5/10.4.4

Graphisme sur variations sinusoïdales

Comme vous avez pu le remarquer dans cette partie, les graphismes harmonieux sont difficilement exploitables sans l'utilisation des mathématiques et de leurs équations.

Nous allons dans ce chapitre vous initier aux courbes qu'il est possible de tracer sous forme paramétrique à l'aide des fonctions sinus et cosinus.

La variété d'équations sous formes sinusoïdales étant à la mesure de l'imagination humaine, nous nous arrêterons dans les différents paragraphes à certaines équations, et nous nous attacherons plutôt à vous donner l'envie d'aller plus loin, en vous apportant la réflexion préliminaire à tout tracé. Le lecteur profane relira avec profit les chapitres 13/2.3 et 13/2.4.

Equations les plus couramment utilisées :

$$COS^2 (x) + SIN^2 (x) = 1$$
$$COS (-x) = COS (x)$$
$$SIN (-x) = - SIN (x)$$
$$COS (x + y) = COS (x) * COS (y) - SIN (x) * SIN (y)$$
$$SIN (x+ y) = SIN (x) * COS (y) + COS (x) * SIN (y)$$

VARIATIONS SUR SINUS ET COSINUS

Nous allons nous intéresser à une formule générale d'équation paramétrique, qui à un premier abord semble ardue, mais, nous le verrons sera d'une étude relativement simple, et permettra le tracé de toute une variété de courbes.

UN ALGORITHME GÉNÉRAL

Voyons d'abord l'algorithme général que nous donnerons à nos programmes pour tracer nos courbes paramétriques.

Il nous faudra d'abord entrer les 8 valeurs constantes, puis effectuer les calculs correspondants à différentes valeurs du paramètre t.

Après chaque calcul, on procèdera à l'affichage du point correspondant.

– Début
  – Entrer les valeurs des constantes
  – POUR t DE valeurinitiale A valeurfinale
    – Calculer l'abscisse x
    – Calculer l'ordonnée y
    – PROCEDURE AFFICHAGE
  – FINPOUR
– FIN

Vous pourrez remarquer que cet algorithme est relativement simple, sauf pour la procédure d'affichage qui va demander une certaine réflexion sur les valeurs que pourront atteindre les coordonnées x et y.

La procédure d'affichage

Afin d'afficher nos courbes, nous allons utiliser l'écran en MODE 2, c'est-à-dire dans sa résolution la plus grande, pour afficher le maximum de pixels.

Dans cette résolution, le nombre de points affichables et égale à 640 x 400.

En étudiant un peu plus notre équation paramétrique générale, les fonctions SIN et COS nous génèrerons des valeurs comprises entre - 1 et +1, d'où certaines valeurs de x et y comprises entre un minima négatif et un maxima positif.

Il est donc logique de prendre pour origine un point central de l'écran, par exemple le point de coordonnées : x = 320 et y = 200.

Les points que nous afficherons à partir de cette origine ne devrons ensuite pas dépasser les limites de l'écran, ni être trop près du centre si l'on veut obtenir des courbes intéressantes à regarder.

Nous utiliserons ainsi un facteur multiplicatif d'agrandissement tel que l'écran quasi complet soit utilisé. Ce facteur multiplicatif sera calculé en fonction des valeurs minimales et maximales de x et y, et dépendra ainsi des 4 paramètres a1, b1, c1 et d1.

En x, le minimum sera atteint quand x sera égal à - a1 - b1, le maximum pour a1 + b1.

De même en y, nous aurons pour minimum - c1 - d1, et pour maximum c1 + d1.

Le rapport de proportionnalité pourra être ainsi :

100 / (c1 + d1) en x
et **300 / (c1 + d1)** en y.

Toute valeur inférieure pourra convenir, et nous vous conseillerons en fait de prendre la plus petite de ces valeurs en tant que rapport commun, afin de garder une courbe représentée dans un repère orthonormé, pour éviter de l'étirer selon l'une des coordonnées.

Le programme

Certains d'entre vous se sont peut-être déjà lancé sur leur Amstrad pour appliquer l'algorithme précédemment décrit. Qu'ils ne se précipitent pas, nous allons ici leur fournir un programme plus complet.

Ce programme proposera par menu de tracer une courbe, ou éventuellement de revoir une courbe précédemment sauvegardée sur disquette.

En effet, lors du tracé d'une courbe, il vous sera possible de la sauvegarder. Nous avons même ajouté un choix d'impression en « hard-copy » sur imprimante compatible Epson (la série DMP 2000 en fait partie).

En résumé, voici les différentes possibilités du programme :

• *Choix 1 du menu :* tracé de courbes par entrée des huit paramètres.

Pendant ce tracé, vous aurez trois possibilités :

– un appui sur la touche **I** pour lancer l'impression sur l'imprimante ;

– un appui sur la touche **S** pour sauvegarder le tracé sur disquette. Il faudra fournir un nom de huit caractères maximum. Le programme se chargera de créer deux fichiers en ajoutant l'extension **.PAR** pour le fichier paramètre, et l'extension **.PIC** pour l'écran graphique comportant uniquement la courbe. Il vous sera, par exemple, possible d'utiliser cette courbe dans une application personnelle ultérieure ;

– un appui sur la touche **F** met fin au tracé et renvoit au menu.

• *Choix 2 du menu :* lecture d'une courbe préalablement sauvegardée sur disquette. Les paramètres sont de plus affichés, pour éventuellement reprendre une étude sur une série de courbes aux paramètres proches.

