De bonnes résolutions (3e et 4e degrés) sur TI 57

Résolution d'équations
du troisième degré (TI 57)
Auteur Jean Rosset

00	33 2	RCL 2
01	32 4	STO 4
02	75	+
03	33 0	RCL 0
04	55	x
05	23	x^2
06	34 4	SUM 4
07	33 1	RCL 1
08	55	x
09	34 4	SUM 4
10	02	2
11	75	+
12	33 0	RCL 0
13	39 4	2nd Prd 4
14	23	x^2
15	55	x
16	33 3	RCL 3
17	34 4	SUM 4
18	03	3
19	85	=
20	-39 4	INV Prd 4
21	33 4	RCL 4
22	-34 0	INV SUM 0
23	40	2nd **Ix**I
24	76	x ⩾ t ?
25	71	RST
26	33 0	RCL 0
27	81	R/S
28	34 1	SUM 1
29	-39 3	INV Prd 3
30	02	2
31	84	+/-
32	-39 1	INV Prd 1
33	33 1	RCL 1
34	32 5	STO 5
35	23	x^2
36	34 3	SUM 3
37	33 3	RCL 3
38	24	$\sqrt{x}$
39	34 1	SUM 1
40	-34 5	INV SUM 5
41	33 5	RCL 5
42	81	R/S
43	86 1	2nd Lbl 1
44	33 1	RCL 1
45	81	R/S
46	33 3	RCL 3
47	84	+/-
48	24	$\sqrt{x}$
49	81	R/S

On croit souvent qu'avec ses cinquante pas de programme, la TI 57 ne peut résoudre que des équations du deuxième degré. Il n'en est rien.

■ Voici deux programmes dont l'un calcule les racines réelles ou complexes d'une équation du troisième degré et l'autre toutes les racines réelles d'une équation du quatrième degré.

Les équations de la forme $x^3 + a_1 x^2 + a_2 x + a_3 = 0$ possèdent toujours au moins une racine réelle que l'on peut calculer par la méthode de Newton. Partant d'une valeur approchée x_0 de cette racine, on calcule $x_1 = x_0 - p(x_0)/p'(x_0)$, avec $p(x) = x^3 + a_1 x^2 + a_2 x + a_3$; x_1 est une meilleure valeur approchée de la

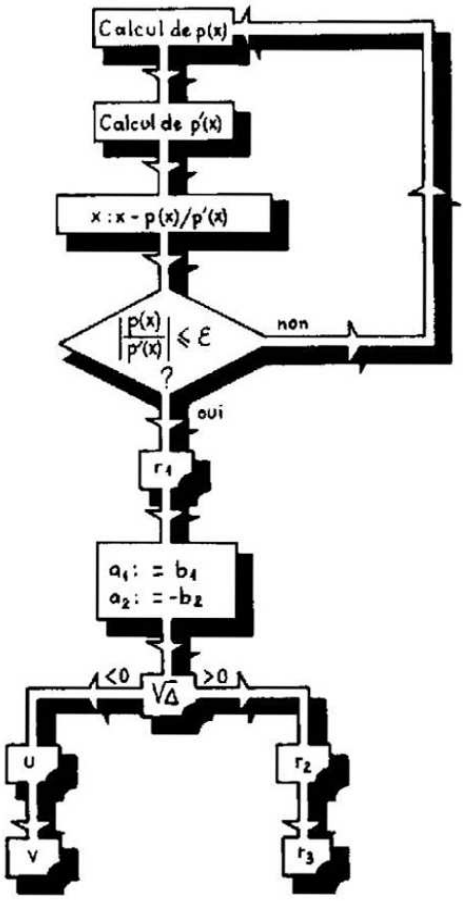
racine. Le processus est itératif et la suite $x_{n+1} = x_n - p(x_n)/p'(x_n)$ tend vers l'une des racines de l'équation $p(x) = 0$. On arrête l'itération dès que $|p(x_n)/p'(x_n)|$ est inférieur à ε, ε étant un nombre positif « petit » que l'on s'est donné à l'avance ($\varepsilon = 10^{-6}$ ou 10^{-8} par exemple).

