SOMMAIRE

Préliminaire lapidaire.

Cette brochure constitue un fragment du travail fait à Grenoble dans le champ de la mini-informatique et de ses applications pédagogiques. Elle réunit des articles reflétant des activités effectivement réalisées à l'I.R.E.M. de Grenoble.

La présentation est à l'image de la diversité de la formation informatique des auteurs : stagiaires au stage INRDP, autodidactes...

Notre démarche ayant évolué vers une analyse et une présentation plus systématique des programmes, nous n'écririons plus les choses de la même façon.

EGALITE DE BEZOUT

Problème.

Deux naturels a et b étant donnés, on cherche u et v entiers tels que $ua + vb = d$, d étant le P.G.C.D. de a et de b.

Si (u, v) est une solution, alors il est évident que $(u + b, v - a)$ en est aussi une. Le problème admet donc une infinité de solutions. On va donc chercher celle pour laquelle on a :

$$0 < v < a.$$

On en déduit $\quad 0 < vb < ab$

$$d - ab < d - vb < d.$$

Comme $d - vb = ua$, il en résulte :

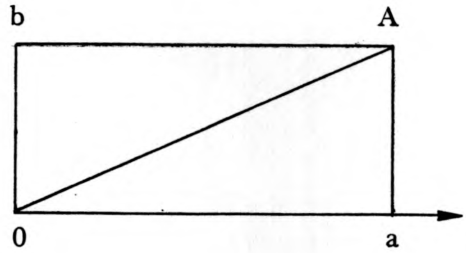
$$\frac{d}{a} - b < u < \frac{d}{a}.$$

Puisque u est entier et que $\frac{d}{a}$ est inférieur à 1, on aura $-b < u \leqslant 0$. On cherche donc une solution pour laquelle v est positif entre 0 et a, u est négatif entre -b et 0.

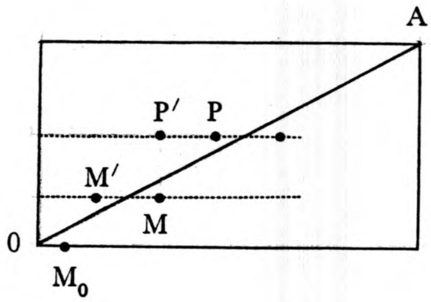
Méthode graphique.

Posons $u = -y$; $v = x$. On cherche x et y tels que $bx - ay = d$, avec $0 < x < a$ et $0 < y < b$.

Supposons $a > b$ et considérons le rectangle des points dont les coordonnées vérifient $0 \leqslant x \leqslant a$ et $0 \leqslant y \leqslant b$.

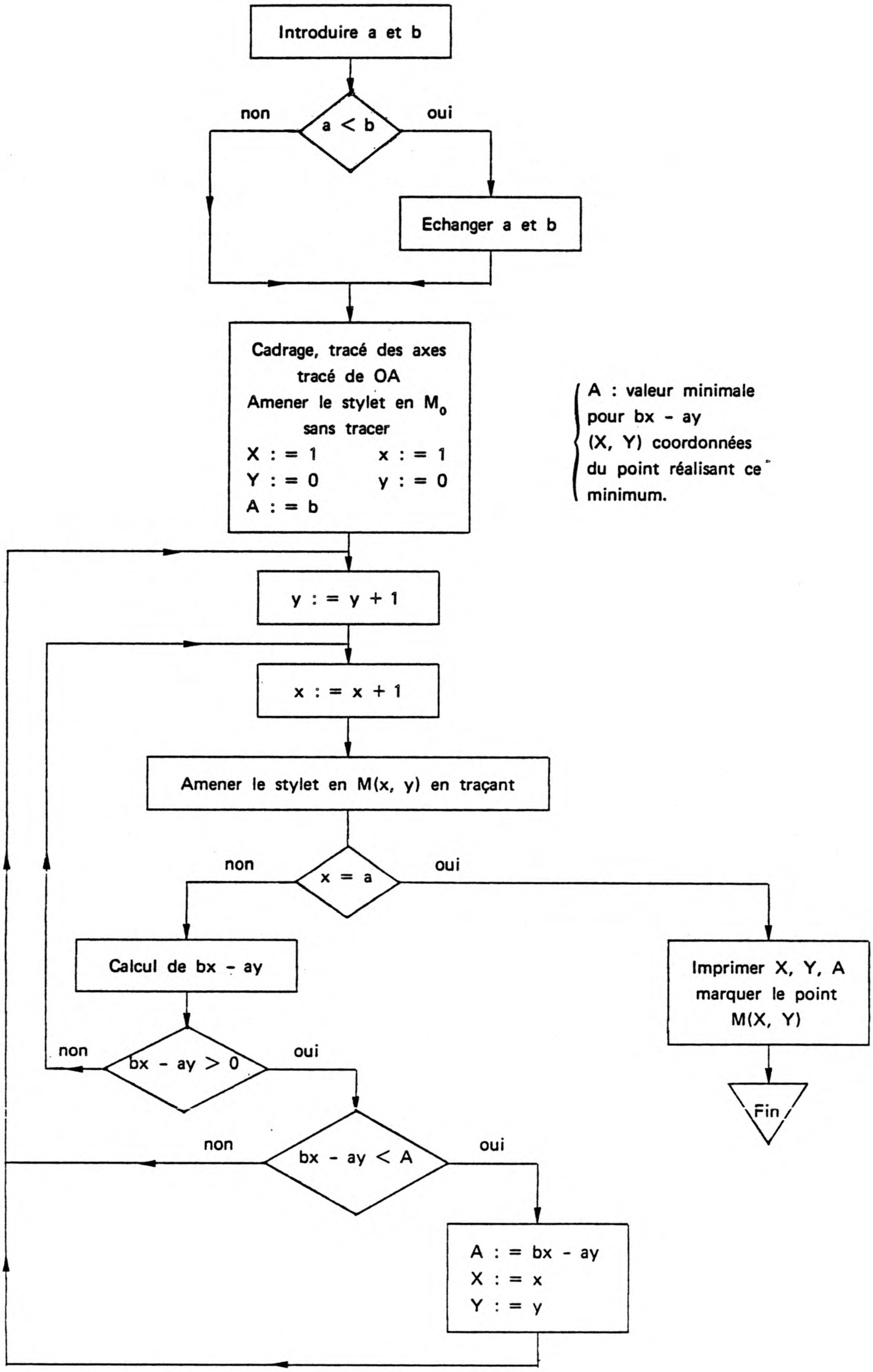


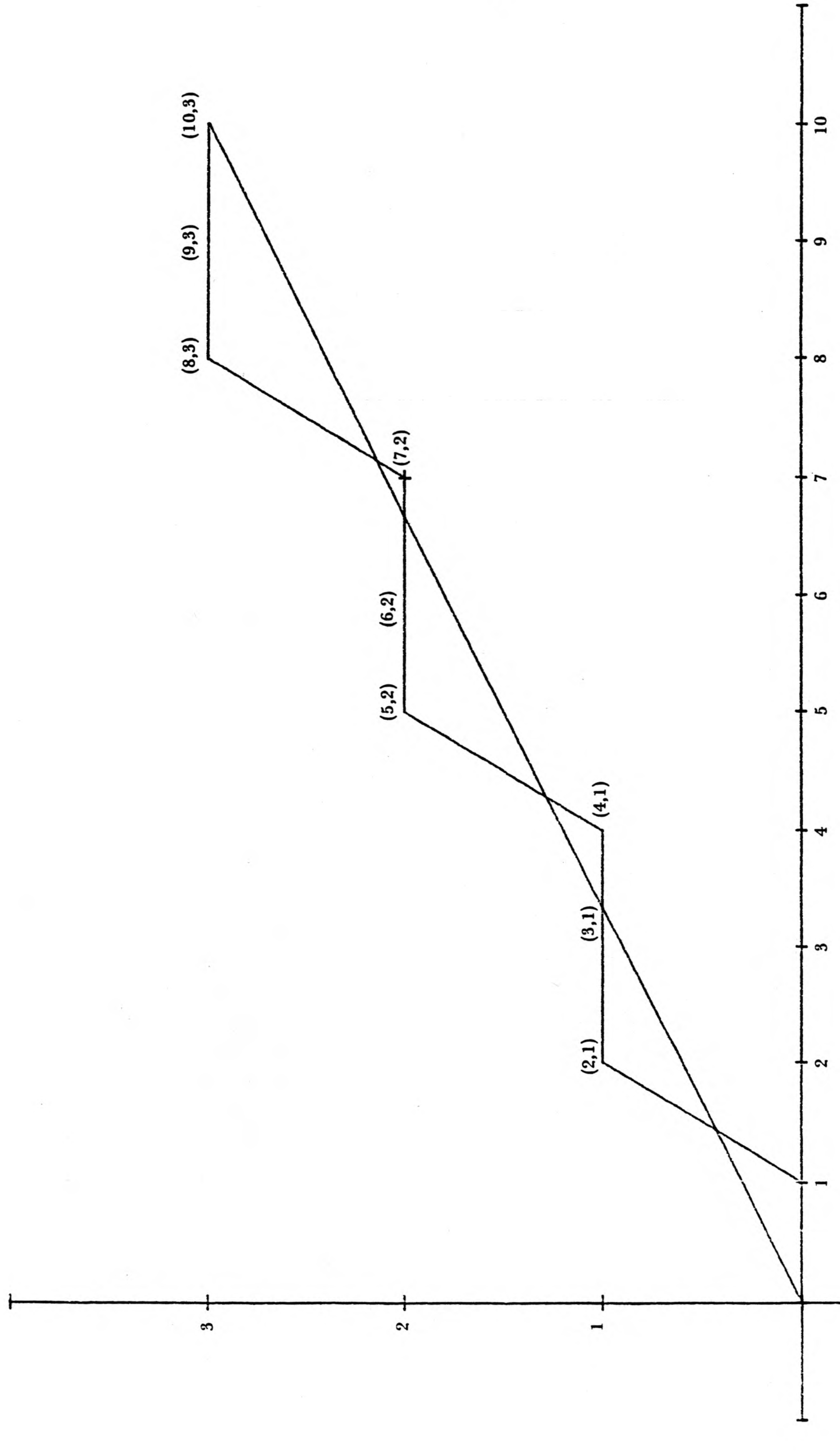
La droite OA est la droite d'équation $bx - ay = 0$. Pour un point situé au-dessus de la droite, on a $bx - ay < 0$. A tout point de coordonnées entières situé en dessous de la droite correspond une valeur de $bx - ay$ positive. On cherche donc un point de cette région pour lequel cette quantité soit minimale. y étant fixe, si x augmente, $bx - ay$ augmente. Sur chaque parallèle à l'axe des x, on va donc chercher le premier point situé en dessous de la droite.



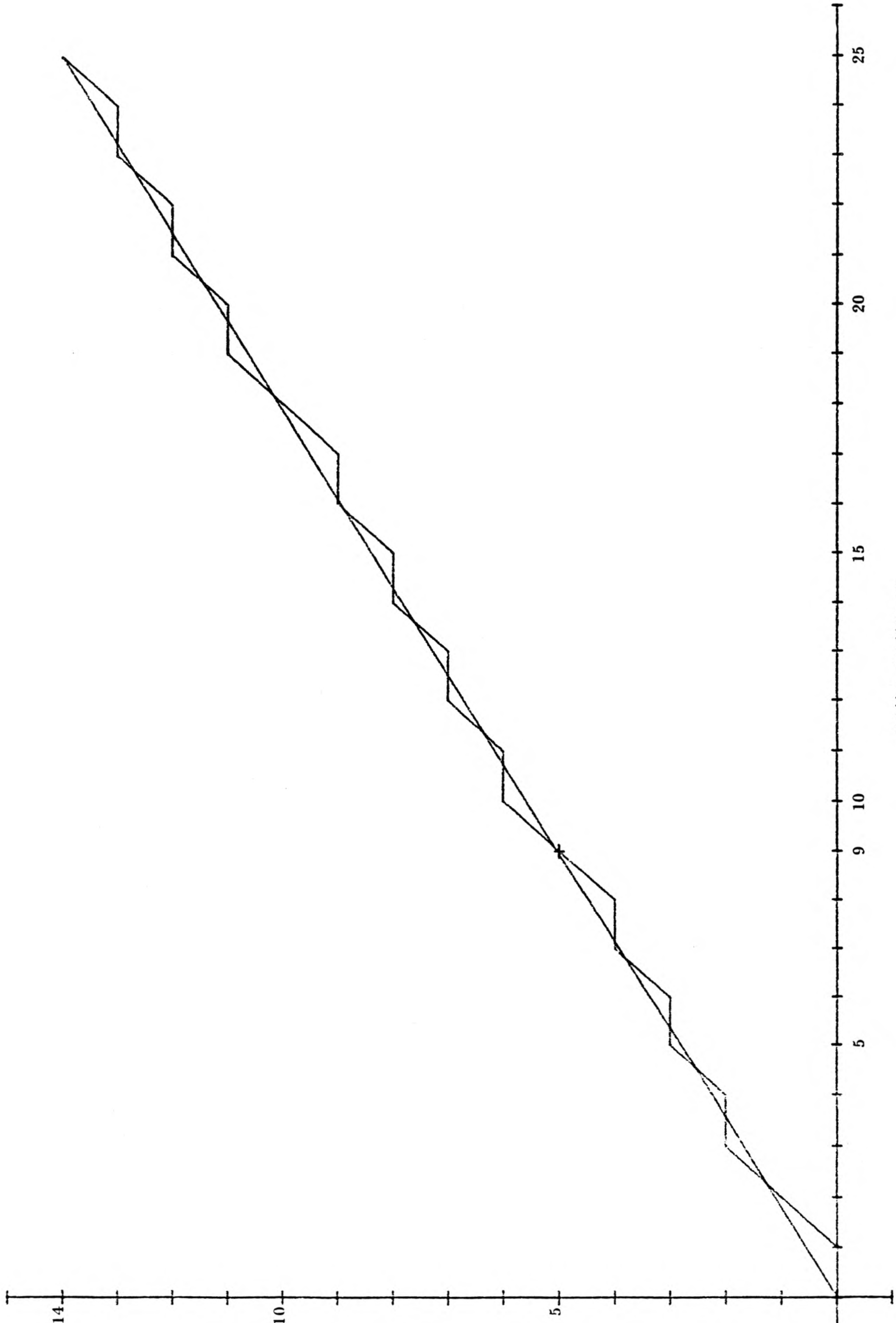
Si le point M(x, y) est situé en dessous de la droite OA tandis que le point $M'(x - 1, y)$ est au-dessus, puisque le coefficient directeur de OA est inférieur à 1, le point $P'(x, y + 1)$ est au-dessus de la droite. On devra donc regarder comment le point $P(x + 1, y + 1)$ est situé par rapport à la droite OA. S'il est au-dessus de la droite, on cherchera le point suivant de la parallèle à l'axe des x. S'il est au-dessous, on recommence pour lui ce qui a été fait pour M.

On commence par le point $M_0(1, 0)$ et on fait tracer le cheminement suivi sur la table traçante.

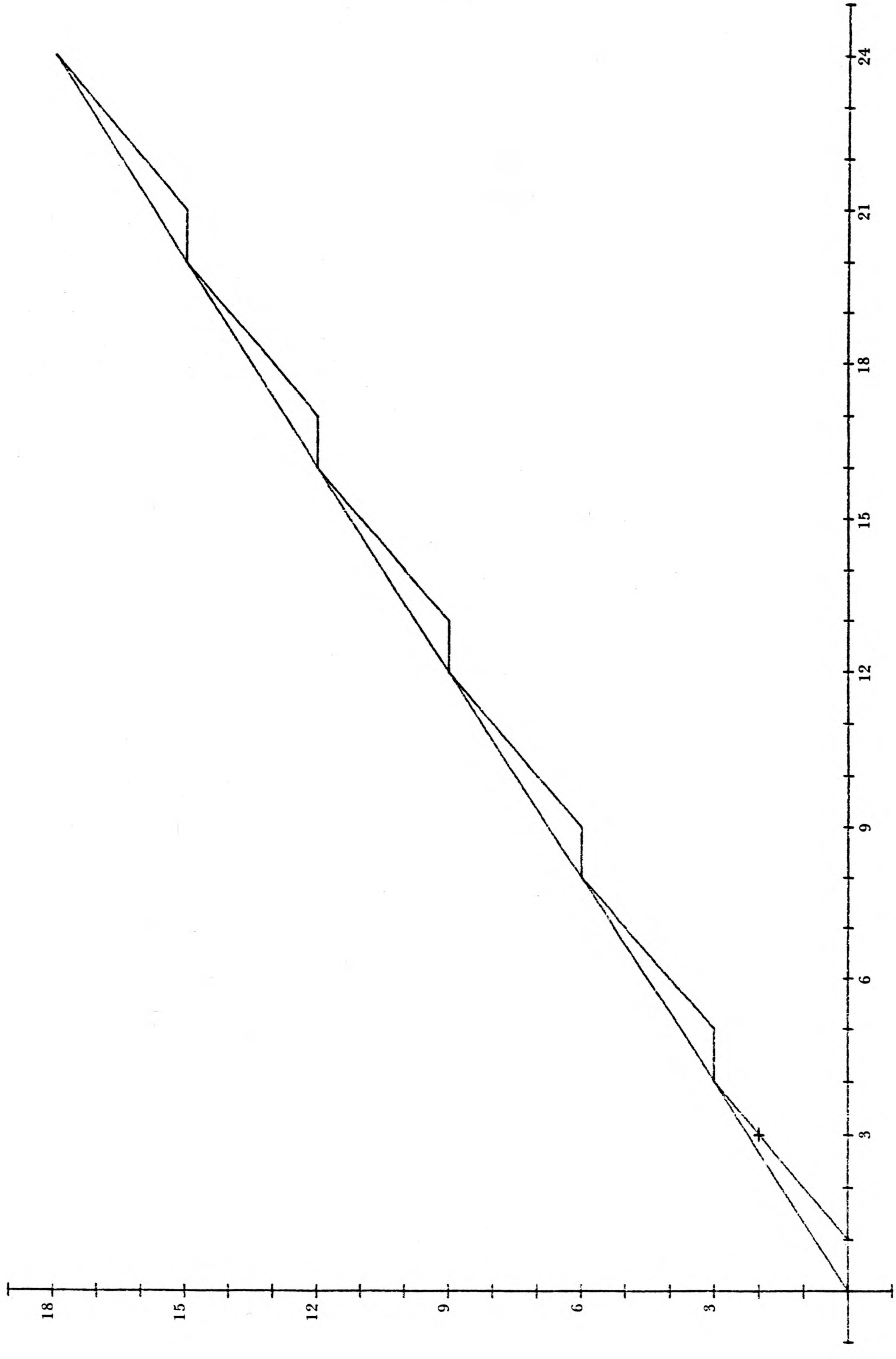

Introduire a et b
non
a < b
oui
Echanger a et b
Cadrage, tracé des axes
tracé de OA
Amener le stylet en M0
sans tracer
X : = 1
x : = 1
Y : = 0
y : = 0
A : = b
A : valeur minimale
pour bx - ay
(X, Y) coordonnées
du point réalisant ce
minimum.
y : = y + 1
x : = x + 1
Amener le stylet en M(x, y) en traçant
non
x = a
oui
Calcul de bx - ay
Imprimer X, Y, A
marquer le point
M(X, Y)
non
bx - ay > 0
oui
Fin
non
bx - ay < A
oui
A : = bx - ay
X : = x
Y : = y



