+1} < \frac{(a_i - b_i)^2}{4}$.
Si à un étape du calcul on a une valeur approchée à 10^{-n} près, à l'étape suivante on aura un encadrement inférieur à 10^{-2n}. Ainsi à chaque itération on double le nombre de décimales justes. La convergence est donc très rapide : le calcul de 100 décimales de $\sqrt{2}$ se fait avec 8 itérations à partir de a_1 = 2 et en seulement 3 itérations si l'on part de la valeur approchée a_1 = 1.41421356237 calculée par la machine.

VII. Calcul de 600 décimales de π

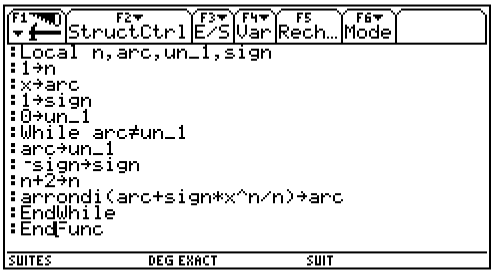
De nombreux calculs utilisent la fonction arctan réciproque de tangente sachant que

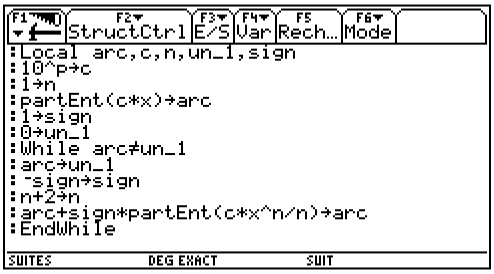
$$\arctan(1) = \frac{\pi}{4} \text{ et que } \arctan(x) = x - \frac{x^3}{3} + \frac{x^5}{5} - \dots .$$

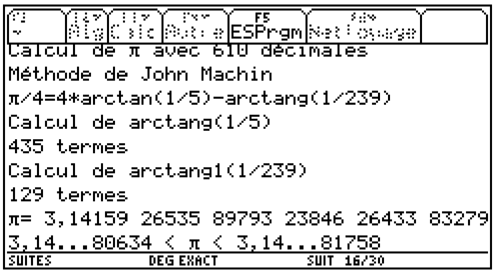
La formule de John Machin :

$$\frac{\pi}{4} = 4\arctan(\frac{1}{5}) - \arctan(\frac{1}{239})$$

permit en 1706 le premier calcul de 100 décimales.
Il suffit de programmer la fonction permettant de calculer arctan :







On obtient donc 607 décimales de π exactes (en 25 minutes de calculs) !

3,14159 26535 89793 23846 26433 83279 50288 41971 69399 37510 58209 74944
59230 78164 06286 20899 86280 34825 34211 70679 82148 08651 32823 06647
09384 46095 50582 23172 53594 08128 48111 74502 84102 70193 85211 05559
64462 29489 54930 38196 44288 10975 66593 34461 28475 64823 37867 83165
27120 19091 45648 56692 34603 48610 45432 66482 13393 60726 02491 41273
72458 70066 06315 58817 48815 20920 96282 92540 91715 36436 78925 90360
01133 05305 48820 46652 13841 46951 94151 16094 33057 27036 57595 91953
09218 61173 81932 61179 31051 18548 07446 23799 62749 56735 18857 52724
89122 79381 83011 94912 98336 73362 44065 66430 86021 39494 63952 24737
19070 21798 60943 70277 05392 17176 29317 67523 84674 81846 76694 05132
00056 81