A partir de ce cahier des charges, nous pouvons vous présenter le programme ci-après listé :

```
10 REM ********************************
20 REM *                              *
30 REM * TRACE DE COURBES D'EQUATIONS *
40 REM * PARAMETRIQUES SOUS LA FORME  *
50 REM * X= A1 COS(A2 t)+B1 SIN(B2 t) *
60 REM * Y= C1 COS(C2 t)+D1 SIN(D2 t) *
70 REM *                              *
80 REM ********************************
90 REM
100 MODE 2
110 PRINT "Nous vous proposons de tracer
 les courbes d'equations parametriques"
120 PRINT "ayant les formes suivantes:"
130 PRINT
140 PRINT
150 PRINT "a1 * COS(a2 * t) + b1 * SIN (
b2 * t)
160 PRINT "c1 * COS(c2 * t) + d1 * SIN (
d2 * t)"
170 GOSUB 1580
180 LOCATE 1,10
190 FOR i = 0 TO 8
200 LOCATE 1,10+i
210   PRINT STRING$(80,CHR$(32))
220 NEXT i
230 LOCATE 1,10
240 PRINT "Veuillez entrer les different
s parametres ci-dessous:"
250 PRINT
260 INPUT "terme a1 ";a1
270 INPUT "terme a2 ";a2
280 INPUT "terme b1 ";b1
290 INPUT "terme b2 ";b2
300 INPUT "terme c1 ";c1
310 INPUT "terme c2 ";c2
320 INPUT "terme d1 ";d1
330 INPUT "terme d2 ";d2
340 REM
350 REM RAPPEL DE L'EQUATION PROPOSEE
360 REM
370 MODE 2
380 GOSUB 1400
390 REM
400 REM CALCUL DES RAPPORT DE
410 REM PROPORTIONALITE POUR
420 REM  L'OCCUPATION ECRAN
430 REM
440 rapportx=ABS(a1)+ABS(b1)
450 rapporty=ABS(c1)+ABS(d1)
460 rapport = MAX(rapportx,rapporty)
```

```
470 REM
480 REM ORIGINE DE L'ECRAN GRAPHIQUE
490 REM
500 ORIGIN 320,220
510 REM
520 REM AFFICHAGE PREMIER POINT
530 REM
540 t=0
550 x = a1 * COS(a2 * t) + b1 * SIN (b2
* t)
560 y = c1 * COS(c2 * t) + d1 * SIN (d2
* t)
570 coordonneex = x * 170 / rapport
580 coordonneey = y * 170 / rapport
590 PLOT coordonneex,coordonneey
600 REM
610 REM CALCUL COMPLET DE LA COURBE
620 REM PAR BOUCLE SUR t
630 REM
640 WHILE 1
650     x = a1 * COS(a2 * t) + b1 * SIN (
b2 * t)
660     y = c1 * COS(c2 * t) + d1 * SIN (
d2 * t)
670     DRAW x/rapport*170,y/rapport*170
680         REM incrementation de t par
690         REM pas fixe mais modifiable
700     t=t+0.01
710         REM test touche
720     a$=INKEY$
730     IF a$ <> INKEY$ THEN GOSUB 790
740         REM
750 WEND
760 REM
770 REM VERIFICATION TOUCHE FRAPPEE
780 REM
790 a = 0
800 IF a$ = "I" OR a$ = "i" THEN a = 1
810 IF a$ = "S" OR a$ = "s" THEN a = 2
820 IF a$ = "F" OR a$ = "f" THEN RUN
830 ON a GOSUB 880,1190
840 RETURN
850 REM
860 REM COPY GRAPHIQUE SUR IMPRIMANTE
870 REM
880 DATA cd,ba,bb,cd,e7,bb,32,bf,a0,cd
890 DATA 6b,a0,21,8f,01,22,c0,a0,11,00
900 DATA 00,3e,06,32,c2,a0,cd,7b,a0,0e
910 DATA 00,3a,c2,a0,47,e5,d5,c5,cd,f0
920 DATA bb.c1,d1,21,bf,a0,be,e1,37,20
```

```
930 DATA 01,a7,cb,11,2b,10,ea,cd,b3,a0
940 DATA 79,cd,aa,a0,13,e5,21,7f,02,37
950 DATA ed,52,e1,38,05,2a,c0,a0,18,cd
960 DATA 23,7c,b5,c8,2b,11,00,00,22,c0
970 DATA a0,3e,03,bd,20,ba,7c,b4,20,b6
980 DATA 3e,04,32,c2,a0,18,af,3e,1b,cd
990 DATA aa,a0,3e,33,cd,aa,a0,3e,10,cd
1000 DATA aa,a0,c9,e5,3e,42,cd,1e,bb,e1
1010 DATA 28,02,e1,c9,3e,0d,cd,aa,a0,3e
1020 DATA 0a,cd,aa,a0,3e,1b,cd,aa,a0,3e
1030 DATA 2a,cd,aa,a0,3e,04,cd,aa,a0,3e
1040 DATA 7f,cd,aa,a0,3e,02,cd,aa,a0,c9
1050 DATA cd,2e,bd,38,fb,cd,2b,bd,c9,3a
1060 DATA c2,a0,fe,06,c8,af,cb,11,cb,11
1070 DATA c9,00,00,00,00
1080 :
1090 MEMORY &9FFF:total=0
1100 FOR i=&A000 TO &A0C2
1110 READ a$:a=VAL("&"+a$):POKE i,a
1120 total=total+a
1130 NEXT
1140 IF total<>24125 THEN PRINT"erreur":
STOP
1150 CALL &A000
1155 PLOT coordonneex,coordonneey
1160 RETURN
1170 REM
1180 REM SAUVEGARDE DE L'IMAGE
1190 LOCATE 1,24
1200 PRINT STRING$(70,CHR$(32));
1210 LOCATE 1,25
1220 PRINT STRING$(70,CHR$(32));
1230 REM
1240 LOCATE 1,24
1250 INPUT "Nom de la courbe (8 lettres
sans extension) ";nom$
1260 IF LEN(nom$) > 8 THEN GOTO 1190
1270 courbe$ = nom$ + ".PIC"
1280 parametre$ = nom$+ ".PAR"
1290 OPENOUT parametre$
1300 WRITE #9,a1,a2,b1,b2,c1,c2,d1,d2
1310 CLOSEOUT
1320 LOCATE 1,24
1330 PRINT STRING$(70,CHR$(32));
1340 SAVE courbe$,b,&C000,&4000
1350 '
1360 RETURN
1370 REM
1380 REM AFFICHAGE CARACTERISTIQUES
```

```
1390 REM
1400 LOCATE 1,24
1410 PRINT "x=" ; USING "####.###" ; a1
;
1420 PRINT "*COS(" ; USING "####.###" ;
a2 ;
1430 PRINT "*t)" ; USING "+####.###" ; b
1 ;
1440 PRINT "*SIN(" ; USING "####.###" ;
b2 ;
1450 PRINT "*t)"
1460 LOCATE 1,25
1470 PRINT "y=" ; USING "####.###" ; c1
;
1480 PRINT "*COS(" ; USING "####.###" ;
c2 ;
1490 PRINT "*t)" ; USING "+####.###" ; d
1 ;
1500 PRINT "*SIN(" ; USING "####.###" ;
d2 ;
1510 PRINT "*t)";
1520 '
1530 RETURN
1540 REM
1550 REM
1560 REM AFFICHAGE DU MENU
1570 REM
1580 LOCATE 1,10
1590 PRINT " 1 -> Trace d'une courbe"
1600 PRINT:PRINT
1610 PRINT " 2 -> Chargement d'une courb
e"
1620 REM
1630 WHILE a$ <> "1" AND a$ <> "2"
1640    a$ = INKEY$
1650 WEND
1660 REM
1670 ON VAL(a$) GOSUB 180,1720
1680 RUN
1690 REM
1700 REM CHARGEMENT D'UNE COURBE
1710 REM
1720 LOCATE 1,24
1730 INPUT "Nom de la courbe (8 lettres
sans extension) ";nom$
1740 courbe$ = nom$ + ".PIC"
1750 parametre$ = nom$ + ".PAR"
```

```
1760 OPENIN parametre$
1770 INPUT #9,a1,a2,b1,b2,c1,c2,d1,d2
1780 CLOSEIN
1790 MODE 2
1800 LOAD courbe$
1810 GOSUB 1400
1820 CALL &BB06
1830 RETURN
```

Lignes 10 à 160 : présentation du programme.