$3 \times 7 - 10 \times 2 = 1$



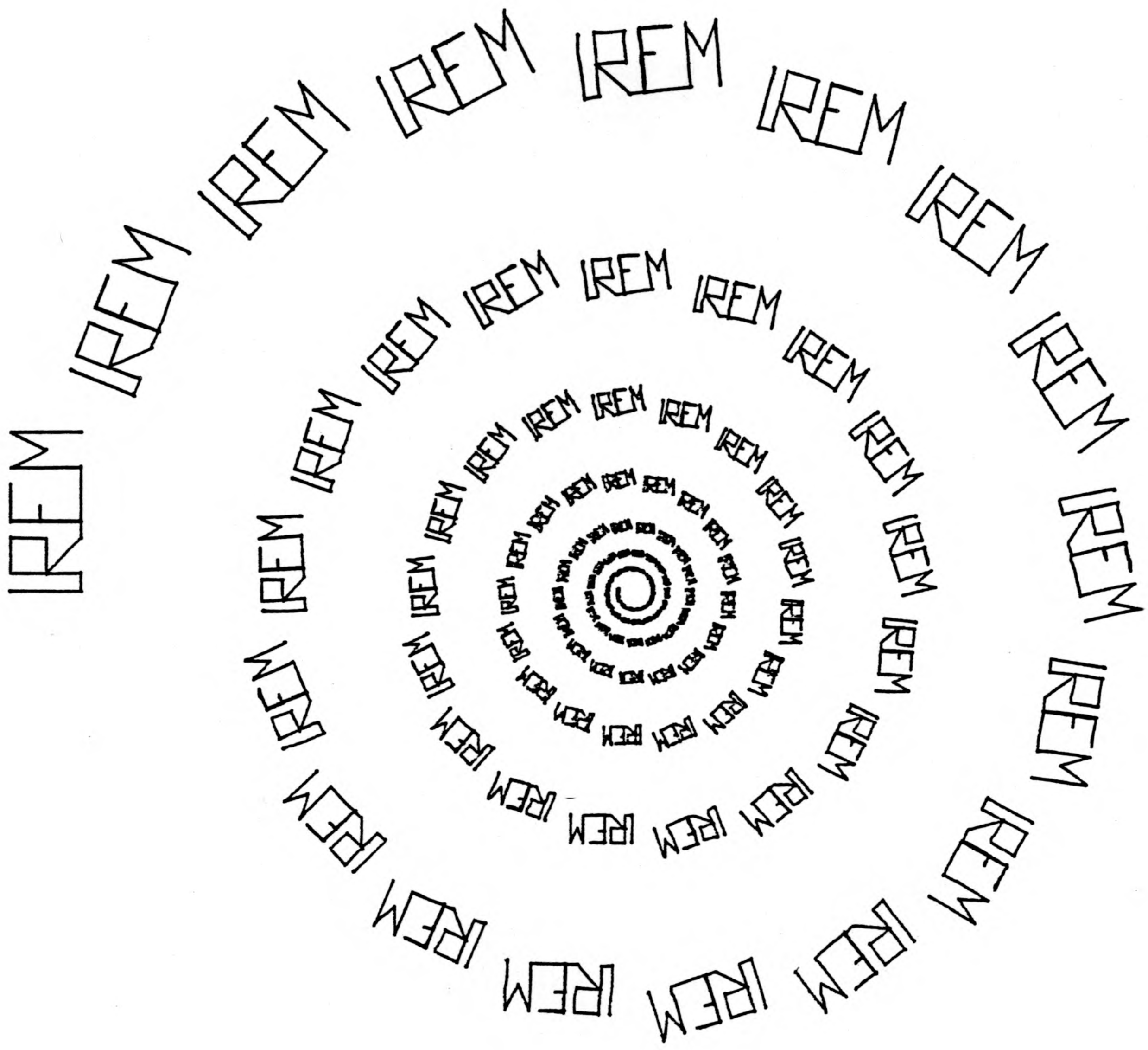
$14 \times 9 - 25 \times 5 = 1$

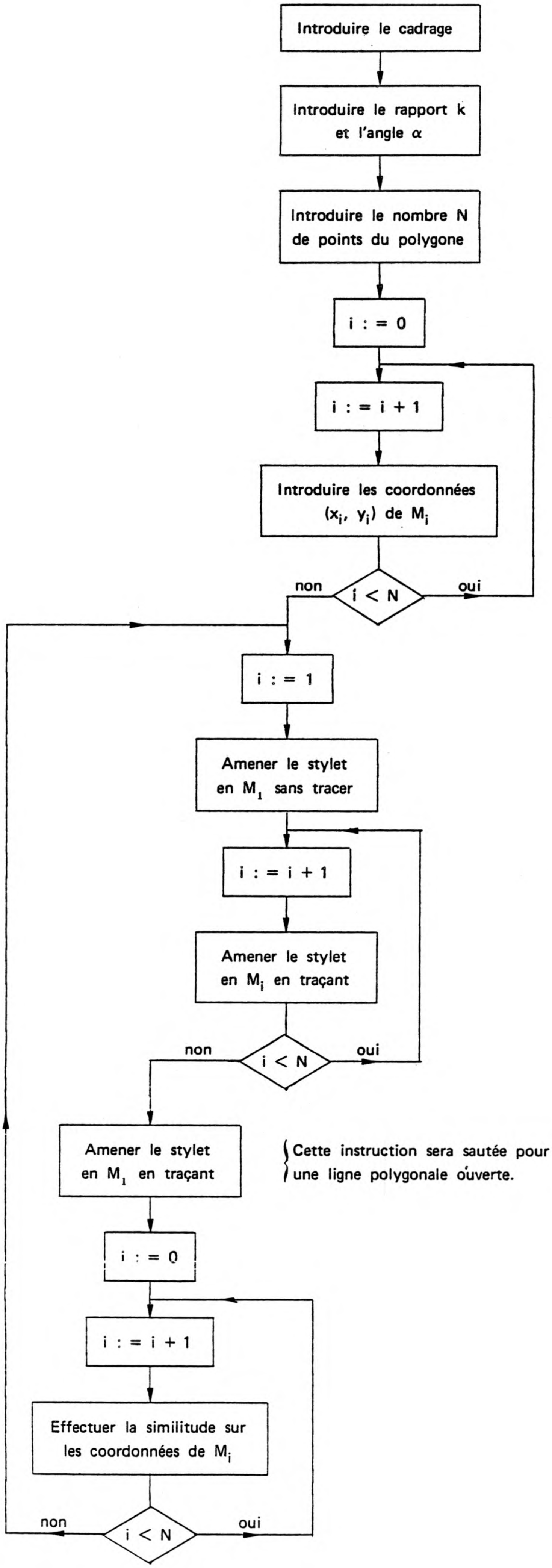


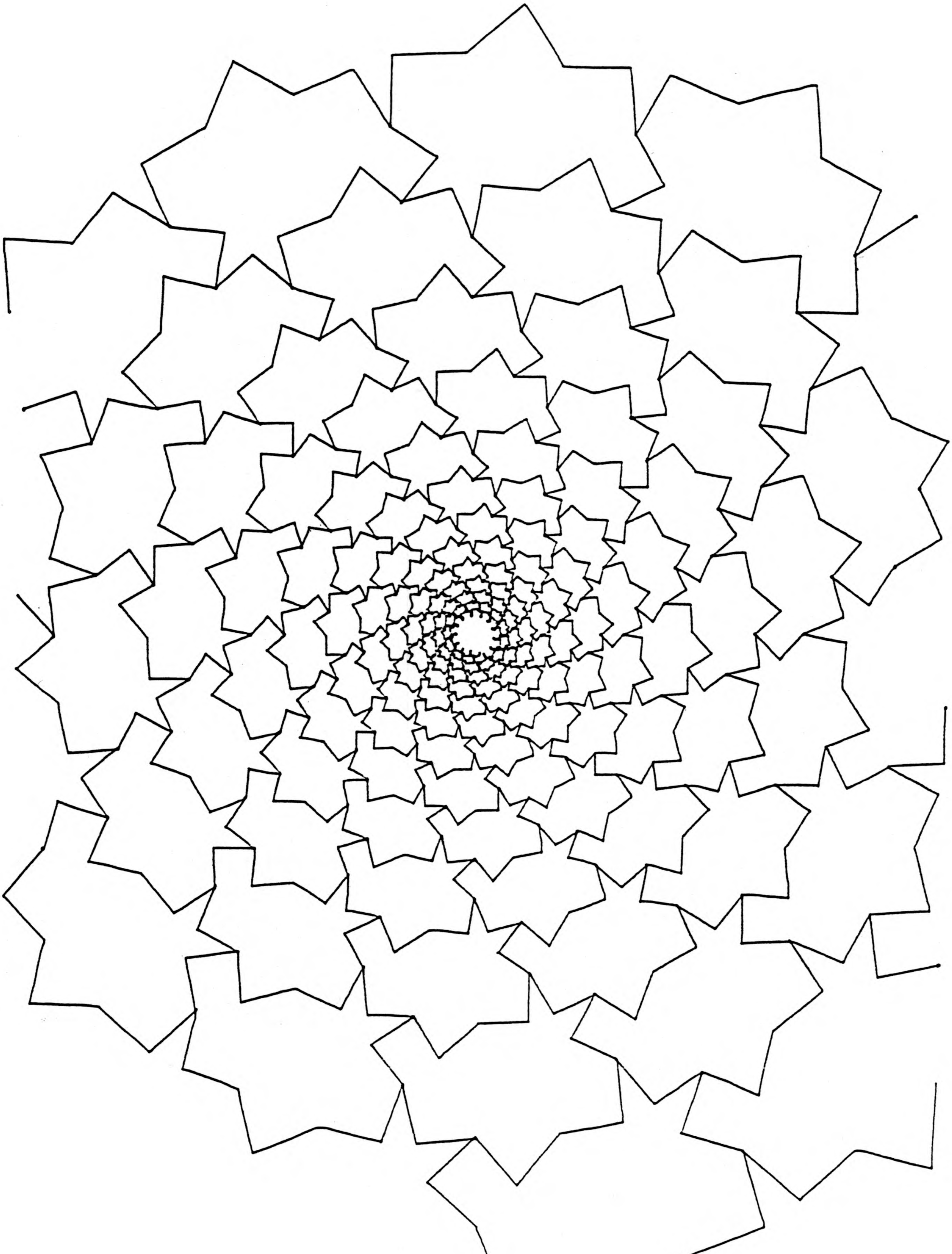
$18 \times 3 - 24 \times 2 = 6$

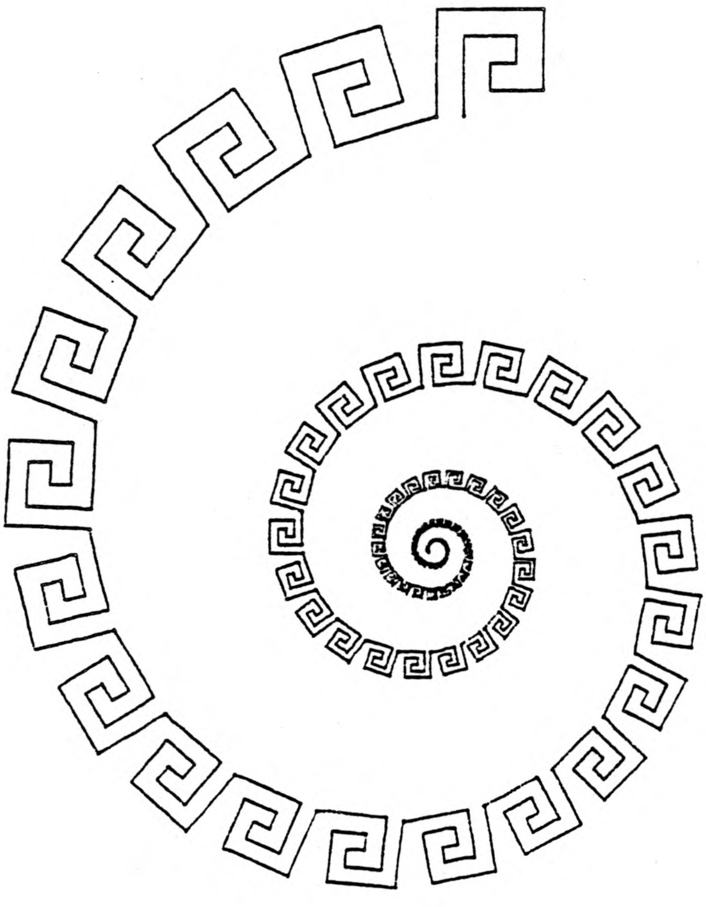
SIMILITUDE

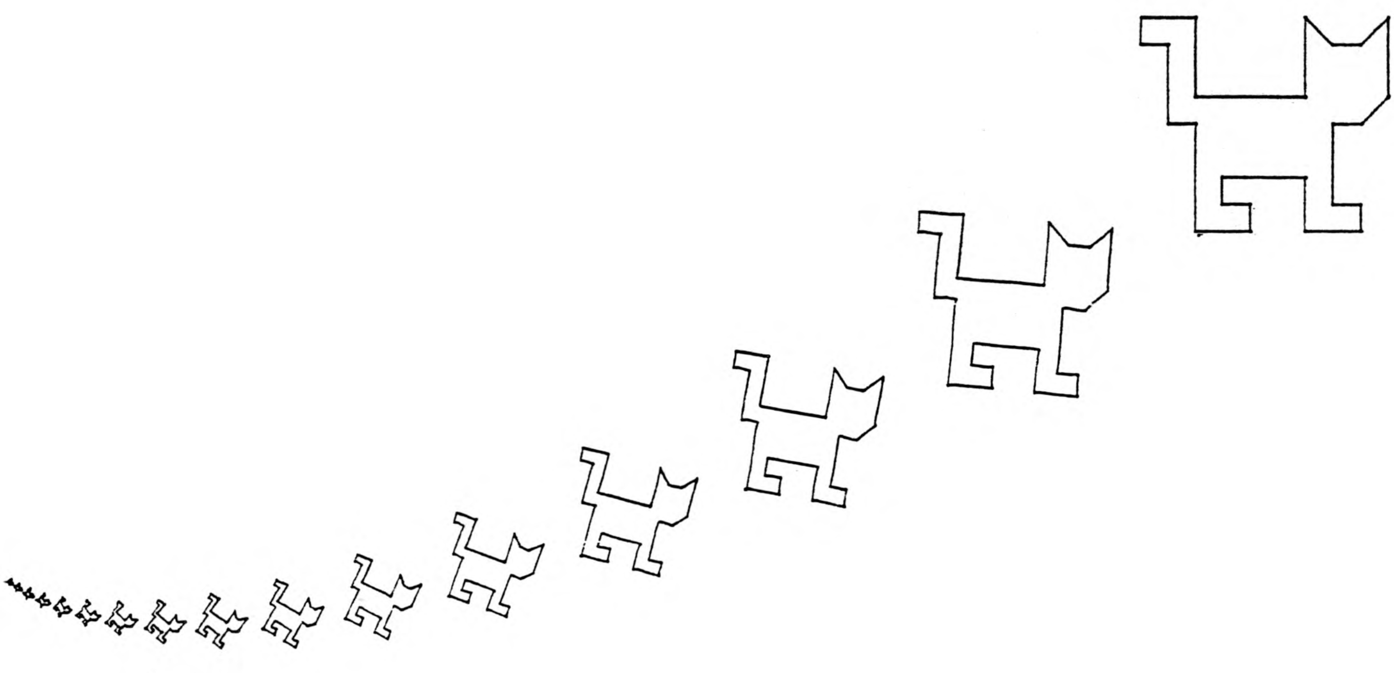
Lorsqu'on dispose d'une table traçante, on peut visualiser des transformations du plan. La figure choisie devra être une ligne polygonale ouverte ou fermée, et on lui fait subir une similitude de centre 0 et de rapport k. En répétant le programme, on obtient une suite de figures.


IREM


Introduire le cadrage
Introduire le rapport k et l'angle α
Introduire le nombre N de points du polygone
i : = 0
i : = i + 1
Introduire les coordonnées (x_i, y_i) de M_i
$i < N$
non
oui
i : = 1
Amener le stylet en M_1 sans tracer
i : = i + 1
Amener le stylet en M_i en traçant
$i < N$
non
oui
Amener le stylet en M_1 en traçant
Cette instruction sera sautée pour une ligne polygonale ouverte.
i : = 0
i : = i + 1
Effectuer la similitude sur les coordonnées de M_i
$i < N$
non
oui







TRACE D'UN RAYON LUMINEUX A TRAVERS UN SYSTEME CENTRE DE NEUF LENTILLES MAXIMUM (sur H.P. 10)

Chacun des 18 dioptres sphériques est donné par son centre, son rayon ; le rayon est un nombre réel ce qui permet de préciser dans quel sens est tournée la concavité du dioptre par rapport au sens de la lumière. A chaque traversée de dioptre l'indice de réfraction peut changer.

Le problème se décompose alors en plusieurs étapes.

1) Stockage des données (centre, rayon, indice).

2) Tracé des dioptres.

3) Tracé d'un rayon lumineux.

Ce rayon peut partir de n'importe quel point de la feuille, dont on précise les coordonnées, et dans une direction donnée par l'angle qu'il fait avec l'axe optique.

O_1 (a_1, O) est le centre du premier dioptre.