Si r_1 est cette racine, le polynôme $p(x)$ peut s'écrire $p(x) = (x - r_1)(x^2 + b_1 x + b_2)$, avec $b_1 = a_1 + r_1$ et

Résolution d'équations
du quatrième degré (TI 57)
Auteur Jean Rosset

00	33 0	RCL 0
01	55	x
02	33 5	RCL 5
03	75	+
04	32 6	STO 6
05	33 1	RCL 1
06	61 9	SBR 9
07	33 2	RCL 2
08	61 9	SBR 9
09	33 3	RCL 3
10	85	=
11	34 6	SUM 6
12	55	x
13	33 5	RCL 5
14	75	+
15	33 4	RCL 4
16	85	=
17	38 6	2nd Exc 6
18	-39 6	INV Prd 6
19	33 6	RCL 6
20	-34 5	INV SUM 5
21	40	2nd **Ix**I
22	76	2nd x ⩾ t ?
23	71	RST
24	33 5	RCL 5
25	81	R/S
26	84	+/-
27	-39 4	INV Prd 4
28	33 4	RCL 4
29	38 3	2nd Exc 3
30	-34 3	INV SUM 3
31	33 5	RCL 5
32	-39 3	INV Prd 3
33	00	0
34	38 0	2nd Exc 0
35	39 5	2nd Prd 5
36	38 1	2nd Exc 1
37	32 2	STO 2
38	33 5	RCL 5
39	34 2	SUM 2
40	71	RST
41	86 9	2nd Lbl 9
42	85	=
43	34 6	SUM 6
44	55	x
45	33 5	RCL 5
46	39 6	2nd Prd 6
47	75	+
48	-61	INV SBR

◀ Fig. 1



$b_2 = -a_3/r_1$. Pour trouver les deux racines réelles ou complexes restant à calculer, il suffit de résoudre l'équation $x^2 + b_1 x + b_2 = 0$. Si le discriminant de cette équation est positif, nous trouverons deux racines réelles r_2 et r_3 ; s'il est négatif, nous aurons deux racines complexes conjugées de la forme $r_{2,3} = u \pm iv$.

Le programme se déroule conformément à l'organigramme de la fig. 1. Il est conçu de telle manière que si les trois racines sont réelles, elles sont affichées aux arrêts successifs (r_1, puis r_2 et r_3). Si en revanche il n'y a qu'une seule racine réelle, la machine, après l'avoir obtenue, essaie de calculer la racine carrée du discriminant de l'équation du second degré ; comme ce discriminant est négatif, l'affichage se met à clignoter. On fait alors CLR et SBR 1 : affichage de u, puis on relance le programme en pressant sur R/S et v est affiché à l'arrêt final.

Seule la mémoire n° 6 n'est pas utilisée : M_0 contient x_i, les coefficients a_1, a_2 et a_3 sont respectivement en M_1, M_2 et M_3 ; les registres 4 et 5 sont des registres de travail et t enfin, c'est-à-dire M_7, le registre de test, contient ε .

Notre premier exemple consistera à résoudre l'équation $x^3 - 6x^2 + 11x - 6 = 0$ qui se factorise en $(x - 1)(x - 2)(x - 3) = 0$. Après avoir chargé les trois coefficients dans les mémoires M_1, M_2 et M_3 ainsi que la quantité $\varepsilon = 10^{-6}$ en M_7, on lance le programme en pressant sur RST puis sur R/S, et l'on obtient successivement $r_1 = 1$, $r_2 = 2$ et $r_3 = 3$. Le temps de calcul est de vingt secondes environ. Notons que, puisque nous avons pris $x_0 = 0$ comme valeur de départ, la première racine calculée est la racine de plus faible module.

Deuxième exemple : résolvons l'équation $x^3 - 4x^2 + 8x - 8 = 0$ qui admet $x = 2$ comme seule racine réelle. Après avoir effacé le contenu des mémoires M_0 à M_6, on entre les trois coefficients de la nouvelle équation (-4,8 et -8) en M_1, M_2 et M_3 et on lance le programme. Au premier arrêt, on obtient bien $r_1 = 2$. Au deuxième arrêt, l'affichage est clignotant (on a tenté d'extraire la racine carrée d'un nombre négatif au pas 38) : cela nous indique que l'équation n'a qu'une seule racine réelle. On relance alors par CLR et SBR 1 : affichage de u qui vaut 1, et une pression sur R/S nous donne v = 1,7320508 (soit une valeur proche de $\sqrt{3}$). Ainsi $r_{2,3} = 1 \pm i\sqrt{3}$.