Ligne 170 : envoi au sous-programme du menu.

Lignes 180 à 220 : effacement du menu uniquement.

Lignes 230 à 330 : acquisition des paramètres.

Ligne 380 : envoi au sous-programme d'affichage des deux équations paramétriques.

Lignes 440 à 460 : sélection du rapport d'affichage.

Ligne 500 : on fixe l'origine du tracé du centre de la partie restante de l'écran.

Lignes 540 à 590 : un premier point est affiché. C'est le point de départ (pour t = 0), qui sert d'origine pour le tracé à l'aide de l'instruction **DRAW** qui suivra.

Ligne 640 : ue boucle infinie est ici réalisée car la condition 1 (de WHILE 1) est toujours réalisée.

Lignes 650 à 670 : tracé d'une portion de droite positionnant le point suivant.

Ligne 700 : incrémentation du compteur de temps t.

Lignes 720 et 730 : test d'appui sur une touche pour sortir de la boucle.

Lignes 790 à 840 : traitement de la touche frappée, pour orientation vers les sous-programmes correspondants.

Lignes 880 à 1150 : les données hexadécimales pour la copie graphique sont placées en mémoire, puis la copie est lancée par **CALL &A000**.

Ligne 1155 : on positionne à nouveau le point de tracé où il s'était arrêté, car la copie graphique le déplace.

Lignes 1190 à 1360 : l'image est sauvegardée à partir d'un nom donné, en deux fichiers précédemment expliqués.

Lignes 1400 à 1530 : affichage dans le bas de l'écran des deux équations paramétriques.

Lignes 1580 à 1680 : affichage et gestion des choix du menu.

Lignes 1720 à 1830 : acquisition du nom d'une courbe déjà sauvegardée et affichage de celle-ci.

Le retour du cercle

Commençons par donner la valeur 0 à b1 et d1, et une valeur R identique à a1 et c1. Prenons aussi 1 pour valeur des coefficients a2, b2, c2 et d2.

Nous obtenons ainsi le système d'équations suivant :

x = R * COS(t)
y = R * SIN(t)

En appliquant l'algorithme précédent, il vous sera facile de dessiner un cercle de rayon R.

Cette équation est donc l'équation paramétrique du cercle tant étudiée en mathématiques.

Le cercle étiré

Reprenons les équations précédentes en modifiant le paramètre R dans la deuxième, afin qu'il soit différent de celui de la première équation.

x = R1 * COS(t)
y = R2 * COS(t)

Ce nouveau système vous permettra de tracer une ellipse.

Electronique et courbes remarquables

Les électroniciens sont peut-être ceux qui utilisent le plus les fonctions trigonométriques et surtout la fonction SINUS.

En effet, toutes les tensions sinusoïdales ont des équations de la forme :

x = Xmax * SIN(w * t + PHI)

avec : Xmax la valeur maximale de la tension ;
w la pulsation, qui peut encore s'écrire :

w = 2 * PI * fréquence
ou w = 2 * PI / période

et qui s'exprime en radian par seconde (rad/s) ;

et PHI le déphasage, que l'on peut considérer comme étant le retard ou l'avance du signal, et qui s'exprime en radian.

Ainsi, on peut considérer deux signaux sinusoïdaux, que l'on désire représenter l'un par rapport à l'autre ; c'est ce que l'on effectue sur un oscilloscope en utilisant un balayage appelé X/Y. On obtient ce que l'on nomme dans le jargon électronicien les courbes de Lissajous.

x = Xmax * SIN(w1 * t + PHI1)
y = Ymax * SIN(w2 * t + PHI2)

Pour intégrer ces équations aux données de notre programme, il nous faut les transformer à l'aide des formules citées plus haut :

Xmax * SIN(w1 * t + PHI1)
= Xmax * SIN(w1 * t) * COS(PHI1)
+ Xmax * COS(w1 * t) * SIN(PHI1)
= Xmax * SIN(PHI1) * COS(w1 * t)
+ Xmax * COS(PHI1) * SIN(w1 * t)

De même pour y, on obtient :

Ymax * SIN(w2 * t + PHI2)
= Ymax * SIN(PHI2) * COS(w2 * t)
+ Ymax * COS(PHI2) * SIN(w2 * t)

Nous obtenons ainsi nos termes :

a1 = Xmax * SIN(PHI1)	a2 = w1
b1 = Xmax * COS(PHI1)	b2 = w1
c1 = Ymax * SIN(PHI2)	c2 = w2
d1 = Ymax * COS(PHI2)	d2 = w2

Si nous prenons par exemple deux signaux, de valeur maximale identique 1, de pulsation identique 1, mais dont l'un est déphasé de $\pi/6$, nous obtenons les équations « électroniques » suivantes :

x = SIN (t)
y = SIN(t + PI/6)

Ce qui nous donne le système d'équations :

x = SIN(t)
y = 0.5 COS(t) + .866 SIN(t)

Les valeurs à donner au programme sont ainsi :

a1 = 0	a2 = 0
b1 = 1	b2 = 1
c1 = 0.5	c2 = 1
d1 = 0.866	d2 = 1

Vous obtenez ainsi la courbe représentée en figure 1.

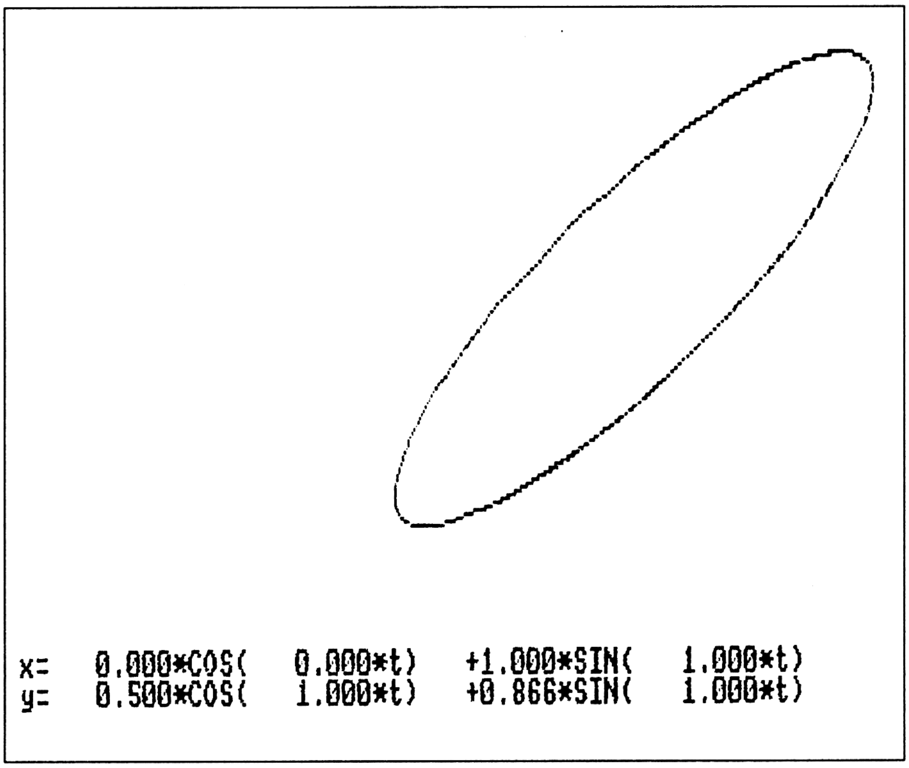

Fig. 1 : Lissajous sur signaux déphasés,

$$x = \text{SIN } t \text{ ; } y = \text{SIN } t + \frac{\pi}{3}$$

Il est aussi possible, non plus de modifier le déphasage, mais la valeur des pulsations des deux signaux (ce qui revient à modifier les fréquences).