A_0 (α_0, β_0) est le point de départ.

$\theta_0 = (\overrightarrow{Ox}, \overrightarrow{A_0A_1})$ est l'angle initial du rayon et de l'axe optique.

$i_1 = (\overrightarrow{O_1A_1}, \overrightarrow{A_0A_1})$.

$j_1 = (\overrightarrow{O_1A_1}, \overrightarrow{A_1A_2})$.

$\theta_1 = (\overrightarrow{Ox}, \overrightarrow{A_1A_2})$.

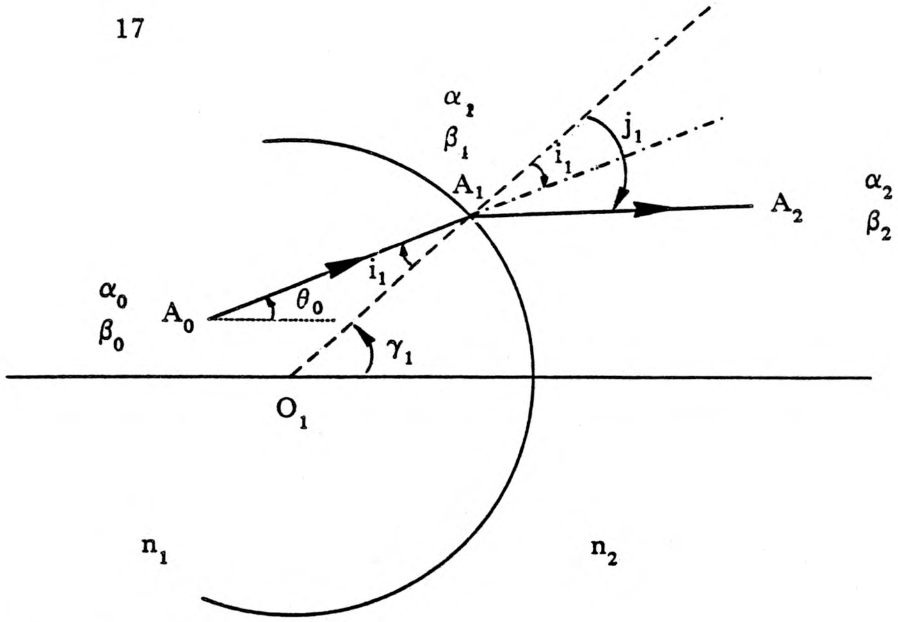
$\gamma_1 = (\overrightarrow{Ox}, \overrightarrow{O_1A_1})$.

$i_1 = (\overrightarrow{O_1A_1}, \overrightarrow{A_0A_1})$

$j_1 = (\overrightarrow{O_1A_1}, \overrightarrow{A_1A_2})$

$\theta_1 = (\overrightarrow{Ox}, \overrightarrow{A_1A_2})$

$\gamma_1 = (\overrightarrow{Ox}, \overrightarrow{O_1A_1})$



Calcul des coordonnées de A_1.

Equation de A_0A_1 : $y = \mathrm{tg}\,\theta_0 \times x + (\beta_0 - \alpha_0\,\mathrm{tg}\,\theta_0)$.

Equation du cercle : $(x - a_1)^2 + y^2 = R_1^2$.

Pour trouver α_1 on est amené à résoudre l'équation
$x^2(1 + \mathrm{tg}^2\,\theta_0) + 2x\,[\,\mathrm{tg}\,\theta_0\,(\beta_0 - \alpha_0\,\mathrm{tg}\,\theta_0) - a_1] + a_1^2 + (\beta_0 - \alpha_0\,\mathrm{tg}\,\theta_0)^2 - R_1^2 = 0$ qui,
en posant : $A = 1 + \mathrm{tg}^2\,\theta_0$ $\quad B = \mathrm{tg}\,\theta_0\,(\beta_0 - \alpha_0\,\mathrm{tg}\,\theta_0) - a_1$ $\quad C = a_1^2 + (\beta_0 - \alpha_0\,\mathrm{tg}\,\theta_0)^2 -$
donne les solutions

$$x_1 = \frac{-B - \sqrt{\Delta'}}{A} \qquad x_2 = \frac{-B + \sqrt{\Delta'}}{A}.$$

On prendra pour α_1 la valeur :

$$\alpha_1 = \frac{-B - \frac{R_1}{|R_1|}\sqrt{\Delta'}}{A}$$

Calcul de θ_1 :

$$\theta_1 = (\overrightarrow{Ox}, \overrightarrow{A_1A_2}) = (\overrightarrow{Ox}, \overrightarrow{O_1A_1}) + (\overrightarrow{O_1A_1}, \overrightarrow{A_1A_2})$$

$$\theta_1 = \gamma_1 + j_1$$

$$\gamma_1 = \mathrm{Arc\ tg}\,\frac{\beta_1}{\alpha_1 - a_1}$$

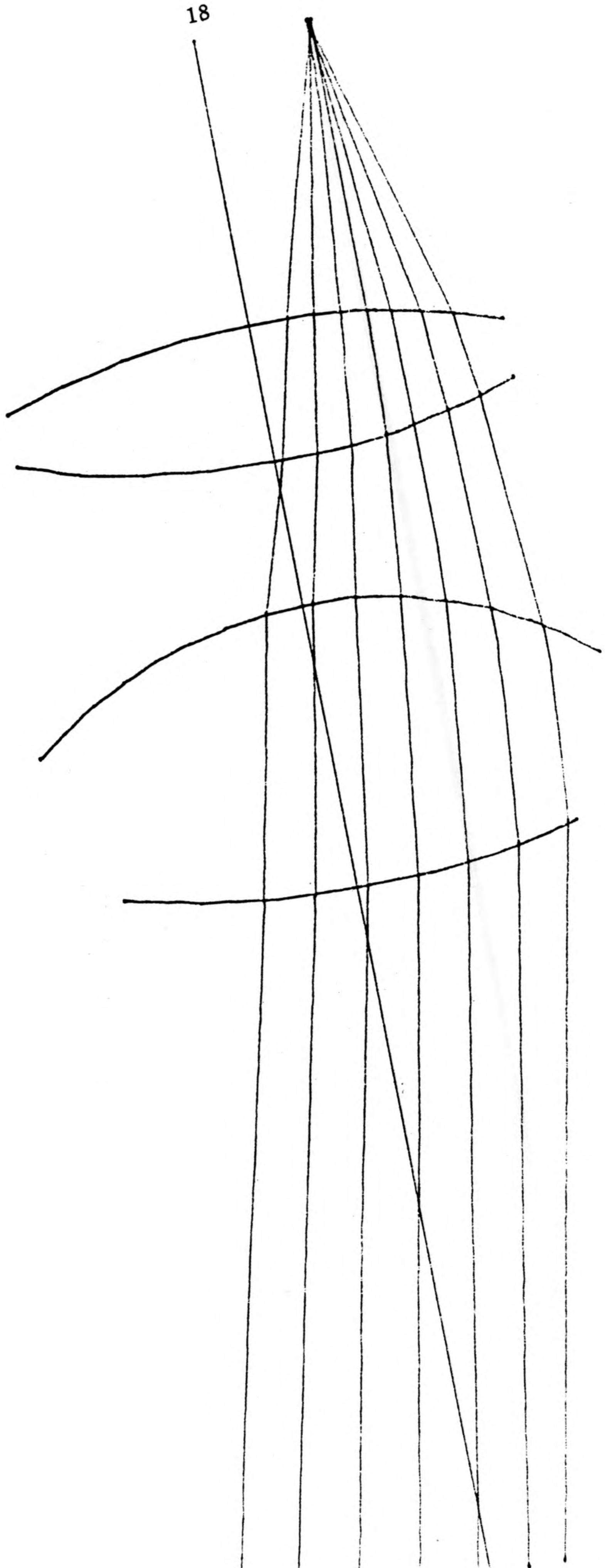
$$n_1 \sin i_1 = n_2 \sin j_1$$

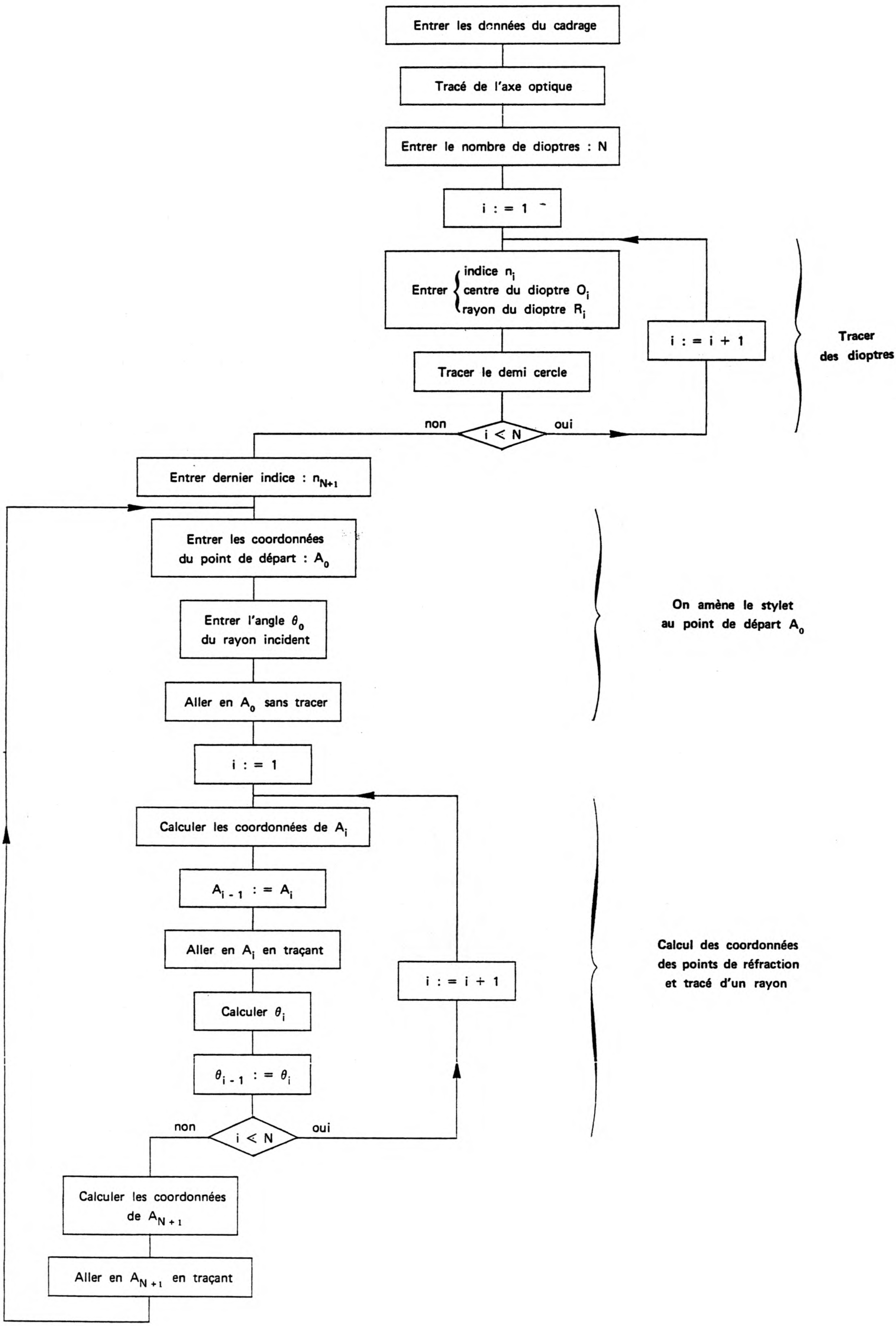
$$\theta_1 = \mathrm{Arc\ tg}\,\frac{\beta_1}{\alpha_1 - a_1} + \mathrm{Arc\ sin}\,(\frac{n_1}{n_2}\sin i_1)$$

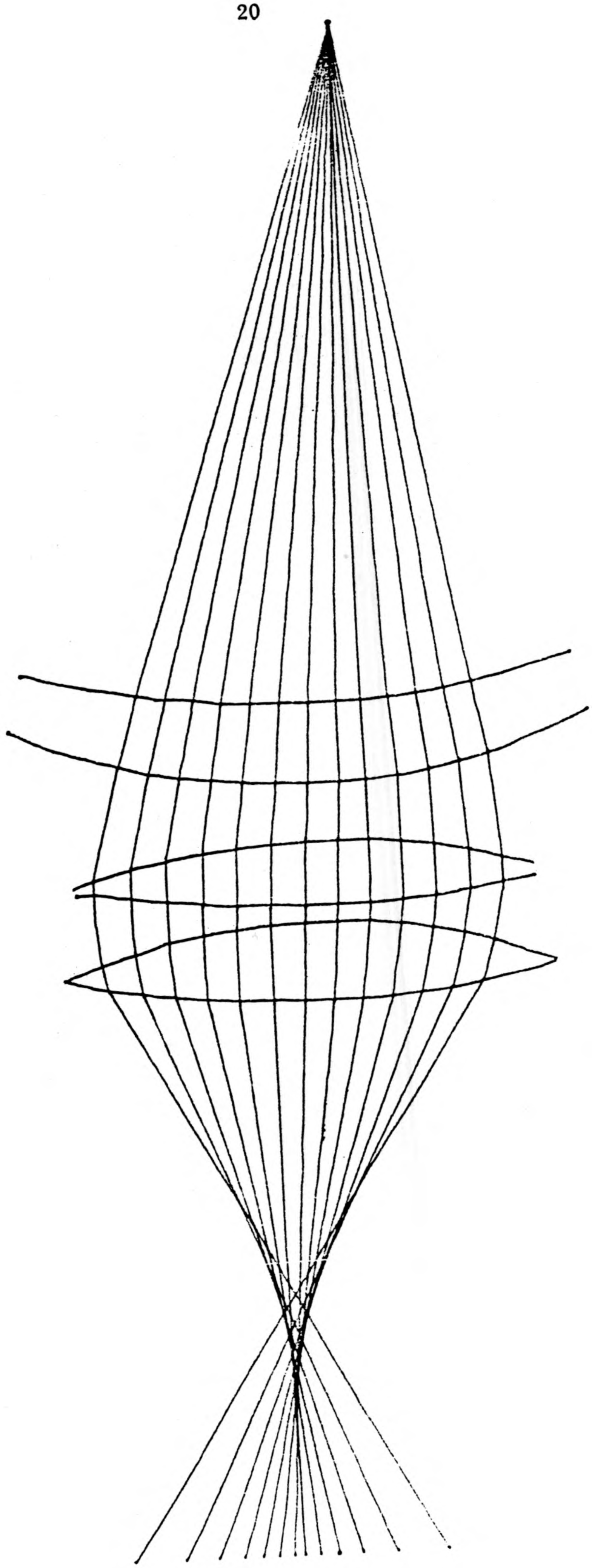
On repart ensuite de A_1 avec l'angle θ_1.

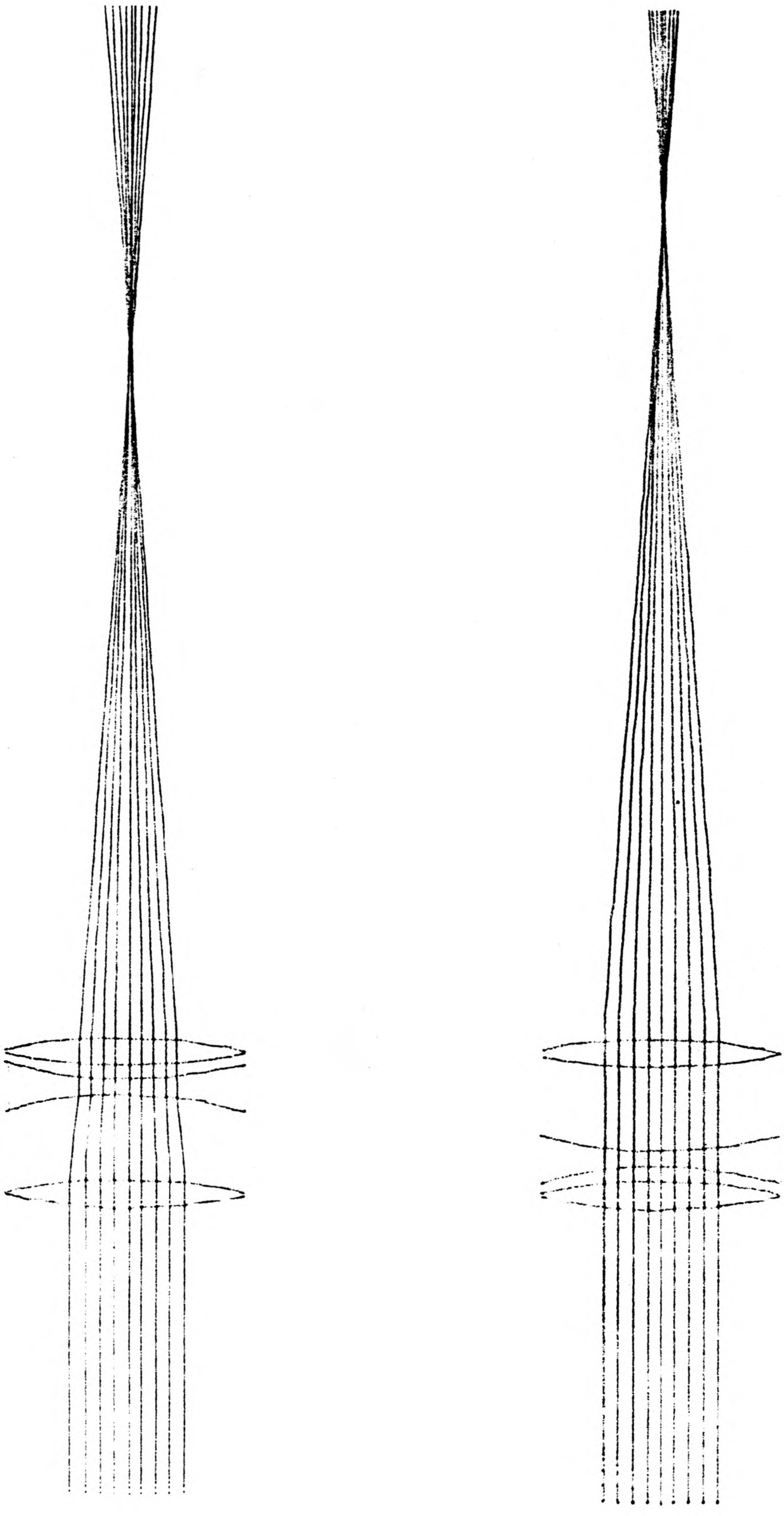
Lorsqu'un rayon est tracé le programme boucle sur l'entrée du point de départ A_0 et de l'angle θ_0 de façon que l'on puisse tracer à partir d'un même point plusieurs rayons avec des angles d'incudence différents.