Le quatrième degré

L'équation $x^4 + a_1 x^3 + a_2 x^2 + a_3 x + a_4 = 0$ possède zéro, deux ou quatre racines réelles et le programme que nous allons examiner nous les indiquera (si elles existent...).

Ici aussi, nous utiliserons la méthode de Newton. Le calcul de p (x) et de p'(x) se fera selon la méthode de Hörner :

$p(x) = x + a_1)\,x + a_2)\,x + a_3)\,x + a_4$

b_1

b_2

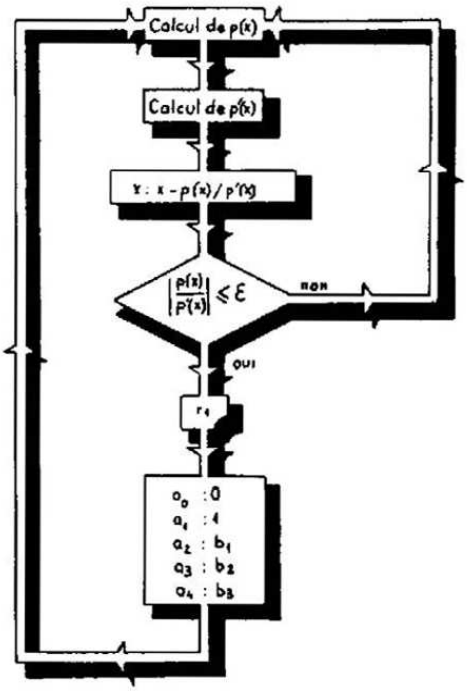
b_3

et $p'(x) = x + b_1)\,x + b_2)\,x + b_3$

Cet algorithme est très commode car il nous permet de former progressivement p'(x) dans la mémoire M_6, au fur et à mesure que nous avançons dans le calcul de p(x) aux pas 0 à 16 du programme.

Une fois la première racine r_1 calculée, il reste à résoudre l'équation du troisième degré $p_1(x) = 0$, avec $p_1(x) = p(x)/(x - r_1)$, soit $x^3 + b_1 x^2 + b_2 x + b_3$. Les b_i n'ayant pas été conservés faute de mémoires disponibles,

Fig. 2 ▶



nous les recalculerons à l'aide des a_i et de r_1 :

- $b_3 = -a_4/r_1$
- $b_2 = (b_3 - a_3)/r_1$
- $b_1 = a_1 + a_0 r$

En ce qui concerne l'utilisation des différents registres de données, b_3 occupera la place de a_4, b_2 celle de a_3, b_1 celle de a_2, 1 celle de a_1 et, bien sûr, 0 celle de a_0 (contenu de M_0). On réalise ainsi la « déflation » du polynôme de départ (d'un polynôme du quatrième degré, on passe à un polynôme du troisième degré), et l'on continue. On trouvera l'organigramme correspondant à la fig. 2.

A titre d'exemple, nous choisirons l'équation $x^4 - 10x^3 + 35x^2 - 50x + 24 = 0$, qui se factorise en $(x - 1)(x - 2)(x - 3)(x - 4) = 0$. Après avoir introduit les différents a_i dans les mémoires M_0 à M_4 et la quantité $\varepsilon = 10^{-8}$ en M_7, on lance le calcul avec RST R/S. Au premier arrêt, on a $r_1 = 1$; au deuxième arrêt $r_2 = 2$, puis $r_3 = 3$ et enfin $r_4 = 4$ (temps total du calcul : environ une minute vingt).

Ces deux méthodes sont sûres et précises, mais on doit signaler qu'il existe d'autres méthodes de résolution des équations du quatrième degré. Certaines d'entre elles sont parfaitement exploitables sur la TI 57 ; je pense en particulier à la méthode dite « des quotients-différences » qui est moins gourmande en pas de programme.

Et maintenant, à quand la résolution d'équation du cinquième et du sixième degrés sur la TI 57 ? (On peut bien rêver, non ?)

Jean Rosset