Par exemple, en « pulsant » l'un des signaux deux fois plus rapidement que l'autre, on obtient les équations suivantes :

x = SIN(t) → Xmax = 1 ; w1 = 1 ; PHI1 = 0
y = SIN(2 * t) → Ymax = 1 ; w2 = 2 ; PHI2 = 0

Les paramètres de nos équations sont simples :

a1 = a2 = c1 = c2 = 0
b1 = b2 = d1 = 1
d2 = 2

et l'on obtient le nœud papillon de la figure 2.

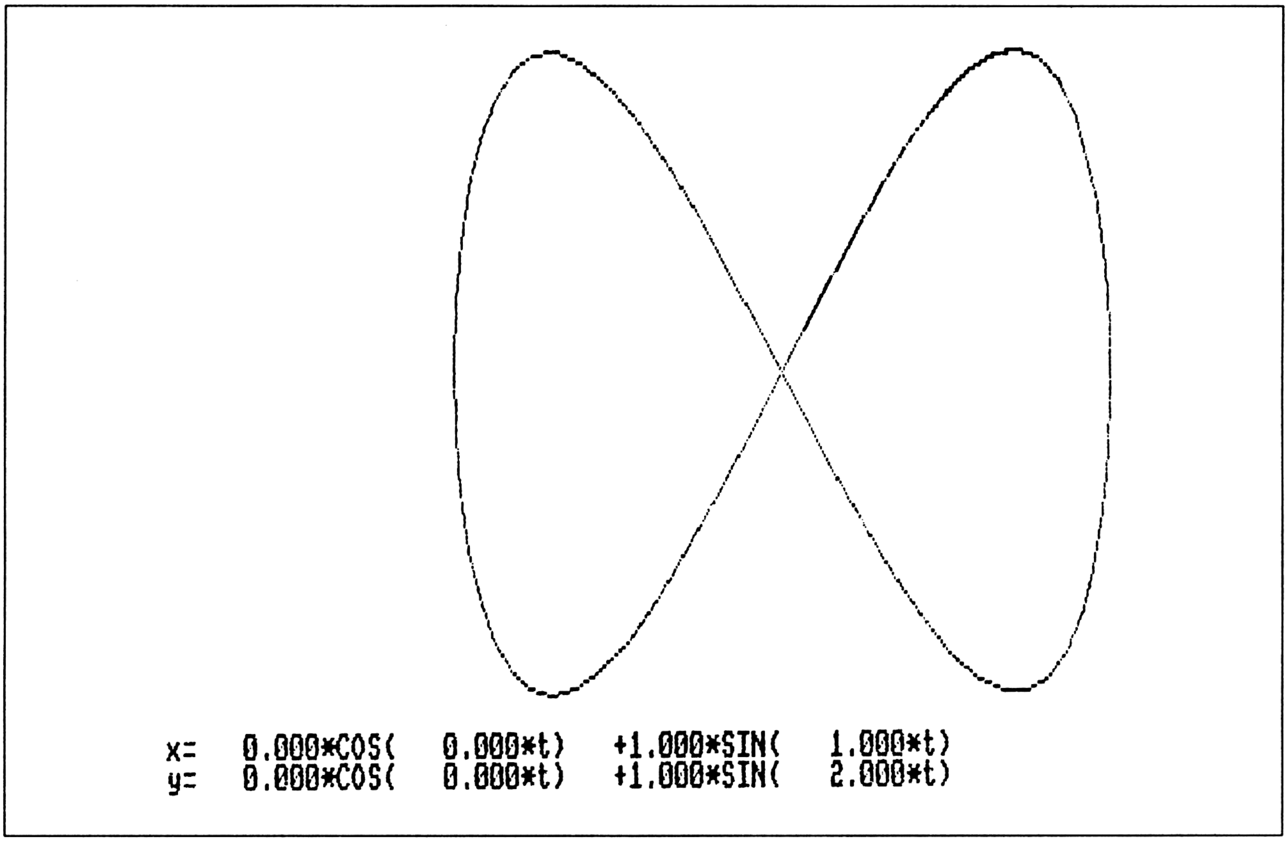

Fig. 2 : Lissajous sur la fréquence,
x = SIN(t) ; y = SIN(2t).

Si vous inversez les paramètres b2 et d2, ce qui revient à doubler la fréquence de x par rapport à y, vous obtenez le même nœud papillon, mais placé par son propriétaire un lendemain de fête bien arrosée.

Et si nous quadruplions la fréquence ? Dans ce cas, le paramètre d2 devient égal à 4, et nous obtenons la courbe de la figure 3.