18




Entrer les données du cadrage
Tracé de l'axe optique
Entrer le nombre de dioptres : N
i : = 1
Entrer indice n_i, centre du dioptre O_i, rayon du dioptre R_i
i : = i + 1
Tracer le demi cercle
non
i < N
oui
Tracer des dioptres
Entrer dernier indice : n_{N+1}
Entrer les coordonnées du point de départ : A_0
Entrer l'angle θ_0 du rayon incident
Aller en A_0 sans tracer
On amène le stylet au point de départ A_0
i : = 1
Calculer les coordonnées de A_i
$A_{i-1} := A_i$
Aller en A_i en traçant
i : = i + 1
Calculer θ_i
$\theta_{i-1} := \theta_i$
Calcul des coordonnées des points de réfraction et tracé d'un rayon
non
i < N
oui
Calculer les coordonnées de A_{N+1}
Aller en A_{N+1} en traçant





CHAMP ELECTRIQUE

Un corps chargé éléctriquement placé en un point A crée en tout point un champ éléctrique. Pour étudier ce champ éléctrique les physiciens utilisent un champ de vecteurs. Si d est la distance du point A à un point M, le vecteur correspondant à l'action de la charge sur le point M est $\overrightarrow{E_M}$ tel que :

$$\overrightarrow{E_M} = \frac{q}{d^2} \times \frac{\overrightarrow{AM}}{\| \overrightarrow{AM} \|} = \frac{q}{d^3} \overrightarrow{AM}$$

q désignant la charge placée en A.

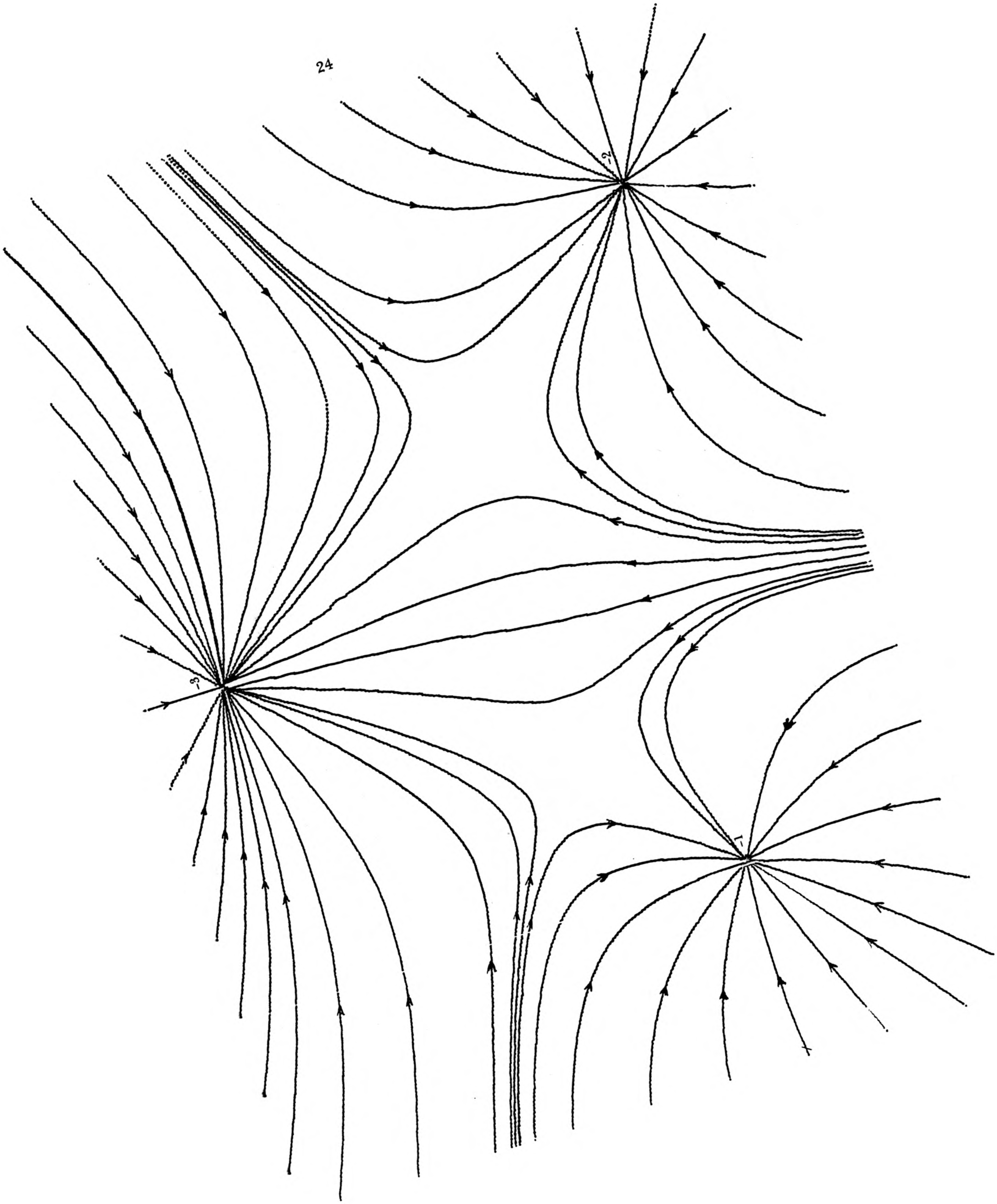
Si on place plusieurs charges éléctriques q_1, ... q_n en des points A_1, ... A_n, le champ créé en M est la résultante des champs créés en M par chacune des charges :

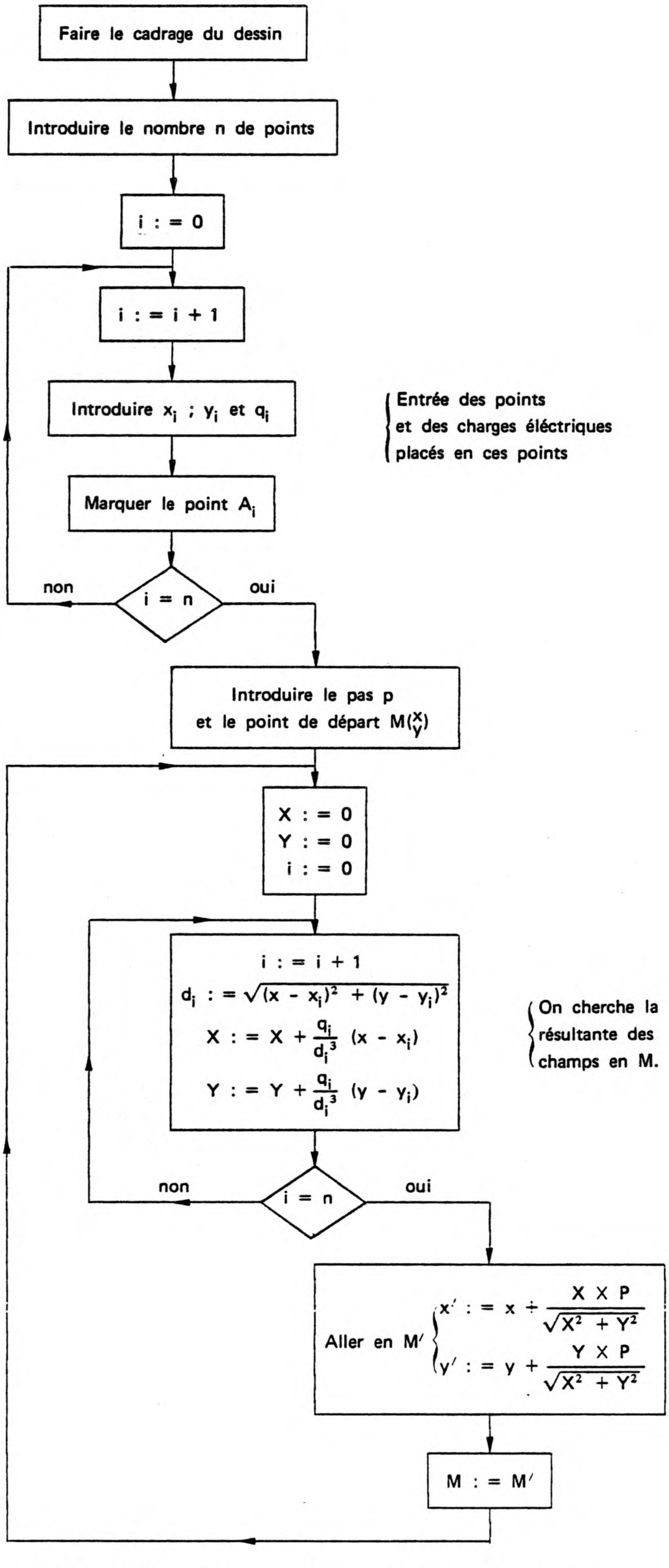
$$\overrightarrow{E_M} = \sum_{i=1}^{n} \frac{q_i}{d_i{}^3} \overrightarrow{A_iM}.$$

On appelle «ligne de champ» les courbes tangentes au vecteur champ en chacun de leurs points. La table traçante ne permet de construire que des lignes polygonales, voisines de ces lignes de champ ; on détermine le champ en M, puis on déplace le stylet de M en M′ tel que $\overrightarrow{MM'}$ soit colinéaire à $\overrightarrow{E_M}$, la distance de M à M′ pouvant être, soit fixée, soit proportionnelle à l'intensité du champ. Ensuite on recommence pour M′.

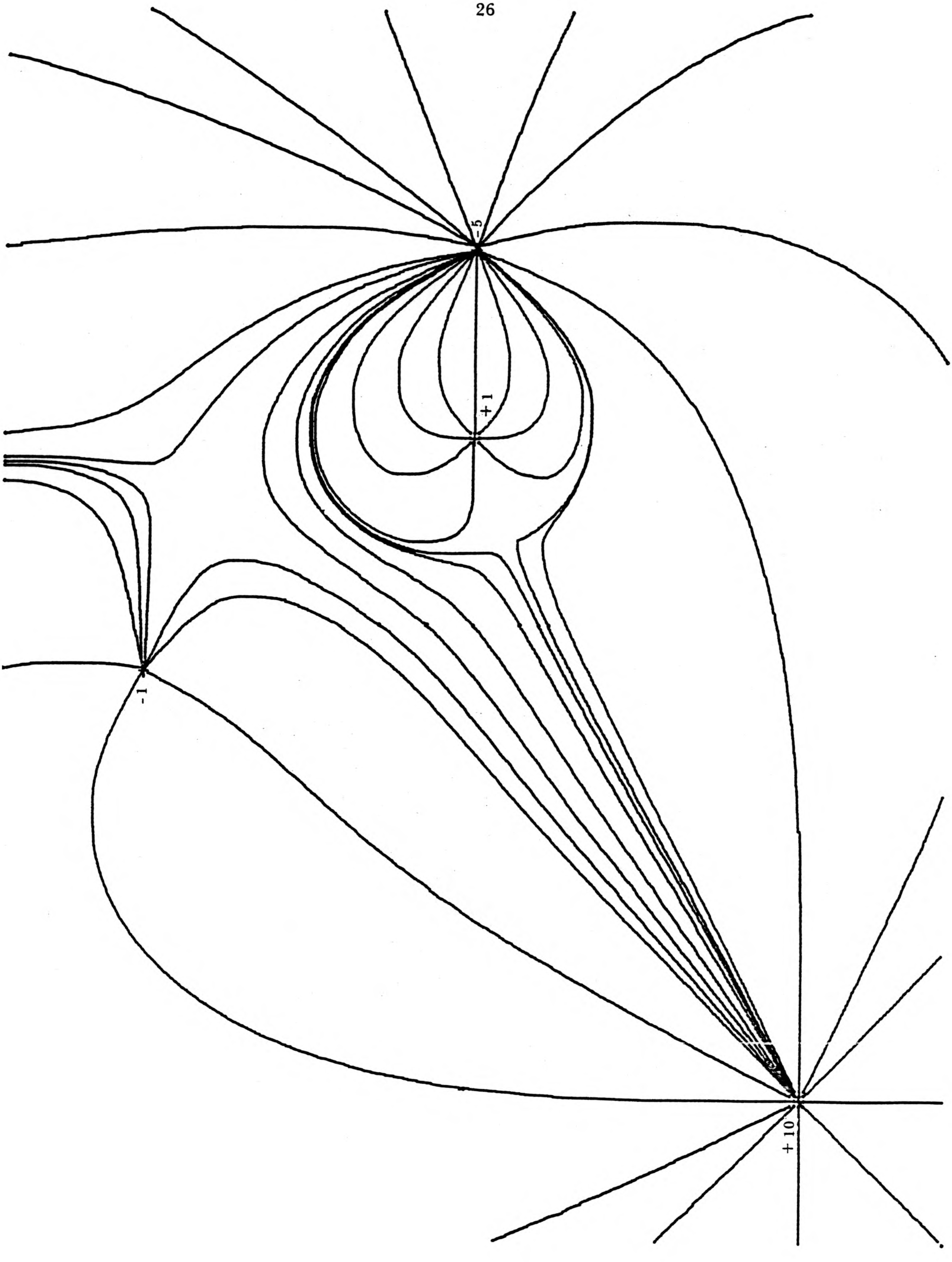
Remarque.

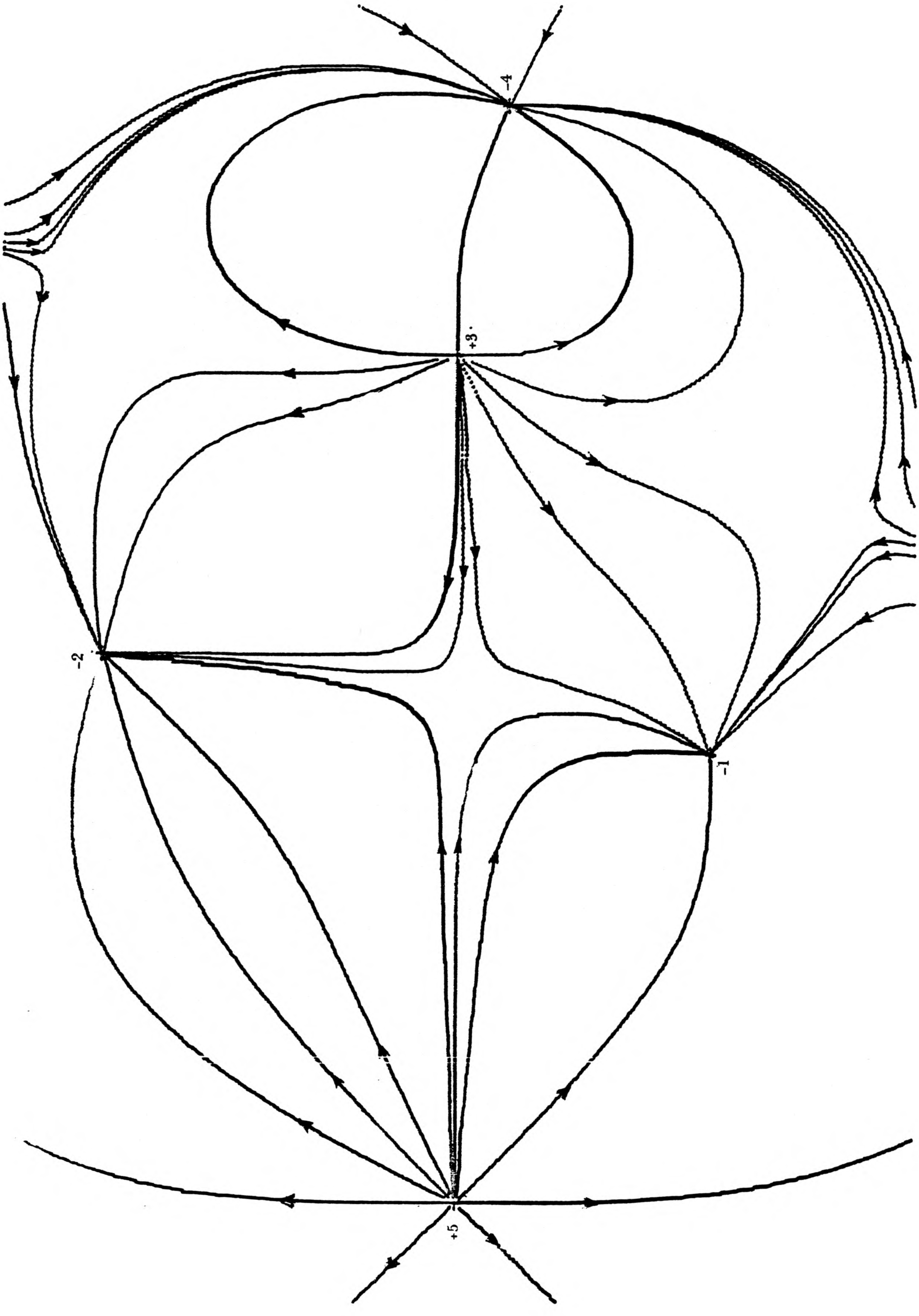
Ceci s'applique aussi aux lignes d'induction d'un champ magnétique.


-2
-3
-1



Pour avoir un pas variable, on introduit un coefficient k à la place de p, le point M' est alors le point de coordonnées $\begin{cases} x' = x + kX \\ y' = y + kY \end{cases}$


-5
+1
-1
+10


-4
+3
-2
-1
+5

ETUDES DE COURBES AU VOISINAGE D'UN POINT
MISE EN EVIDENCE DE DISCONTINUITES

I – INTRODUCTION.

Rappel de la définition choisie pour la continuité de f en x_0 : pour tout ϵ strictement positif, existe-t-il α strictement positif tel que pour tout $x, |x - x_0| < \alpha$ entraine $|f(x) - f(x_0)| < \epsilon$.

Problème lors de l'utilisation de la machine.

La construction de la courbe (C) nécessite le choix d'un pas «p», on ne peut donc pratiquement que comparer $f(x_0 + p)$ à $f(x_0)$.

ϵ ne peut pratiquement être choisi de manière arbitraire : il dépend de p. On peut seulement montrer que, en certains points, il existe ϵ appartenant à $\mathbb{R}$ tel que $|f(x_0 + p) - f(x_0)| > \epsilon$.

Remarques.

Si le tracé de f semble continu, il n'est pas prouvé que f est ou n'est pas continue. Il faudrait modifier simultanément et arbitrairement p et ϵ, ce qui semble pratiquement irréalisable.

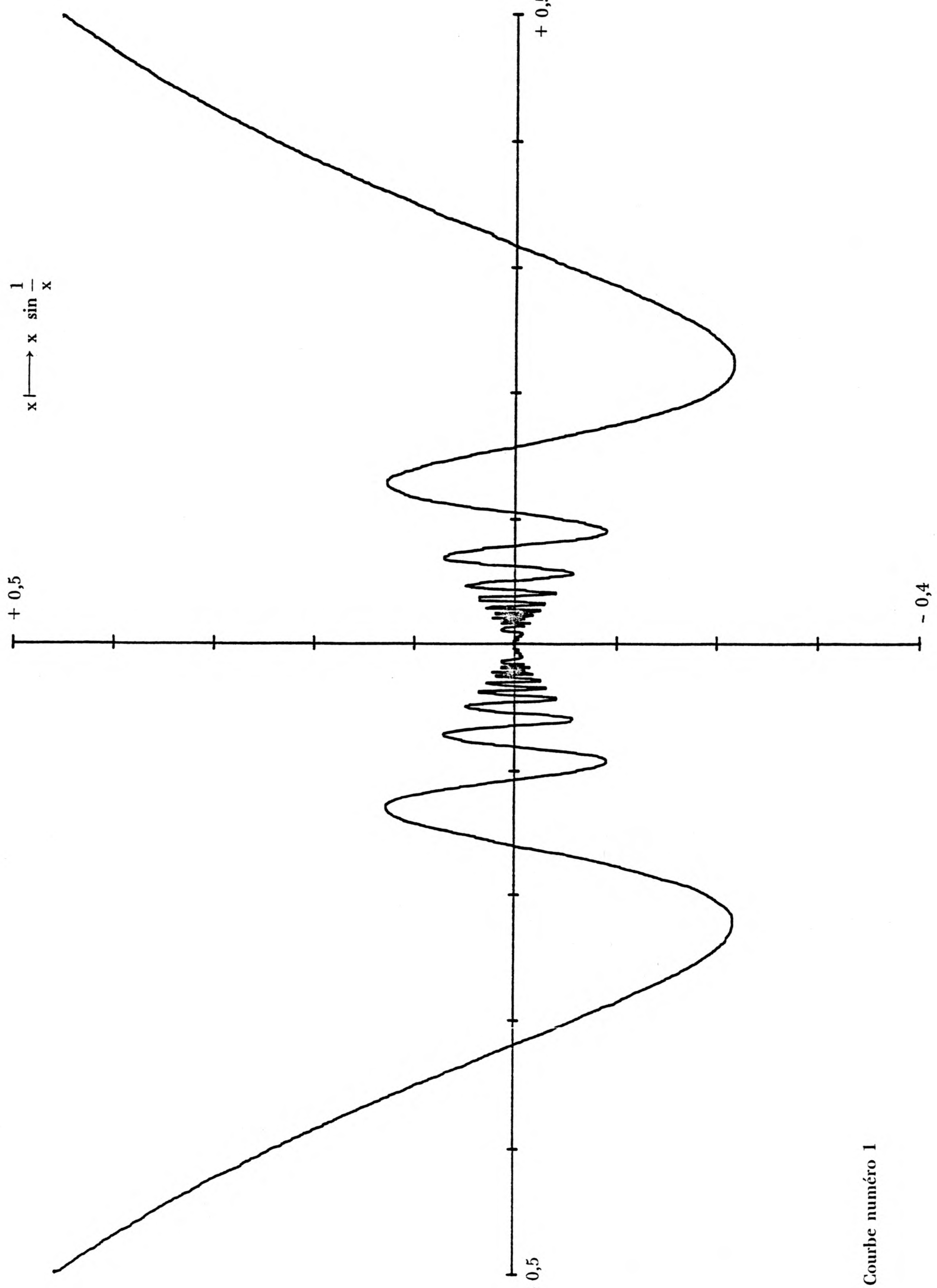
La machine ne prouve rien en ce qui concerne la continuité.

On montrera dans certains des exemples qui suivent que, p étant initialement choisi, des choix différents de ϵ peuvent conduire à des tracés différents.

II – METHODE GENERALE.

1. Programmation d'un tracé de courbe.

Méthode classique.



Courbe numéro 1

2. Introduction des données.

x_0 valeur initiale de la variable x.

p : «pas» de la courbe.

ϵ : choisi en fonction de l'ordre de grandeur de p (légèrement supérieur à kp, k dépendant de $f'(x_0)$).

3. Programmation de la fonction étudiée.

Calcul de $f(x_0 + p)$.

4. Comparaison de $|f(x_0 + p) - f(x_0)|$ à ϵ.

Lorsque $|f(x_0 + p) - f(x_0)| > \epsilon$, impose au stylet de se relever (tracé discontinu).

III – EXEMPLE I. (Courbe numéro 1).

$x \xmapsto{f} x \sin \frac{1}{x}$. Etude de f au voisinage de 0.

IV – EXEMPLE II.

$x \xmapsto{f} x\,[1 - E(\frac{1}{x})]$. Etude de f au voisinage de $\frac{1}{q}$ où $q \in Z$ avec $-1{,}0 \leqslant x \leqslant 0{,}6$.

Courbe numéro 2.

Pas $p = 10^{-3}$

$\epsilon = 5 \times 10^{-1}$.

Le tracé semble continu sur l'intervalle considéré $\doteq$ sur tout l'intervalle on a $|f(x_0 + p) - f(x_0)| < 5 \times 10^{-1}$.

Exemple :

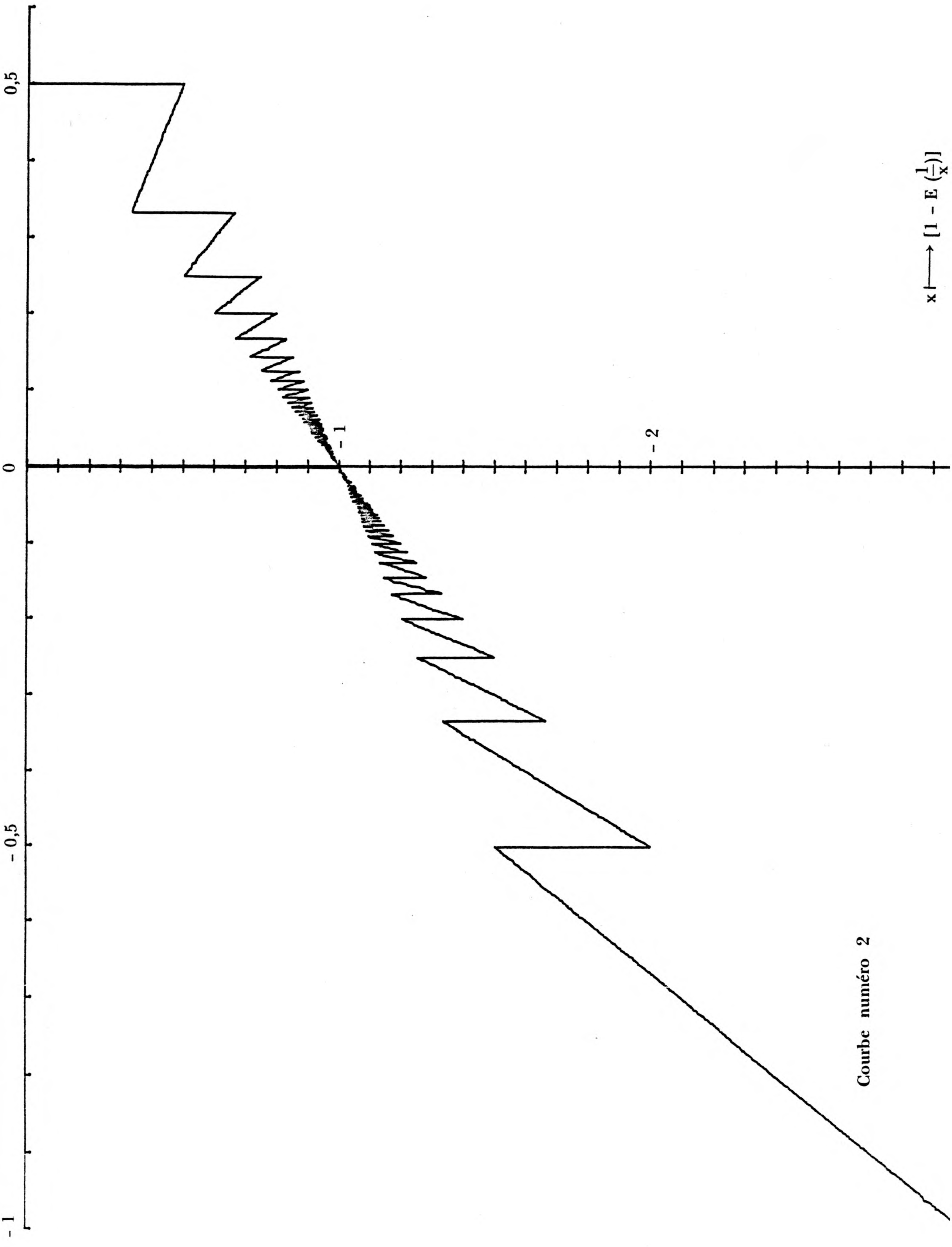
$x_0 = -\frac{1}{2} \qquad x_0 + p = -\frac{1}{2} + \frac{1}{1\,000} \qquad |f(x_0 + p) - f(x_0)| = 0{,}466$

donc $|f(x_0 + p) - f(x_0)| < 0{,}5$.

Courbe numéro 3.

Pas $p = 10^{-3}$

$\epsilon = 5 \times 10^{-2}$



$x \longmapsto [1 - E(\frac{1}{x})]$

Courbe numéro 2

Le tracé fait nettement apparaître les discontunuités aux points d'abscisse $x_0 = \frac{1}{q}$ sauf à partir du moment où $| f(x_0 + p) - f(x_0) | < 5 \times 10^{-2}$ ce qui se produit pour des valeurs de x inférieures à $\frac{1}{10}$.

Exemple :

$x_0 = \frac{1}{20}$ $\quad x_0 + p = \frac{1}{20} + \frac{1}{1\,000}$.

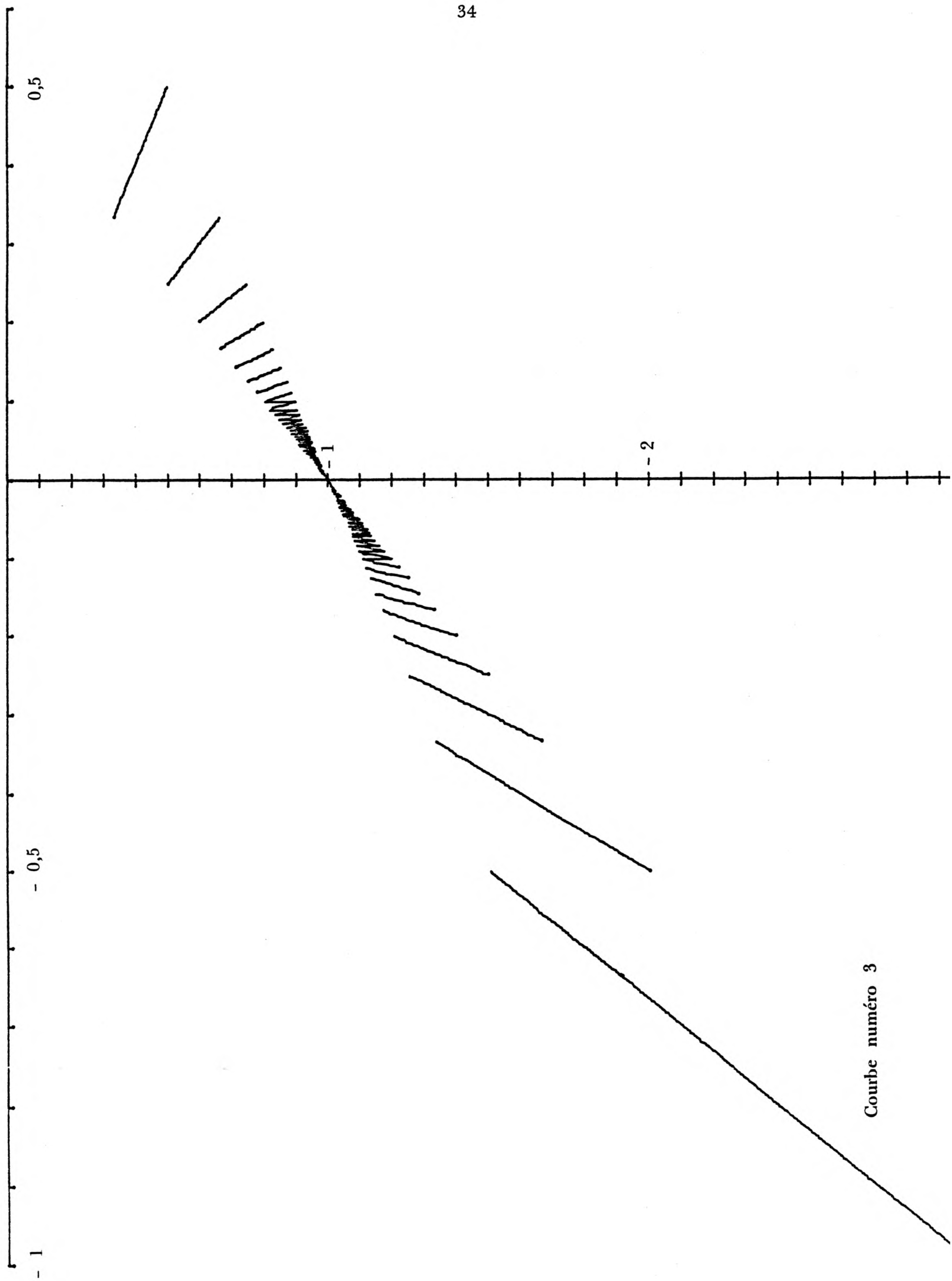
On trouve $f(x_0) = 0{,}950$ $\quad f(x_0 + p) = -0{,}918$ $\quad |f(x_0 + p) - f(x_0)| < 4 \times 10^{-2}$ donc $| f(x_0 + p) - f(x_0) | < 5 \times 10^{-2}$.

Courbe numéro 4.

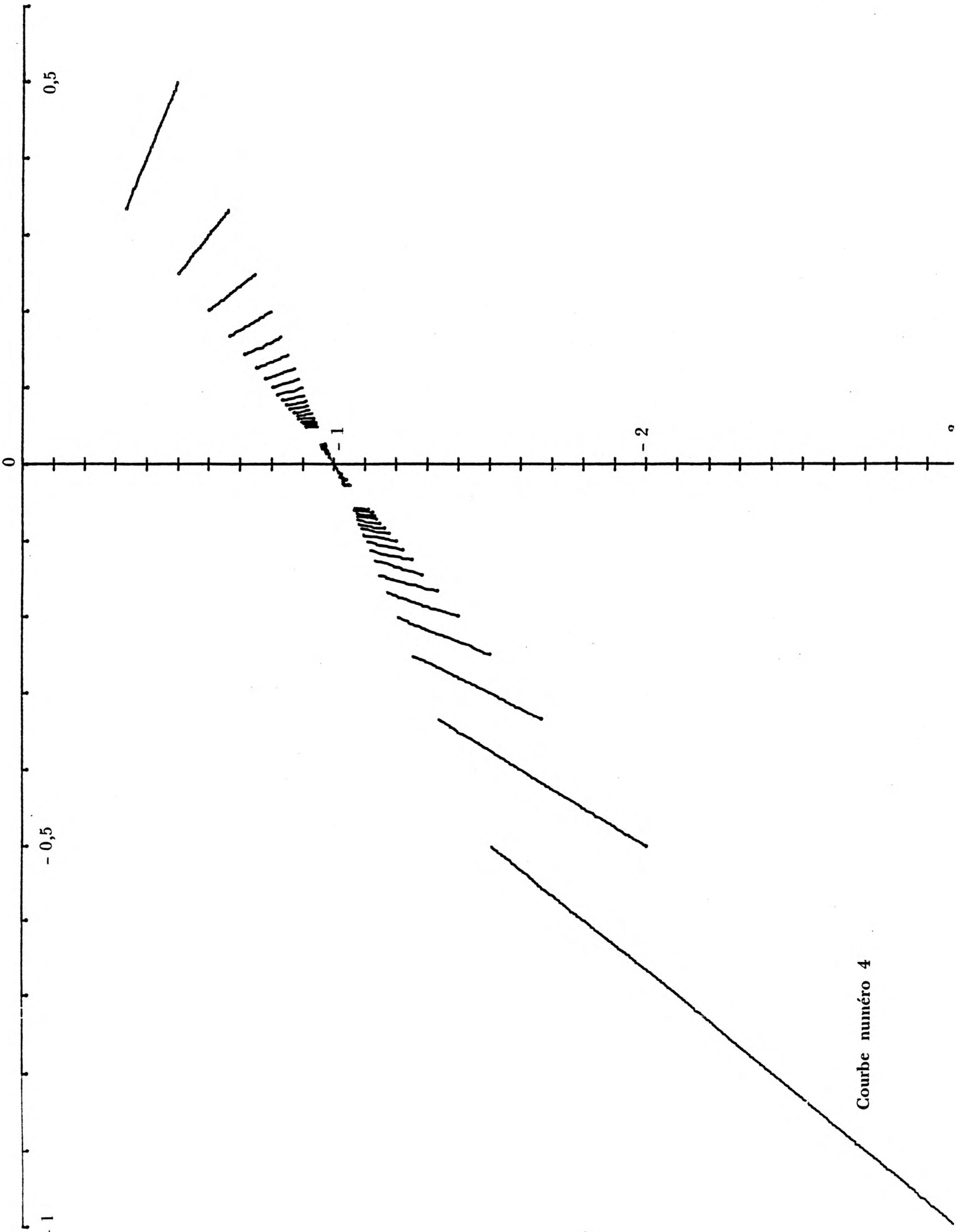
Pas $p = 5 \times 10^{-4}$

$\epsilon = 1 \times 10^{-2}$

Il y a une zone où aucun tracé n'est fait car les points sont à une distance supérieure à un centième.