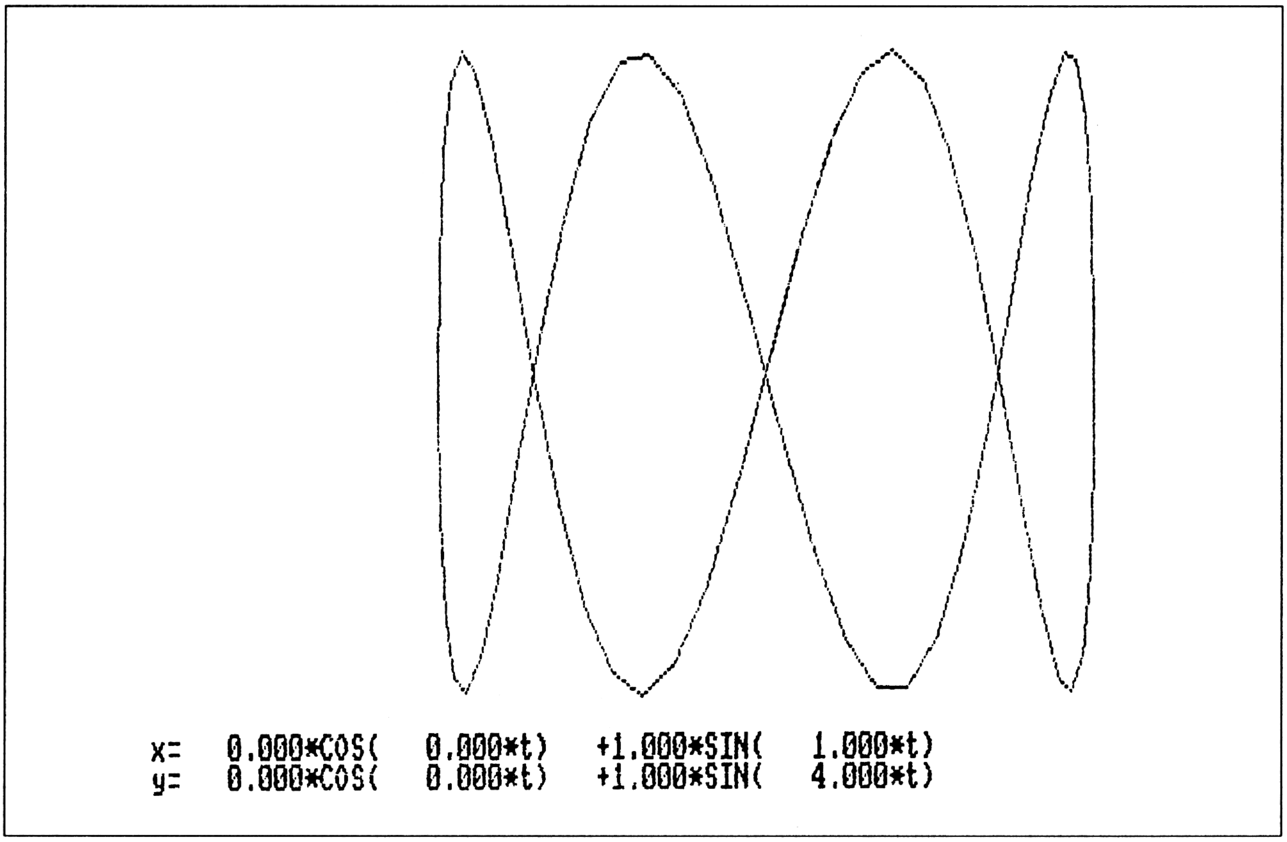

Fig. 3 : x = SIN(t) ; y = SIN(4 t).

Pour rester dans le domaine de Lissajous, il est possible d'imaginer des signaux complètement farfelus (ce que vous dirons les électroniciens, car il leur sera certainement très difficile, voir impossible de les réaliser).

Ces signaux, si vous restez tout de même dans les limites raisonnables de la folie sinusoïdale, vous donnerons probablement des figures proches de relevés réels de Lissajous, et vous permettrons certainement d'empêcher de dormir quelque électronicien pointilleux, en ne lui révélant pas les équations. La figure 4 représente une représentation en X/Y de deux signaux presque probables, et prend pour paramètres :

a1 = SIN(PI/6) = 0.5	a2 = 5
b1 = COS(2 * PI/3) = - 0.5	b2 = 5
c1 = SIN(PI/4) = 0.707	c2 = 6
d1 = COS(5 * PI/6) = - 0.866	d2 = 6

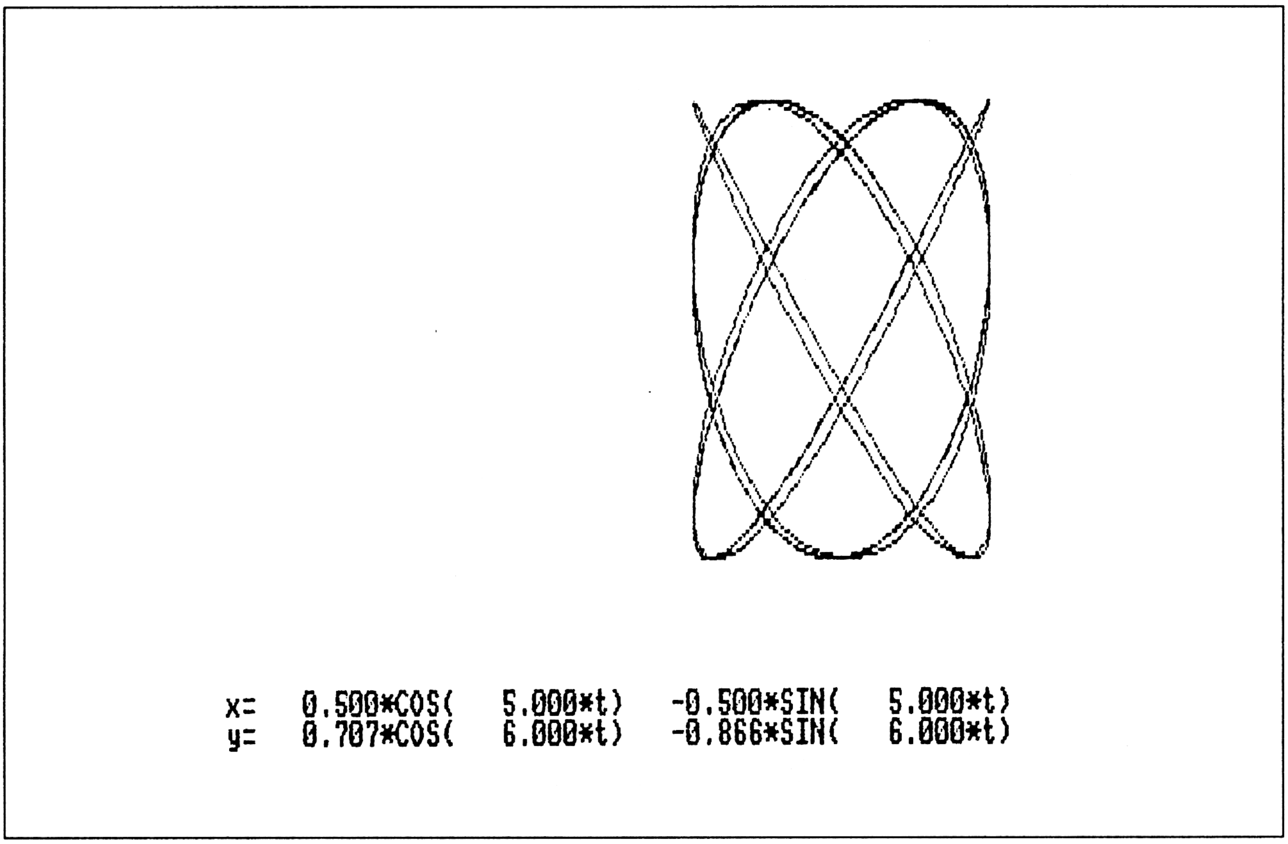

Fig. 4 : Lissajous sur signaux sinusoïdaux bizarres.

Divagations trigonométriques

Notre étude est restée jusque maintenant proche de courbes plus ou moins connues, tracées en fait par des équations remarquables.

Nous vous proposons maintenant de faire voyager votre imagination sur les nombres au sein de notre programme, et de résoudre certains problèmes de représentation.