Courbe numéro 3



Courbe numéro 4

VISUALISATION DES DROITES D'UN PLAN FINI

But.

Il s'agit de montrer aux élèves qu'il existe d'autres plans que le plan physique habituel, et que dans ce plan, les droites, telles qu'elles ont été définies par les axiomes d'incidence, ont un tracé particulier.

Méthode.

Deux points M et N du plan, de coordonnées (A, B) et (A_1, B_1), étant choisis, la droite passant par les points M et N, est l'ensemble des points T du plan tels que :

$$\overrightarrow{MT} = a \, . \, \overrightarrow{MN}$$

Tous les points du plan sont successivement étudiés, chaque point appartenant à la droite étant aussitôt représenté par le symbole «X».

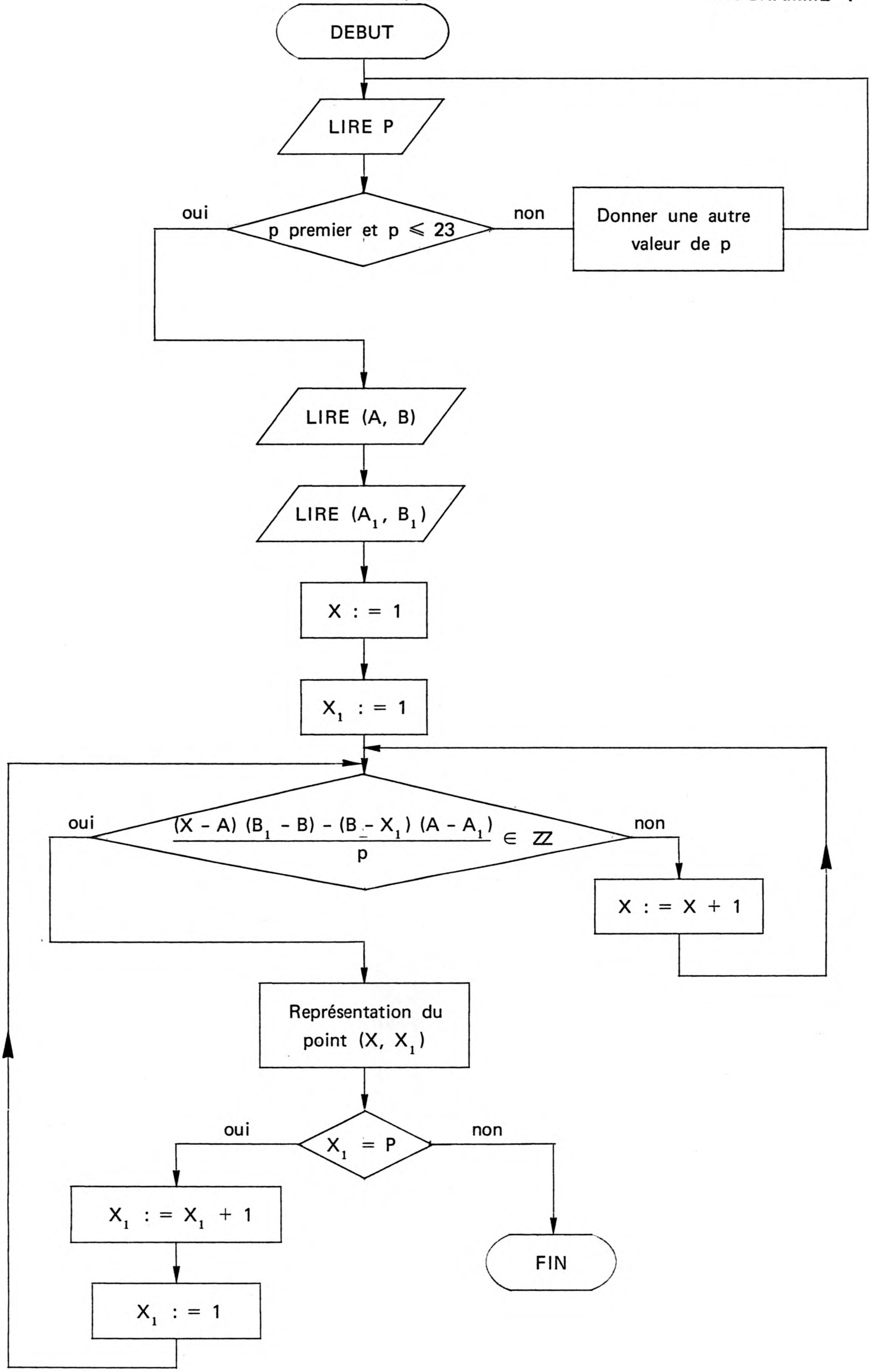
Propositions d'utilisation du programme.

a) Vérification de l'axiome d'Euclide : (pour cet exercice, il est préférable de choisir $p < 10$). Une droite D_1 étant tracée à l'aide du premier programme, choisir un point K extérieur à cette droite. A l'aide du deuxième programme, tracer d'une part D_1, d'autre part toutes les droites contenant K. L'existence et l'unicité de la droite contenant K et disjointe avec D_1 seront alors vérifiées.

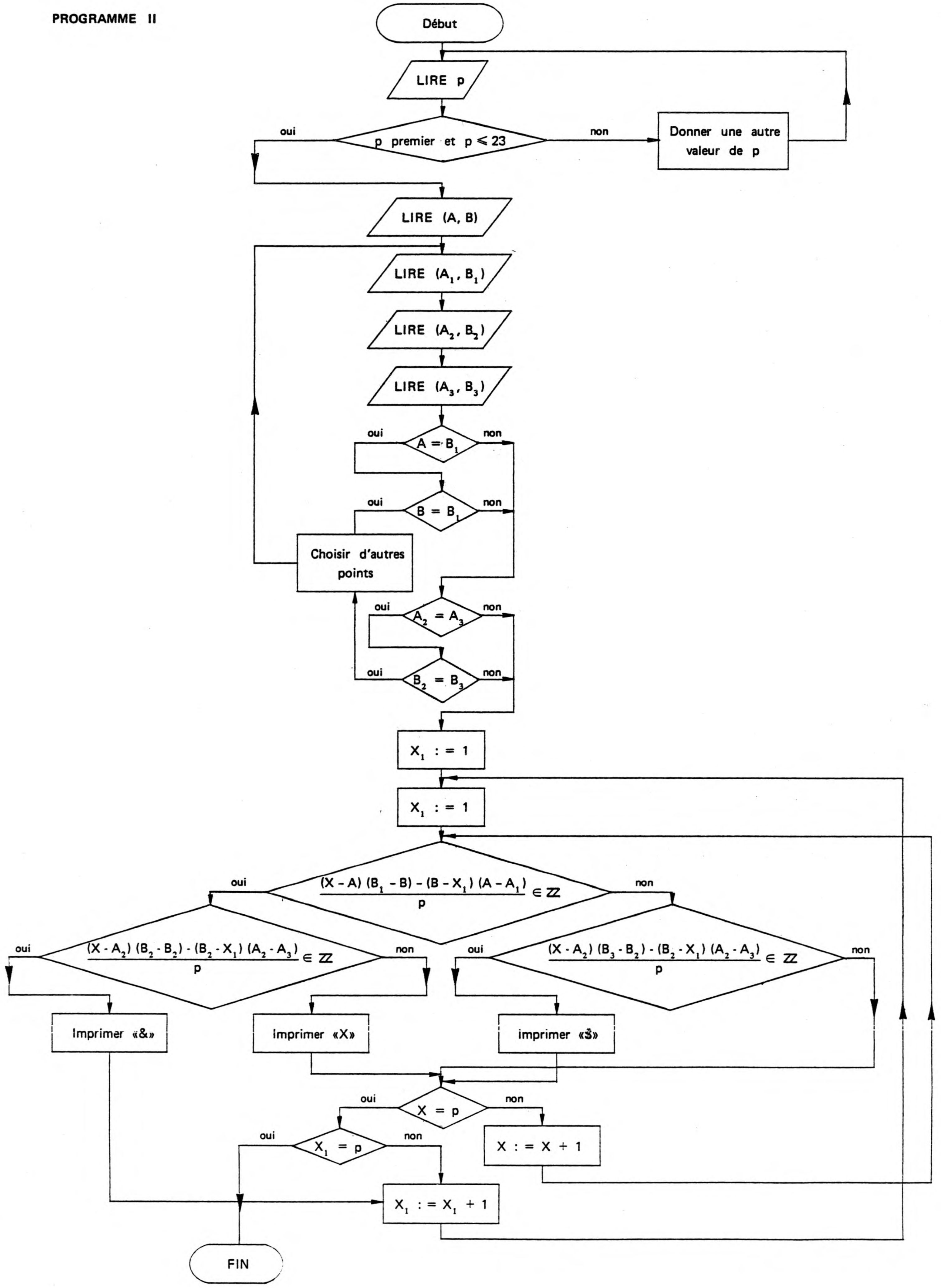
Il est également possible de faire cet exercice à l'aide de papier calque.

b) Deux droites ayant deux points communs sont confondues.

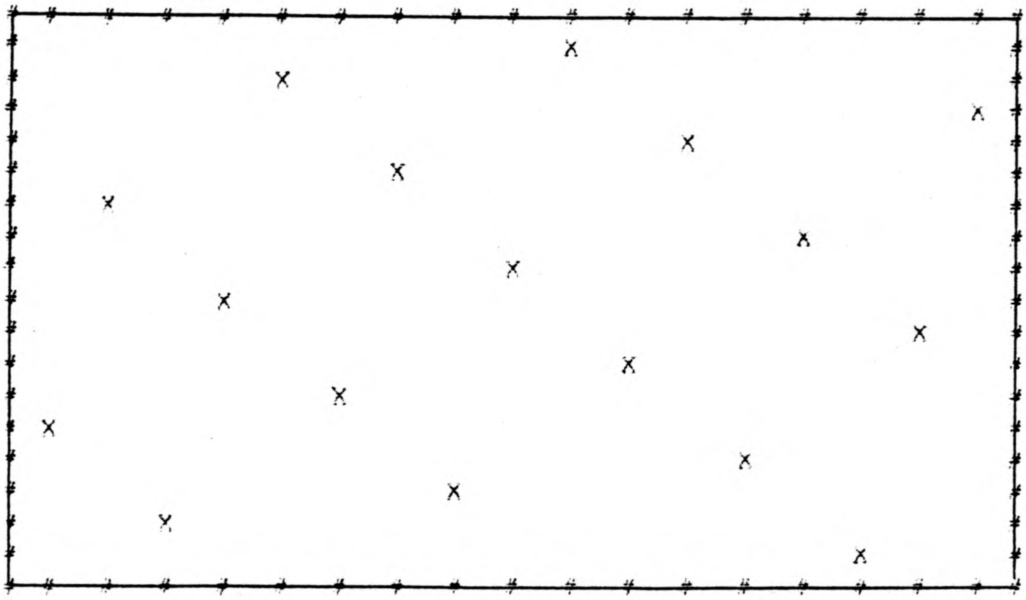
c) Détermination de la parallèle à une droite donnée passant par un point donné (suite logique de a)) : parmi toutes les droites contenant K, choisir la droite parallèle à D_1 par simple lecture des coordonnées des points.


DEBUT
LIRE P
p premier et p ⩽ 23
oui
non
Donner une autre valeur de p
LIRE (A, B)
LIRE (A_1, B_1)
X : = 1
X_1 : = 1
oui
$\frac{(X - A)(B_1 - B) - (B - X_1)(A - A_1)}{p} \in \mathbb{Z}$
non
X : = X + 1
Représentation du point (X, X_1)
oui
X_1 = P
non
X_1 : = X_1 + 1
X_1 : = 1
FIN

PROGRAMME II

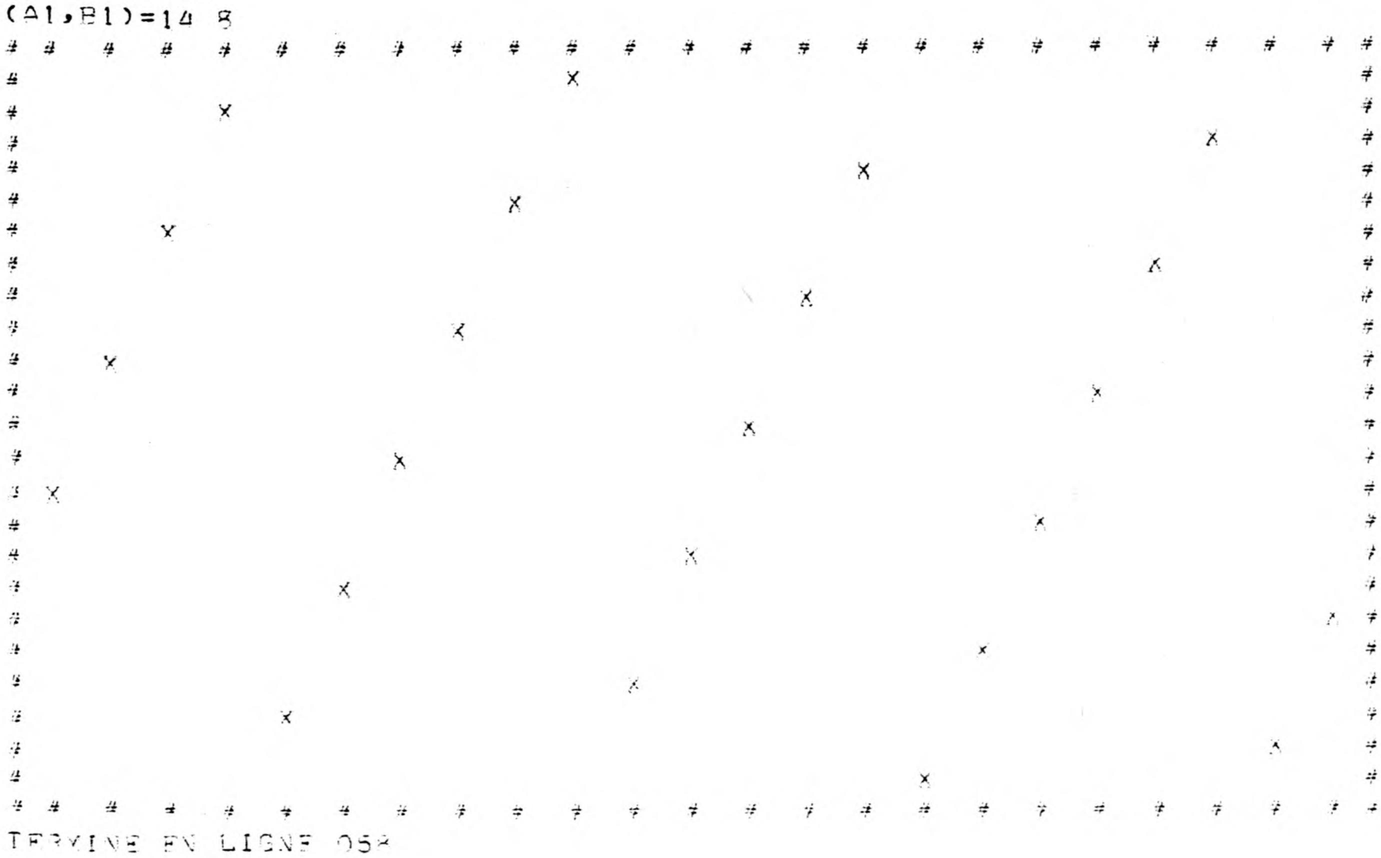


```
EXECUTER A PARTIR DE 1
                    VISUALISATION DES DROITES D'UN PLAN FINI
CE PROGRAMME PERMET DE VISUALISER UNE DROITE DANS UN
PLAN FINI, CONNAISSANT LES COORDONNEES DE DEUX POINTS
DE CETTE DROITE
 P AU CARRE REPRESENTE LE NOMBRE DE POINTS DU PLAN
P=17
(A,B)=2 6
(A1,B1)=4 9
```



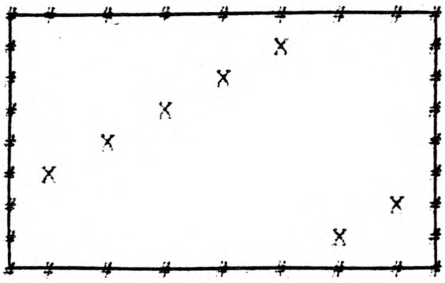
```
TERMINE EN LIGNE 058
```

```
EXECUTER A PARTIR DE 1
                    VISUALISATION DES DROITES D'UN PLAN FINI
CE PROGRAMME PERMET DE VISUALISER UNE DROITE DANS UN
PLAN FINI, CONNAISSANT LES COORDONNEES DE DEUX POINTS
DE CETTE DROITE
 P AU CARRE REPRESENTE LE NOMBRE DE POINTS DU PLAN
P=23
(A,B)=12 16
(A1,B1)=14 8
```