Notre première courbe folle semble encore proche de Lissajous si nous l'arrêtons au point représenté sur la copie graphique de la figure 5.

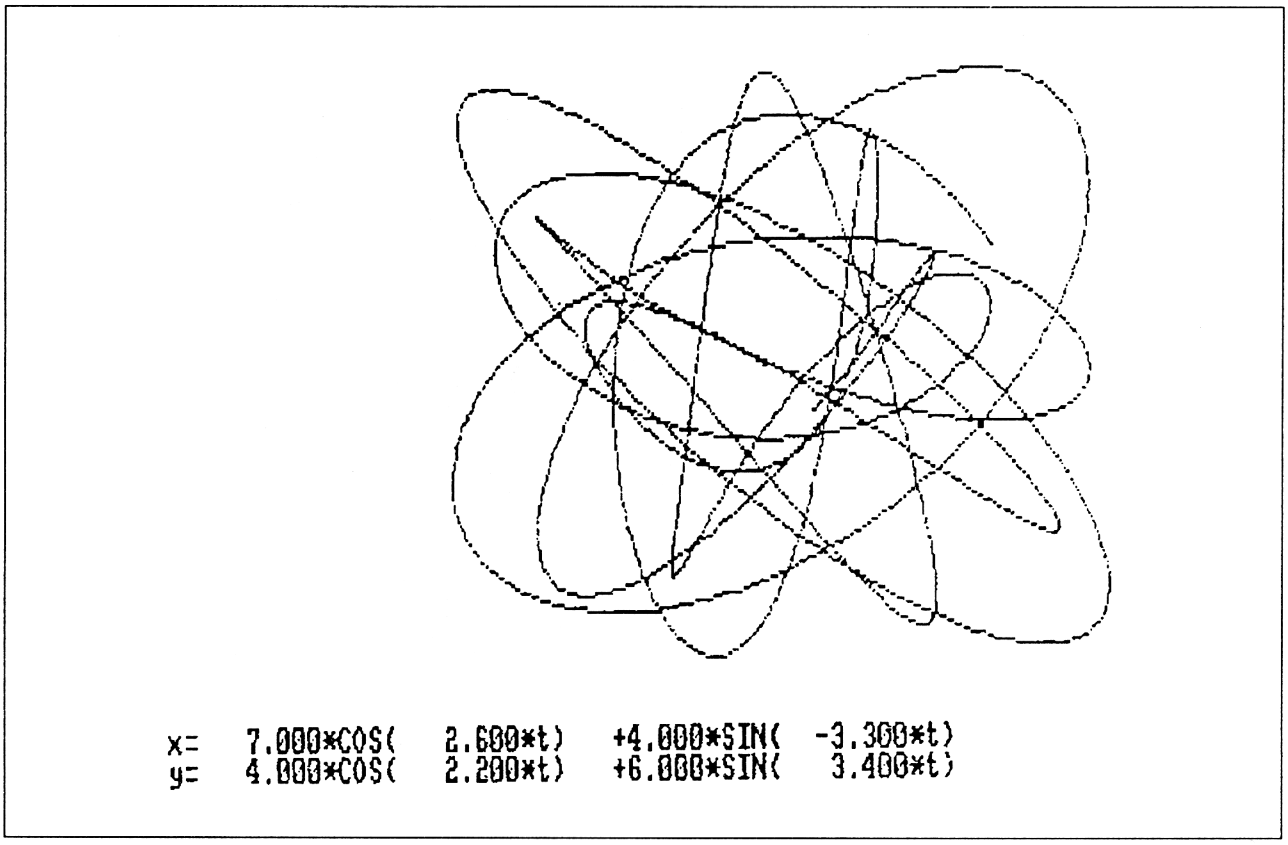

Fig. 5 : Bizarrerie sinusoïdale.

Pas question de la présenter à un technicien électronique, il découvrirait de suite la supercherie.

L'ensemble des nombres étant infini, nous vous proposons d'augmenter, doucement pour l'instant les valeurs des constantes a1, a2, b1, b2, c1, c2, d1 et d2.

Dans les environs des centaines, prenons par exemple :

a1 = 101	a2 = 102
b1 = 103	b2 = 104
c1 = 105	c2 = 106
d1 = 107	d2 = 108

La courbe obtenue avec notre programme est représentée en figure 6.

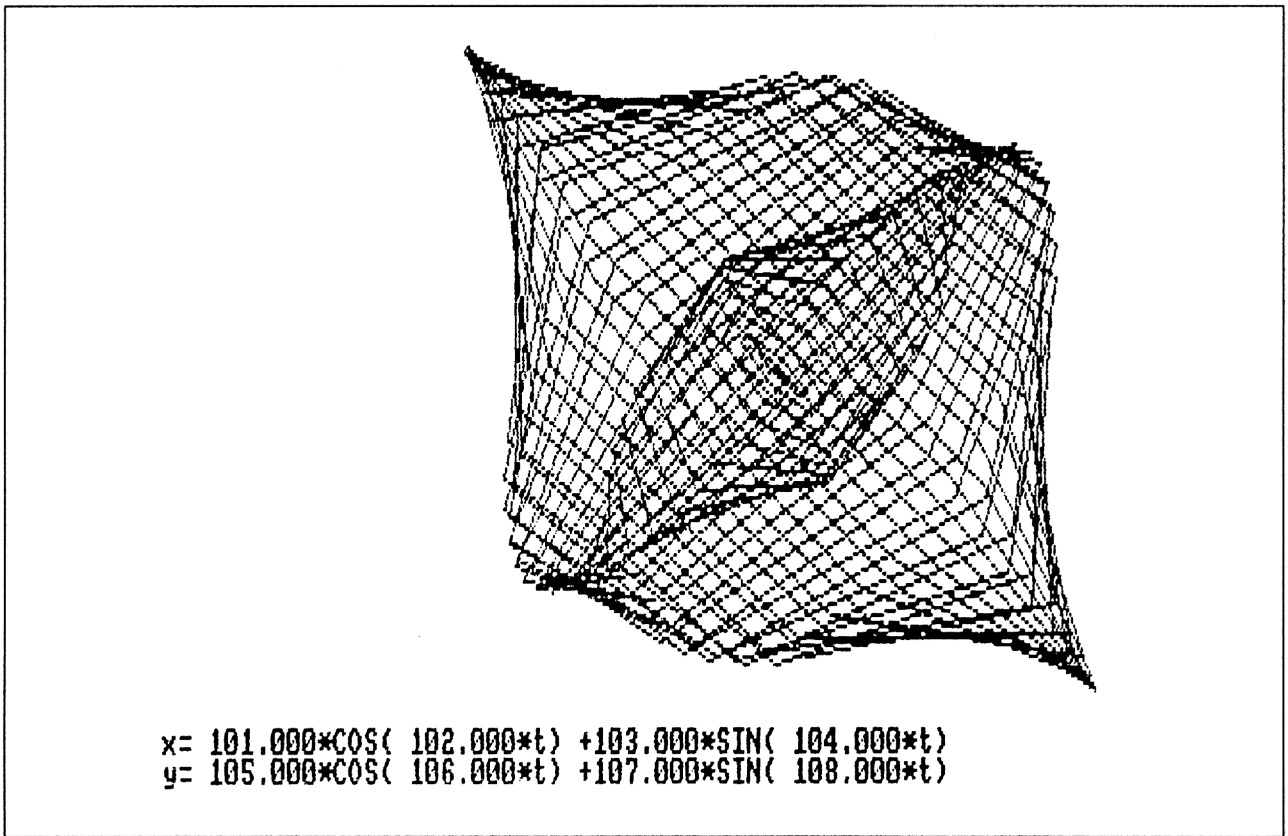

Fig. 6 : Tracé pour t = 0.01.