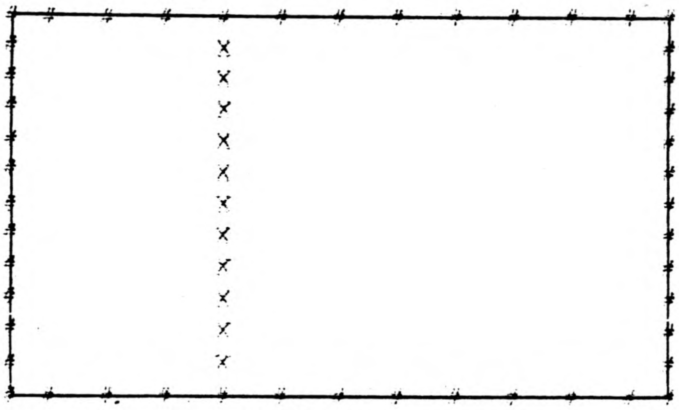
```
TERMINE EN LIGNE 058
```

```
EXECUTER A PARTIR DE 1
                        VISUALISATION DES DROITES D'UN PLAN FINI
CE PROGRAMME PERMET DE VISUALISER UNE DROITE DANS UN
PLAN FINI, CONNAISSANT LES COORDONNEES DE DEUX POINTS
DE CETTE DROITE
 P AU CARRE REPRESENTE LE NOMBRE DE POINTS DU.PLAN
P=12
LA VALEUR DE P DOIT ETRE UN NOMBRE PREMIER INFERIEUR A 25
DONNE UNE AUTRE VALEUR DE P
P=7
(A,B)=1 5
(A1,B1)=2 4
```



```
TERMINE EN LIGNE 058
```

```
EXECUTER A PARTIR DE 1
                        VISUALISATION DES DROITES D'UN PLAN FINI
CE PROGRAMME PERMET DE VISUALISER UNE DROITE DANS UN
PLAN FINI, CONNAISSANT LES COORDONNEES DE DEUX POINTS
DE CETTE DROITE
 P AU CARRE REPRESENTE LE NOMBRE DE POINTS DU PLAN
P=11
(A,B)=4 8
(A1,B1)=4 10
```



```
TERMINE EN LIGNE 058
```

```
EXECUTER A PARTIR DE 1
                        VISUALISATION DES DROITES D'UN PLAN FINI
CE PROGRAMME PERMET DE VISUALISER DEUX DROITES D'UN PLAN
FINI, CONNAISSANT LES COORDONNEES DE DEUX POINTS DISTINCTS
 POUR CHACUNE D'ELLES.
 P AU CARRE REPRESENTE LE NOMBRE DE POINTS DU PLAN
 ON NOTERA & LEUR POINT D'INTERSECTION.
 LE NOMBRE DE POINTS DU PLAN EST EGAL A P AU CARRE
P=15
 LA VALEUR DE P DOIT ETRE UN NOMBRE PREMIER
 INFERIEUR A 25. DONNE UNE AUTRE VALEUR DE P
P=13
(A,B)=2 9
(A1,B1)=4 7
(A2,B2)=12 11
(A3,B3)=3 10S
# # # # # # # # # # # # # # #
#                   X     $ #
#                 &         #
#         $     X           #
# $           X             #
#           X       $       #
#         X $               #
#   $   X                   #
#     X               $     #
#   X         $             #
# X   $                     #
#                       $ X #
#               $       X   #
#       $             X     #
# # # # # # # # # # # # # # #
TERMINE EN LIGNE 078


EXECUTER A PARTIR DE 1
                        VISUALISATION DES DROITES D'UN PLAN FINI
CE PROGRAMME PERMET DE VISUALISER DEUX DROITES D'UN PLAN
FINI, CONNAISSANT LES COORDONNEES DE DEUX POINTS DISTINCTS
 POUR CHACUNE D'ELLES.
 P AU CARRE REPRESENTE LE NOMBRE DE POINTS DU PLAN
 ON NOTERA & LEUR POINT D'INTERSECTION.
 LE NOMBRE DE POINTS DU PLAN EST EGAL A P AU CARRE
P=11
(A,B)=3 8
(A1,B1)=6 10
(A2,B2)=3 8
(A3,B3)=6 10
# # # # # # # # # # # # #
#                 &     #
#         &             #
# &                     #
#               &       #
#       &               #
#                     & #
#             &         #
#     &                 #
#                   &   #
#           &           #
#   &                   #
# # # # # # # # # # # # #
TERMINE EN LIGNE 078
```

EXPERIENCE MINIORDINATEUR EN QUATRIEME

Conditions de l'expérience.

Pour l'année scolaire 1976-1977, l'I.R.E.M. de Grenoble a proposé une expérience d'utilisation d'un miniordinateur en classe de quatrième. Des professeurs de deux établissements se sont portés volontaires : certains du Lycée Hector Berlioz de La Côte Saint André, les autres du Lycée Jean Prévost de Villard de Lans. L'I.R.E.M. fournissant l'appareil, une H.P. 10 avec sa table traçante, il a été décidé de partager l'année en périodes de six semaines environ pendant lesquelles l'appareil serait dans l'un ou dans l'autre établissement. A La Côte Saint André deux classes sont concernées par cette expérience, tandis qu'à Villard de Lans ce sont trois classes.

Apprentissage de la programmation.

Les première séances, qui ont été pour les classes bénéficiant de dédoublement les heures de travaux dirigés, ont été consacrées à l'apprentissage des modes opératoires de la calculatrice, le passage des calculs numériques aux calculs de valeurs numériques d'une expression contenant une lettre est ainsi mis en évidence, puisque la même séquence peut être utilisée après avoir mis dans les registres d'autres nombres. De même l'utilité d'un programme apparaît à ce stade. Des élèves ont ainsi réalisé les programmes de calcul des valeurs numériques d'un polynôme du second degré. Ceci a alors amené assez naturellement la recherche des approximations successives d'une racine d'un tel polynôme. Avec la table traçante, on a fait construire point par point le graphique de la fonction correspondante ce qui a rendu une telle recherche plus «parlante».

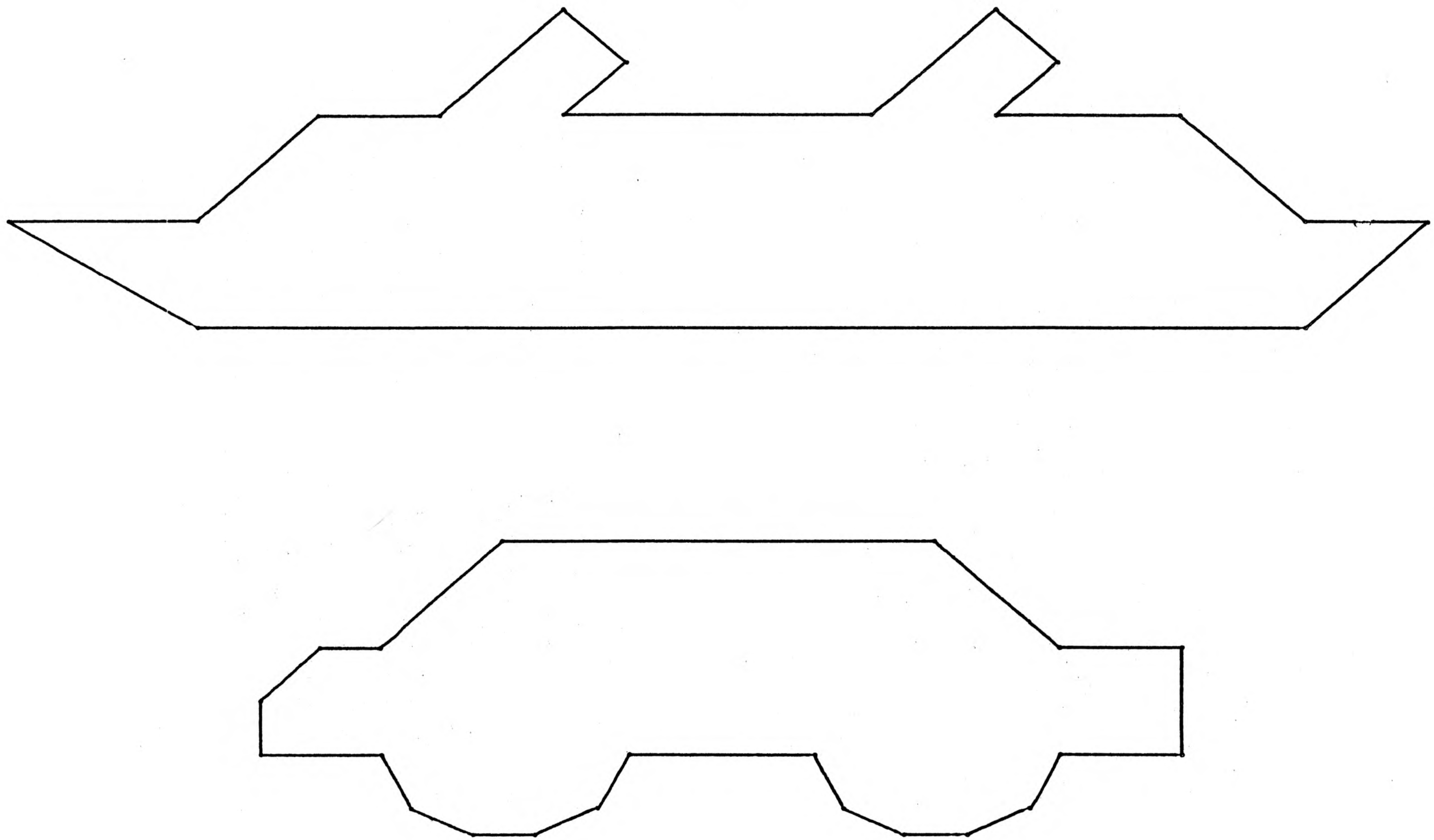
Utilisation en géométrie de l'ordinateur.

Il a tout d'abord été fait un controle de l'acquisition de la notion de coordonnées : les élèves ont pris un dessin de leur choix sur un quadrillage et relevé les coordonnées des divers points. Puis ils ont essayé de faire reproduire leur dessin par l'ordinateur. Certains eurent des surprises !! Pour ne pas avoir à apprendre aux élèves le codage de la traçante, nous avions préparé un programme : les élèves n'avaient à introduire que les coordonnées de leurs points dans l'ordre habituel.

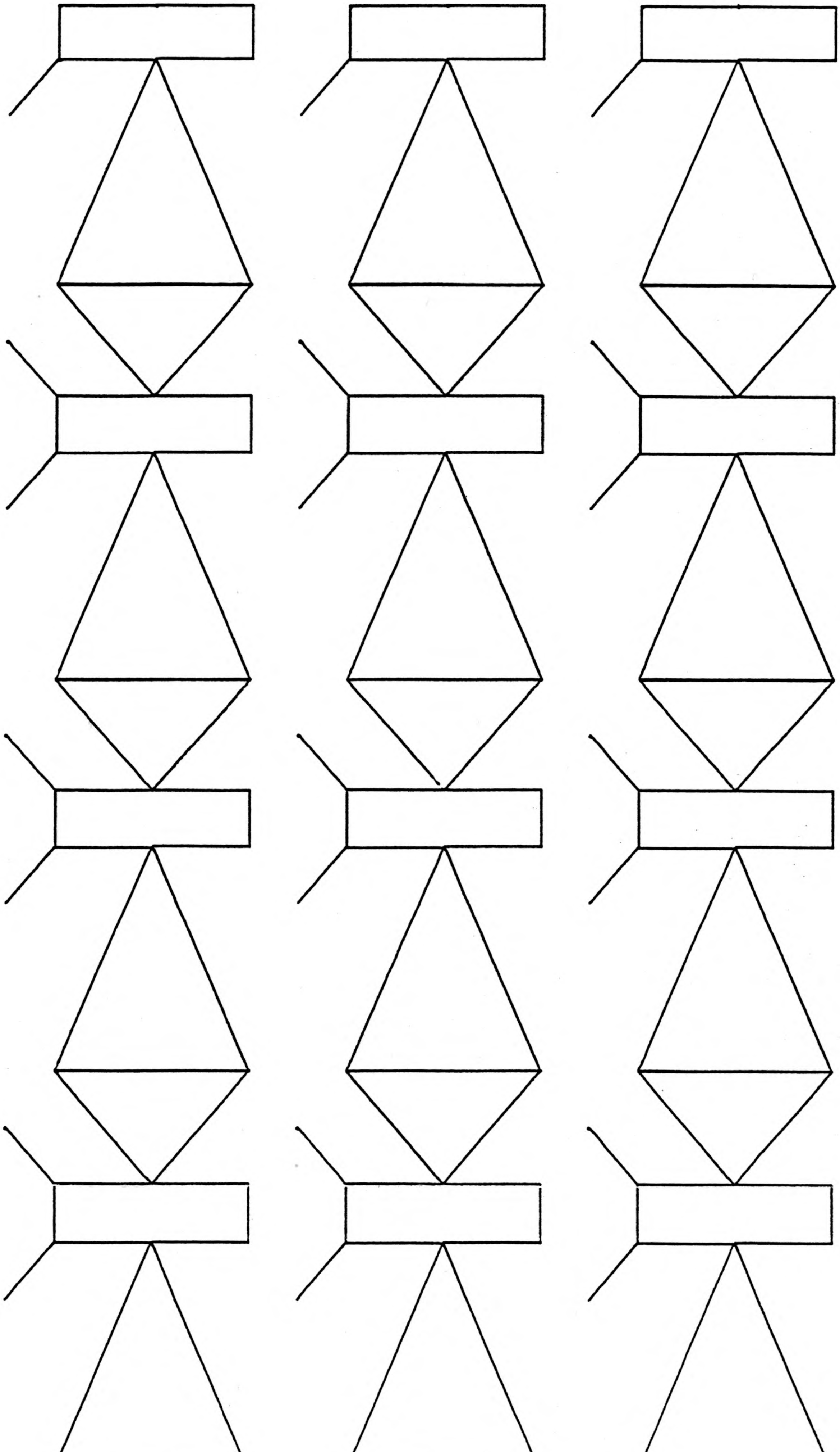
Les transformations du plan sont aussi beaucoup plus facilement perçues lorsque l'on voit toute une figure être transformée. Nous avons utilisé des translations et des similitudes. La composition des translations, la commutativité ou l'associativité de cette composition, l'existence d'une réciproque sont alors perceptibles. Enfin nous avons prévu une approche de l'axiome de Thalès qui sera utilisé plus tard dans l'année : étant donné deux droites graduées AB et CD, on recherche l'abscisse du projeté d'un point. L'élève propose sa solution qui est acceptée ou rejetée suivant qu'elle est exacte ou à 1/100 près, ou à plus de 1/100.

Conclusion.

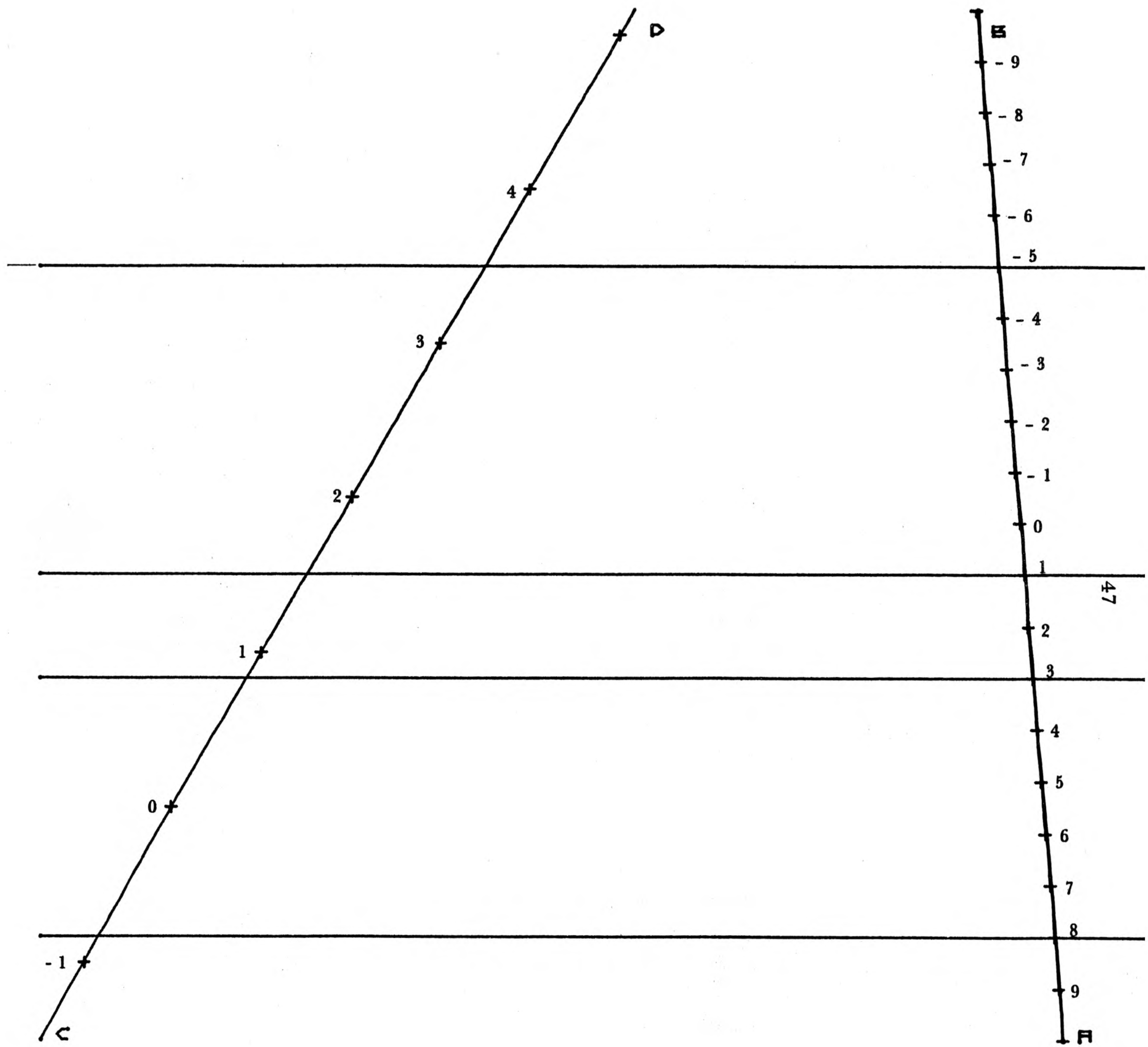
Nous avons essayé d'utiliser le miniordinateur pour le calcul algébrique et pour visualiser certains problèmes de géométrie, sans sortir du cadre du programme de la classe. Si les élèves ont réalisé eux mêmes quelques programmes, pour les cas plus compliqués ils n'ont fait qu'utiliser un programme déjà fait. Cependant nous pensons qu'ils ont pu se rendre compte qu'un ordinateur ne peut rien faire d'autre que ce qu'on lui a demandé de faire, et nous pensons que «démythifier» l'ordinateur est un aspect non négligeable de notre expérience.



Exemples de dessins réalisés par des élèves de 4ème



GRAD. D'UN PT
DE AB =
8,00
GRAD. DU
PROJETE ?
- 0,84
VALEUR EXACTE
- 0,8333333
GRAD. D'UN PT
DE AB =
3,00
GRAD. DU
PROJETE ?
0,85
NON
0,84
VALEUR EXACTE
0,8333333
GRAD. D'UN PT
DE AB =
1,00
GRAD. DU
PROJETE ?
1,50
BRAVO
GRAD. D'UN PT
DE AB =
- 5,00
GRAD. DU
PROJETE ?
3,40
NON
3,50
BRAVO.