Bien qu'agréable à regarder, une question se pose sur cette figure : mais où sont donc passées les harmonieuses courbes des fonctions SINUS et COSINUS ?

Le problème provient de la ligne 700 du programme, qui concerne l'incrémentation de la valeur t.

700 t = t + 0.01

La pulsation étant tellement élevée, que la valeur w * t varie beaucoup trop pour obtenir des valeurs de SINUS et COSINUS proches les unes par rapport aux autres.

Comme nous relions nos points à l'aide de l'instruction DRAW, nous obtenons des sections de droites reliant deux points calculés très éloignés.

La solution est de diminuer la valeur de l'incrément de t, par exemple dans notre cas, à 0.001 au lieu de 0.01, et tout rentre dans l'ordre, comme le confirme la figure 7.

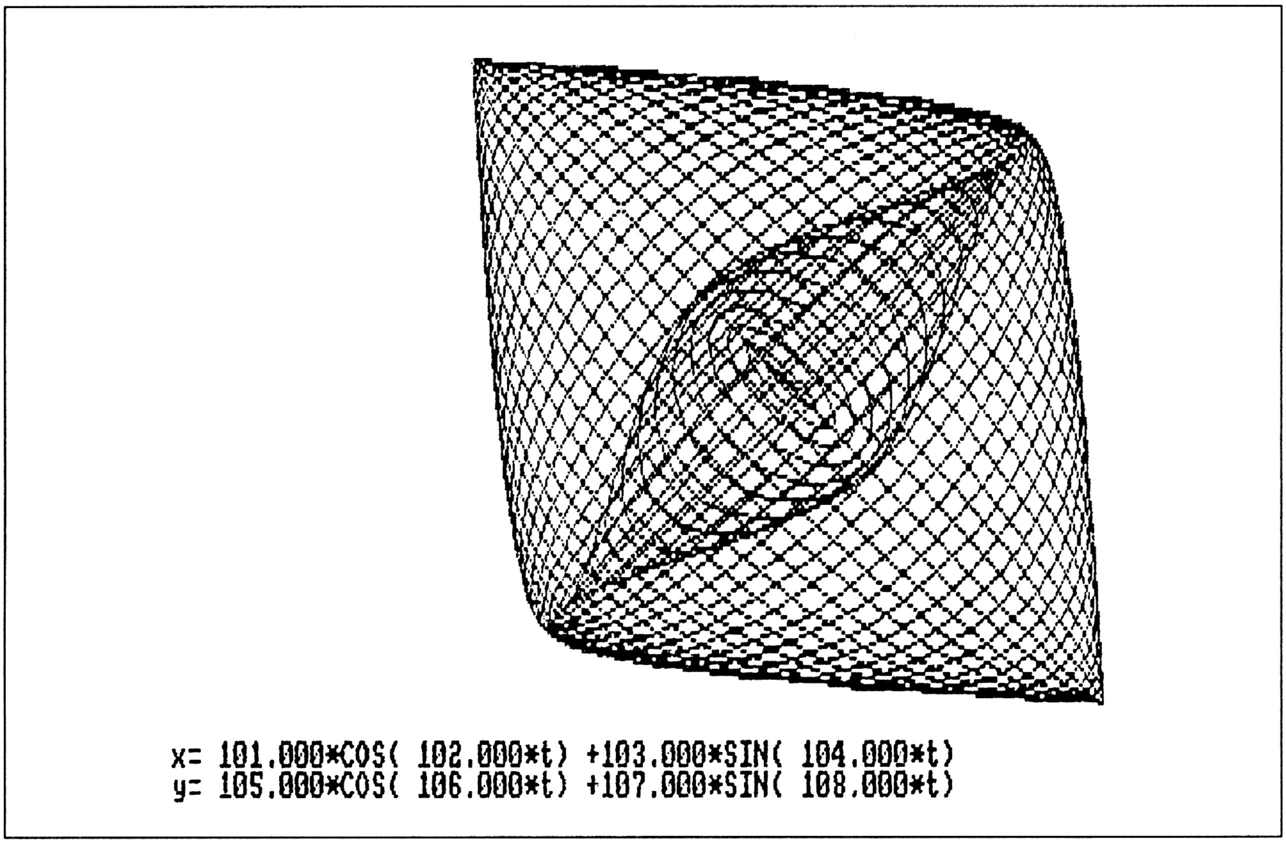

Fig. 7 : Tracé pour t = 0.001.

Toujours autour de la centaine, la figure 8 nous permet d'admirer une rosace sinusoïdale.