ESSAIS D'OPTIMISATION D'UN PROGRAMME DE CALCUL DE C_n^P

Ce travail a été fait au cours d'un stage d'initiation à l'emploi d'une Hewlett Packard 30. Les stagiaires étaient des professeurs de mathématiques, avec le concours d'un professeur d'électronique et de deux professeurs de dessin industriel.

Première Phase.

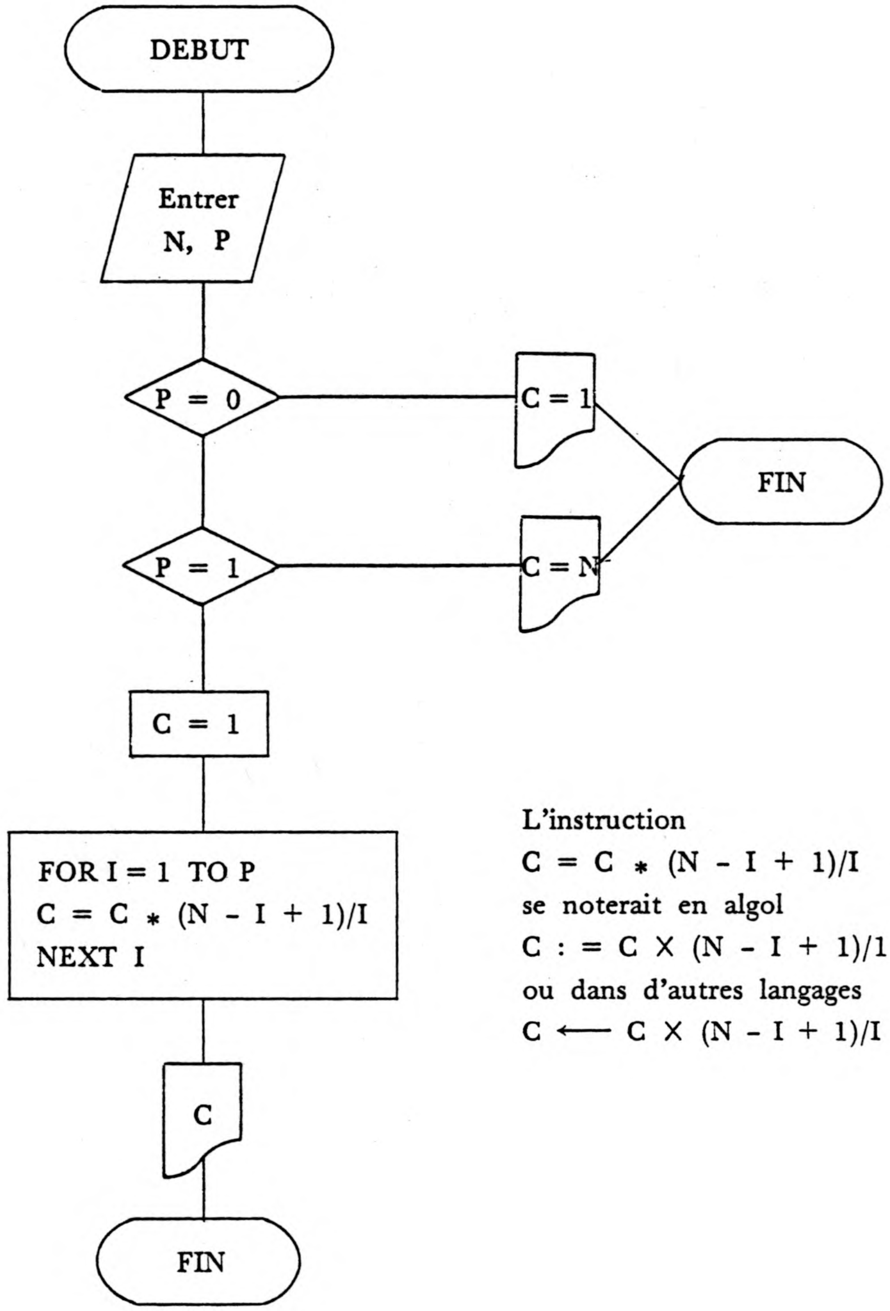
Après avoir rencontré, en début de stage, la notion de sous-programme, une collègue (enseignant en terminale C) fait un programme de calcul de $C_n^P \; \dfrac{n\,!}{P\,!\;(n-1)\,!}$ ce qui demande un sous-programme de calcul de factoriel et 3 appels de ce sous-programme. En basic, c'est élémentaire, je n'en donne donc pas l'organigramme. A l'exécution, on teste ce programme avec l'idée de déterminer expérimentalement les limites de la machine. On obtient un résultat avec tous les chiffres significatifs jusqu'à C_{37}^{16} puis un résultat en «notation scientifique» (c'est-à-dire un nombre compris entre 0 et 10, avec un maximum de 11 chiffres après la virgule) multiplié par une puissance de 10. Ce résultat n'est pas satisfaisant puisqu'il est tronqué. Mais on peut ainsi calculer jusqu'à C_{69}^{35}. Impossible de calculer C_{70}^{1} dont nous connaissons pourtant la valeur !...

En réfléchissant à ce dépassement de capacité qui ne se produit que si un nombre est de valeur absolue supérieure à $9{,}99999999999 \times 10^{99}$ on arrive à conclure que l'ennemi est factoriel 70 car C_{70}^{1} est acceptable. Dans le calcul de C_n^P il faut donc éviter n ! qui est supérieur à C_n^P.

On peut donc aller «plus loin» avec $C_n^P = \dfrac{n(n-1)\ldots(n-p+1)}{P\,!}$.

Deuxième Phase.

On utilise cette dernière formule qui nous amène à programmer une boucle.



Boucle de calcul de C
fondée sur la relation
de recurrence
$C_N^I = C_N^{I-1} \times (N - I + 1)/I$

L'instruction
C = C * (N - I + 1)/I
se noterait en algol
C : = C × (N - I + 1)/1
ou dans d'autres langages
$C \longleftarrow C \times (N - I + 1)/I$

A l'exécution on s'aperçoit d'abord que à cause de la symétrie des coefficients on pourrait, dès le début, si $P > \frac{N}{2}$ remplacer P par N - P. D'où un calcul plus court. Il faut préciser que le calcul est extrêmement rapide : à peine les valeurs de P et N entrées, la valeur C_N^P est imprimée.

De nouveau le résultat est tronqué à partir de C_{37}^{16}. Par contre, on n'arrive au «dépassement numérique» que pour C_{400}^{200}.

Par ailleurs, une question se pose. Même si on se limite à C_{37}^{16}, peut-on faire entièrement confiance aux derniers chiffres significatifs car la machine fait P multiplications et P divisions ? (Dans les divisions il y a certainement des résultats tronqués). Pourtant C_N^P est un entier et on pense alors faire faire à la machine le calcul que l'on fait à la main en simplifiant numérateur et dénominateur afin d'éviter toute «troncature».

Troisième phase.

Lorsque la machine fait un quotient tel que $C_N^I = C_N^{I-1} \times (N - I + 1)/1$. On va chercher le pgcd de $N - I + 1$ et I et diviser ces deux nombres par leur pgcd. Alors $\dfrac{N - I + 1}{I} C_N^{I-1} = \dfrac{a}{b} C_N^{I-1}$ avec a et b premiers entre eux. Mais b divise C_N^{I-1} puisque C_N^P est un entier donc il faut d'abord effectuer la division de C_N^{I-1} par b, puis le produit par a. D'où le programme simplifié (il faudrait compléter par les tests $P > \frac{N}{2}$, $P = 0$, $P = 1$).

```
10    INPUT N, P        (entrée de N et P)
20    PRINT «N=» ; N, «P=» ; P
30    C = 1
40    FOR I = 1 TO P
50    X = I
60    Y = N - I + 1
70    IF X = Y THEN 130
80    IF X > Y THEN 110
90    Y = Y - X
100   GOTO 70
110   X = X - Y
120   GOTO 70
130   A = (N - I + 1)/X
140   B = I/X
150   C = C/B
160   C = C*A
170   NEXT I
180   PRINT «C=» ; C
190   END
```

On pourrait ajouter des conditions pour arrêter ce programme dès que C_N^P a plus de 10 chiffres significatifs. Mais avant de fignoler, on teste afin de comparer les résultats aux précédents (avec $N < 37$). Alors là, grosse surprise, les résultats sont identiques. Par contre, il faut un temps considérable pour les obtenir. Bien sûr on s'y attendait mais peut être n'avait on pas bien réalisé que, introduire une recherche de pgcd dans une boucle, cela fait P recherches, puisqu'il y a P boucles...

Quant à l'identité des chiffres significatifs ; alors notre brave Hewlett ne fait donc pas de troncature ? ... C'est que voilà, cette petite cachotière (comme d'ailleurs vos calculatrices de poche !...) n'affiche pas tous les chiffres sur lesquels elle travaille, elle garde en réserve 2 chiffres significatifs au moins et sur P divisions la troncature n'affecte que les chiffres cachés tant que P n'est pas très grand. Donc, à la poubelle le programme avec pgcd ! ...

Une question encore ; nous avons vu expérimentalement qu'il fallait se limiter à $N \leqslant 37$. Peut-on prévoir ce résultat théoriquement ? C'est un de mes élèves de TS qui va relancer cette question. Plusieurs d'entre eux viennent en effet me trouver avec leurs calculatrices de poches neuves, afin de me demander à quoi sert la touche N ! Ils ne connaissent pas encore cette fonction. Je leur explique. Aussitôt, ils se mettent précipitament à pousser la machine dans ses retranchements. Je reconnais la réaction déjà observée souvent : «jusqu'ou peut-elle aller ?... Peut-on la coincer ?». Le maximum est 13 ! (Texas instruments). Ils repartent, mais, le lendemain, l'un d'eux revient en me disant : «je sais pourquoi elle s'arrête à 13 !» et il me sort une formule que j'avais complètement oubliée : la formule de Stirling,

$$\log N\,! \approx (\log N - \log e) + \frac{1}{2}\log 2\pi + \frac{1}{2}\log N$$

appliquée à 14 ! on constate qu'il y a bien un dépassement de capacité prévisible.

Je reprends alors ma recherche sur C_N^P

$$\text{Soit } N = 2k \qquad C_N^{N/2} = C_{2K}^K \approx \frac{\left(\frac{2K}{e}\right)^2 \times 2\sqrt{\pi K}}{\left(\frac{K}{e}\right)^{2K} 2\pi K} \approx \frac{2}{\sqrt{\pi K}}$$

$$\text{d'ou } \log C_{2K}^K = 2K\log 2 - \frac{1}{2}\log\pi - \frac{1}{2}\log K$$

$$\text{si } N < 100 \qquad \log K < 2 \qquad \frac{1}{2}\log K < 1 \qquad \frac{1}{2}\log\pi \approx 0{,}25$$

$$\text{et } \log C < 10 \implies 2K\log 2 - 1{,}25 < 10$$

$$2K\log 2 < 11{,}25$$

$$2K < \frac{11{,}25}{\log 2} \approx 37$$

le calcul est donc bien à limiter pour $N = 37$. Je reprends alors le 2ème programme en ajoutant des conditions pour que $N < 37$ et je le fais essayer dans une classe de terminale.

Evidemment je me fais ridiculiser car, après 2 ou 3 essais corrects, les élèves essayent $N < 0$ ou N non entier !...

Il me faut donc reprendre le programme afin d'ajouter des conditions pour que la machine proteste si N, ou P sont négatifs ou non entiers. Une fois ce programme au point, je me prépare à le ranger dans mes archives quand on me demande «Et si $N > 37$, alors, la machine est coincée...» (petit sourire de triomphe).

Quatrième Phase.

Puisqu'il en est ainsi, nous allons faire un calcul en «multiprécision» !... Tant pis s'il est parfaitement inutile !...

Il faut donc obtenir C_N^P même si N est très grand. Pour cela on considère C_N^P comme un nombre exprimé en base 10^6 et on range dans un tableau les tranches de 6 chiffres consécutives.

Exemple : $C_{30}^{15} = 155\ 117\ 520$

tableau de 2 tranches $T(1) = 155$
$T(2) = 177\ 520.$

Il reste maintenant a examiner comment faire les opérations sur les différentes tranches du tableau si on veut multiplier ce nombre par A ou le diviser par B

Division par B : exemple numérique.

T(1)	T(2)	T(3)			
123 456	123 456	123 400	100		
56	123 456		1 234	561 234	561 234
	561	123 400	T(1)	T(2)	T(3)

On remplace T(1) par le quotient entier de T(1) par 100. Soit R_1 le reste de cette première division. On prend comme 2ème dividende $R_1 \times 10^6 + T(2)$.
On remplace T(2) par le quotient entier de ce 2ème dividende, par 100. Soit R_2 le reste.
Le 3ème dividende est $R^2 \times 10^6 + T(3)$ etc...

On fait donc une boucle de calcul comme suit : s'il y a K tranches dans ce tableau

```
R = 0 (initialisation du 1er reste)
FOR I = 1 TO K
E = T(I)
```

```
T(I) = INT((E + (10 ↑ 6) * R)/B)
REM : INT est la partie entiˋere de l'expression
R = E + (10 ↑ 6) * R - T(I) * B
NEXT I
```

Multiplication par A : exemple numérique

$$\begin{array}{cccc} & 12 & 12 & \\ & \underbrace{123\,456}_{T(1)} & \underbrace{123\,456}_{T(2)} & \underbrace{123\,456}_{T(3)} \\ & & \times\ 100 & \\ \hline \underbrace{12}_{T(0)} & \underbrace{345\,612}_{T(1)} & \underbrace{345\,612}_{T(2)} & \underbrace{345\,600}_{T(3)} \end{array}$$

On multiplie T(3) par A. La retenue R_1 est le quotient entier de $T(3) \times A$ par 10^6.
On remplace T(3) par $T(3) \times A - R_1 \times 10^6$.
On multiplie T(2) par A, on ajoute la retenue R_1, la nouvelle retenue R_2 est le quotient entier de $T(2) \times A + R_1$ par 10^6.
On remplace T(2) par $T(2) \times A + R_1 - R_2 \times 10^6$ etc...

Donc on initialise la retenue à zéro puis on fait la boucle

```
FOR I = K TO 0   STEP - 1
E = T(I) * A + R
R = INT (E/(10 ↑ 6))
T(I) = E - R * (10 ↑ 6)
NEXT I
```

Phase provisoirement définitive.

Principe du programme :

La machine demande les valeurs de P et N

si $N \leqslant 0$ ou N non entier elle refuse
si $P < 0$ ou $P > N$ ou P non entier elle refuse
(en justifiant son refus)

si $P > \frac{N}{2}$ elle remplace P par $N - P$

si $P = 0$ elle donne $C_N^P = 1$

si $P = 1$ elle donne $C_N^P = N$

si $1 < P \leqslant \frac{N}{2}$ elle utilise la boucle de la 2ème phase c'est-à-dire à partir de la formule $C_N^I = C_N^{I-1} \times \frac{N - I + 1}{I}$ mais si $N < 37$ elle fait le calcul de la deuxième phase tandis que si $N \geqslant 37$ elle utilise la technique indiquée pour le calcul en multiprécision.

Il faut choisir dans le programme la dimension du tableau car en basic on ne peut donner une dimension variable. J'ai choisi 30 comme dimension mais en cas de besoin (N très grand) on peut modifier cette dimension.

Il faut initialiser le tableau pour y mettre le nombre 1 dans sa dernière «tranche» et zéro dans les autres.

Lorsqu'on lit le résultat il faut éventuellement ajouter des zéros pour que toutes les «tranches» aient 6 chiffres significatifs ex : si $T(3) = 67$, dans la suite des chiffres de C_N^P on aura 000067.

J'ai choisi des tranches de 6 chiffres pour qu'aucune multiplication intermédiaire ne dépasse la capacité de la machine mais si N n'est pas exagérément grand on peut prendre des tranches de plus de 6 chiffres. Avec des tranches de 6 chiffres on peut prendre pour N un nombre de 4 chiffres, le produit aura en général 10 chiffres (capacité maximum pour un entier).

Lorsque $N < 37$, le calcul est immédiat mais si N est grand il faut s'armer de parience. Exemple : 5 minutes pour sortir C_{300}^{150} dont voici la valeur
93 759 | 702 772 | 827 452 | 793 193 | 754 439 | 064 084 | 879 232 | 655 700 | 081 358 |
920 472 | 352 712 | 975 170 | 021 839 | 591 675 | 861 424.

Par la suite certains stagiaires ont essayé de reprendre le calcul de C_N^P avec la méthode du triangle de Pascal c'est-à-dire la formule de récurrence $C_N^P = C_{N-1}^P + C_{N-1}^{P-1}$. Je ne donnerai pas ce programme qui est très classique. Théoriquement il a l'énorme avantage de ne faire faire que des additions ce qui élimine les troncatures et fait des opérations plus rapides. Mais cela oblige à utiliser des tableaux ce qui allonge notablement la durée des calculs. (Par contre l'intérêt pratique pour un stage d'initiation est de faire justement manipuler des tableaux).

Si $N < 37$ c'est sans intérêt pour notre projet. Mais d'après Martinie, si l'on a une machine permettant d'utiliser une procédure récursive et si l'on fait des calculs en multiprécision, cela doit être plus performant (Encore faut-il que la machine utilisée fasse vraiment les additions beaucoup plus vite que les multiplications et divisions ...).

N'ayant pas cette possibilité sur la Hewlett je m'en suis tenue à la dernière phase...

Pour conclure cette recherche, je dirai qu'il n'est peut-être pas très utile de chercher C_{300}^{150} mais que cette étude met en évidence certaines vertus du mariage math-Informatique. D'abord la redécouverte du plaisir de l'expérimentation et de la recherche, avec son suspense ?... Ensuite, ce que je pourrais appeler les «retentissements» : on cherche un programme banal et on se trouve entrainé dans des allers et retours entre la machine et la mathématique, l'une obligeant à réfléchir sur l'autre et réciproquement. J'ajoute que j'ai aussi constaté ces «retentissements» auprès des élèves qui s'accrochent ainsi de manière surprenante à certains problèmes qui n'auraient probablement sucité qu'un intérêt peu appréciable sans la motivation de la programmation.

ont participé à la réalisation de cette brochure :

Monique Bernard (frappe), Gérard Buisine (Art. 3), Annie Martinet (Art. 6), Jean-Claude Monnet (Art. 1, 2, 4, 7), Jeanine Puig (Art. 8), les stagiaires du stage AI2 de 1976 (Art. 5) et l'équipe «inf» de l'IREM de Grenoble.