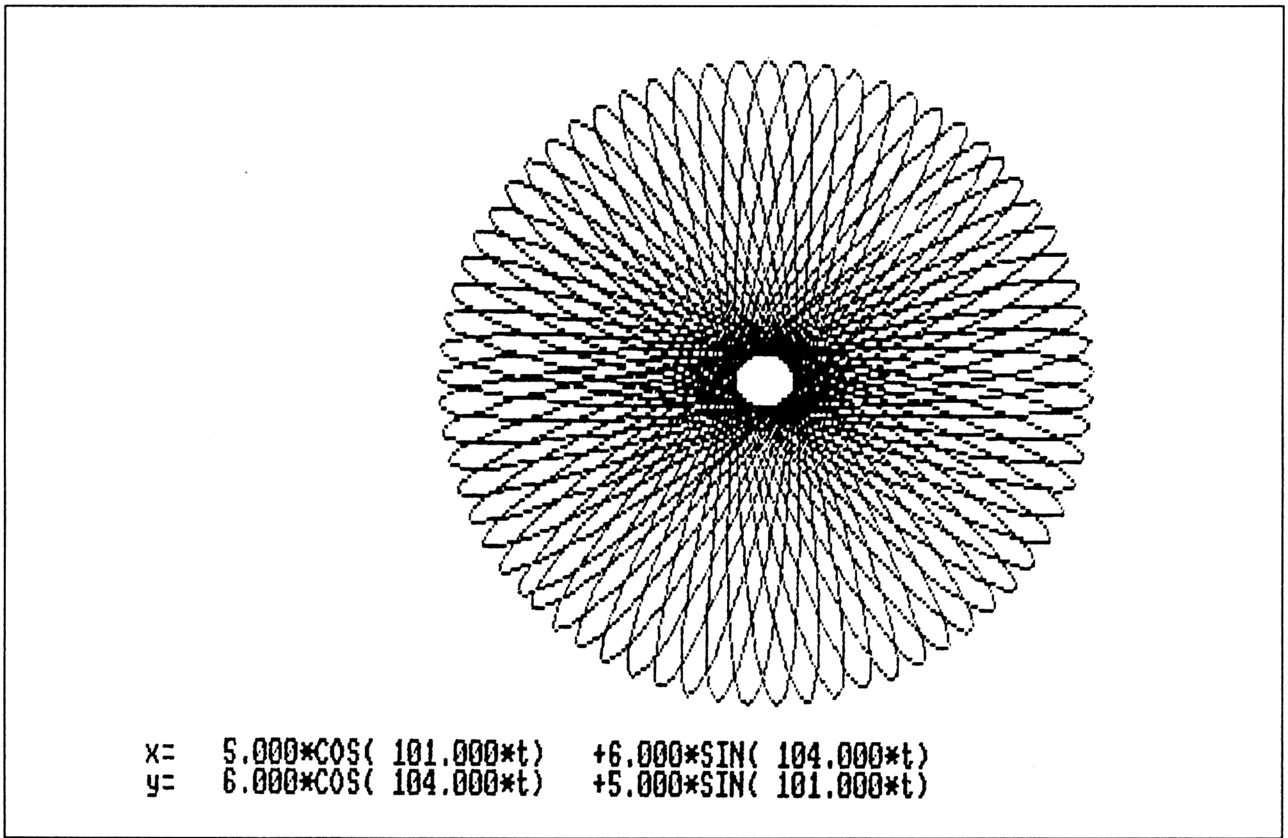

Fig. 8 : Rosace sinusoïdale pour t = 0.001.

Si nous poussions nos investigations plus loin dans les grands nombres, en passant dans l'ordre du millier pour les valeurs de w, nous retrouvrions le même problème, et serions obligés de diminuer l'incrément de t à 0.0001.

De là à faire trouver au programme la valeur adéquate de l'incrément en fonction de la valeur maximale des pulsations données (paramètres a2, b2, c2 et d2), il n'y a qu'un pas que nous vous invitons à franchir pour parfaire vos talents de programmeur. Signalons simplement que cette valeur devra être inversement proportionnelle à la pulsation.

En poussant l'irréel

Pas question ici de faire le tour de toutes les courbes inimaginables, votre bibliothèque ne supporterait pas le poids des feuilles.

Nous faisons confiance à votre imagination, et nous proposons pour conclure les paramètres d'une courbe découverte pas Stanley Miller.

a1 = - 0.7	a2 = 3.01
b1 = 1	b2 = 0.99
c1 = 1	c2 = 1.01
d1 = 0.1	d2 = 15.03

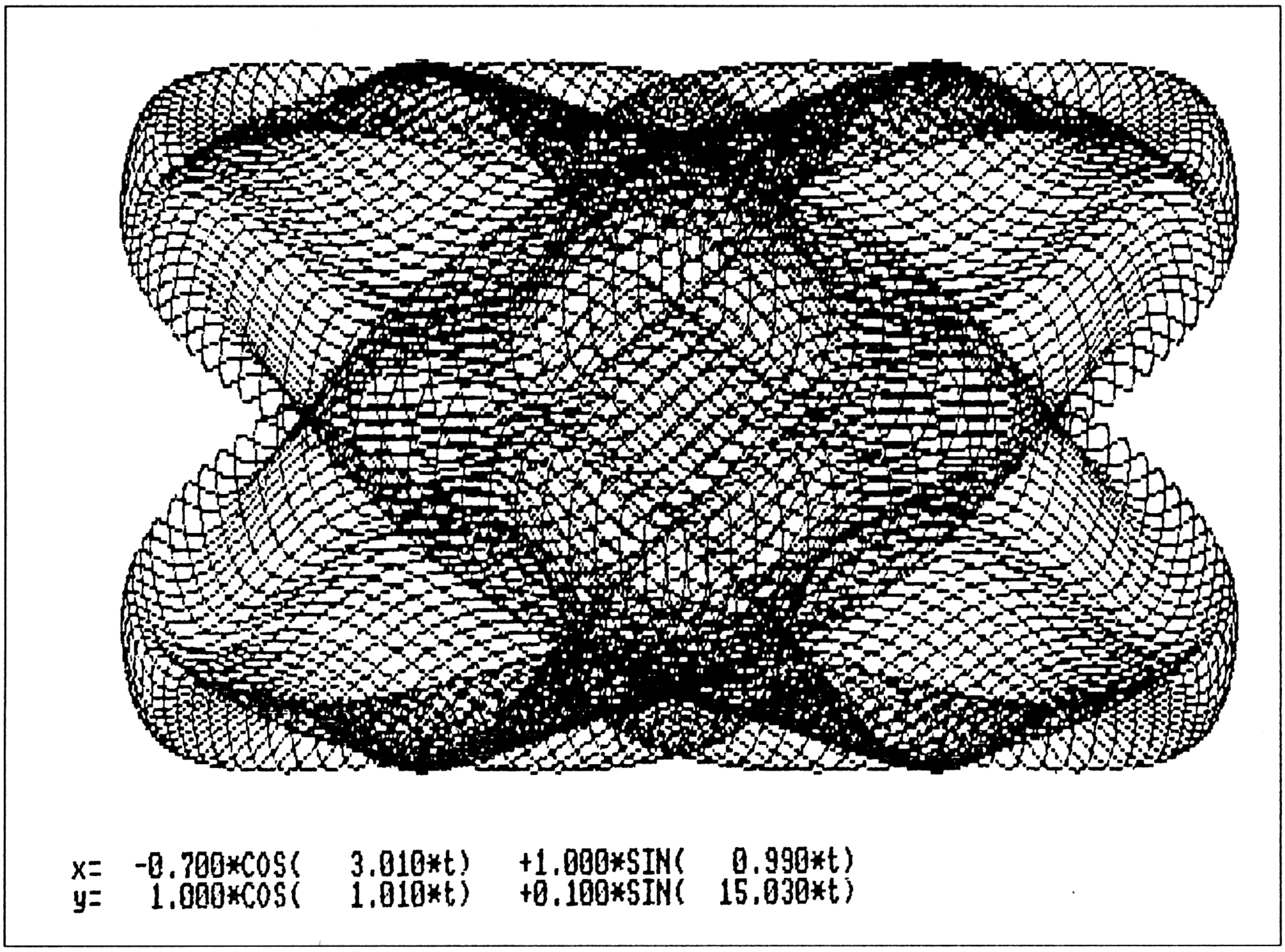

Fig. 9 : La dentelle de S. Miller.

Il vous faudra un peu de patience pour découvrir l'étendue de cette dentelle à faire rougir une bigouden.